Volume 25 Number 3 October 2001 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Knowledge Based Software Engineering
Information Technology

Guest Editors:

Pavol Návrat
Cene Bavec, Matjaž Gams

The Slovene Society Informatika, Ljubljana, Slovenia

Informatica **25** (2001) Number 3, pp. 295–440

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at USA: http://, Europe: http://ai.ijs.si/informatica, Asia: http://www.comp.nus.edu.sg/ liuh/Informatica/index.html.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.

The subscription rate for 2001 (Volume 25) is
- USD 80 for institutions,
- USD 40 for individuals, and
- USD 20 for students

Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Tech. Support: Borut Žnidar, Kranj, Slovenia.
Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.
Printed by Biro M, d.o.o., Žibertova 1, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 1 4773 900, Fax (+386) 1 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Cybernetica Newsletter, DBLP Computer Science Bibliography, Engineering Index, INSPEC, Linguistics and Language Behaviour Abstracts, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 1000 Ljubljana, Slovenia.

Post tax payed at post 1102 Ljubljana, Slovenia taxe Percue.

Knowledge based software engineering (Introduction to special issue)

Special Issue Editor: Pavol Návrat
Slovak University of Technology
Department of Computer Science and Engineering
Ilkovičova 3, 812 19 Bratislava, Slovakia
navrat@elf.stuba.sk, http: www.elf.stuba.sk/~navrat/

Recent advances in the field of Software Engineering show that the field not only gradually matures but there are some important developments going on. This special issue is intended to give the readers of Informatica a focused view on some very interesting results in an area that is frequently named Knowledge Based Software Engineering (KBSE). Research in this area aims at advancing Software Engineering towards more automated methods by incorporating explicitly represented knowledge from the problem domain but also from the solution domain, i.e. design and programming knowledge. Research is taking inspiration also from Artificial Intelligence techniques in a wider sense.

The area of Knowledge Based Software Engineering has, despite its young age, a noteworthy record of history of its development. It is a standard practice that researchers in an emerging area first start gathering at scientific workshops and conferences. There have been at least two strings of international conferences devoted to Knowledge Based Software Engineering.

In 1983, Rome Air Development Center published a report calling for the development of a Knowledge-Based Software Engineering Assistant which would employ artificial intelligence techniques to support all phases of the software development process (this report appears in [1]). This initiated a series of regular conferences. In 1991, the conference was named the Knowledge-Based Software Engineering Conference, KBSE-6. After KBSE-11 in 1996, the decision was made to change the name to Automated Software Engineering Conference.

The other one is Joint Conference on Knowledge Based Software Engineering, which started in 1994 as a joint effort primarily of scientists from Japan and Eastern Europe. The Conference is well alive taking place biannually as a truly international gathering with participants from all over the world. Since 1998 in Slovakia, rigorously refereed and selected papers submitted to the Conference are published in a monograph book published by IOS Press [2] [3] . The next Conference will be in Maribor in 2002 [4] .

Such conferences are but one of several forums of communication. There are several international scientific journals where papers from this area can be found. However, this special issue of Informatica is one of the first such devoted to Knowledge Based Software Engineering (another one to be named is the special issue of the IEICE Transactions on Information and Systems from 2000 based on papers from the 1998 JCKBSE Conference [5] .

It is an extraordinary honour for me that the Editorship of Informatica have entrusted me with editing this special issue, for which I am very thankful.

I have take special care in preparing this issue. Papers were solicited in a regular process that started with a call for papers. Thus, this special issue is not a collection of papers presented at some conference. All the submissions were prepared originally for this special issue. Here are the main topics from the call for papers showing roughly what is understood under the description Knowledge Based Software Engineering:

- software architecture and component reuse
  - agents on internet and intranet
  - domain modelling
  - semiformal and formal specifications
  - design patterns
- post-object-oriented programming paradigms including
  - agent-oriented
  - aspect-oriented
  - generative programming
  - multiparadigm appraches
- knowledge discovery
- program understanding and learning.

Response from prospective authors has been quite encouraging. I received some dozen submissions. After a preliminary review, 9 submissions were subject to a further rigorous refereeing process. Here, work of all the assigned reviewers has been invaluable for a successful completion of the

process. Their carefully elaborated reviews were essential in deciding on acceptance and rejection. Finally I selected 4 papers for the special issue which amounts for an acceptance rate well below 50%. I wish to emphasise this fact, which illustrates the high standard of quality of Informatica. I wish to thank all the authors who submitted to the special issue. There could be found many interesting ideas in all the submissions. Even some submissions, which ultimately had to be rejected, were very good papers.

The papers have been accepted solely based on their scientific merit. However, as it turned out, their themes spread nicely across a wider spectrum of topics so the readers get an opportunity to make themselves acquainted with several interesting problems and approaches to solving them.

One of the central themes in the current Knowledge Based Software Engineering research is to advance understanding of design patterns, frameworks, and applications. All these concepts can be viewed also as various kinds of knowledge. Knowledge Engineering offers a different conceptual framework to view and to deal with such knowledge. But there are more common concepts that can be found in both software systems and intelligent systems: let us name just intelligent software agents, and components.

In the collection of papers, the first one by Soundarajan and Fridella is titled *Understanding OO Frameworks and Applications: An Incremental Approach.* When reusing the design built into a framework, application developers need to understand behaviour of such frameworks. Authors propose techniques that will allow the framework developer to specify the framework behaviour precisely and allow the application developer to make use of it when building their application from the framework.

The second paper *Towards Software Design Automation with Patterns* by Sikici and Topaloglu focuses on software development process based on applying design patterns. A new approach to describing patterns, viewed as knowledge pieces, is presented. Patterns as represented by their descriptions can be combined or abstracted in the development process.

The next paper *Support of Knowledge Management in Distributed Environment* by Paralič, Paralič and Mach deals with a slightly different topic: knowledge management within an organisation. Their special focus is on organisations with distributed environments. Their approach makes use of mobile agents and distributed programming. Resulting system provides support to distributed organisational memory as a way of domain knowledge modelling.

The fourth paper by Lee and Tsai is on *A Novel Agent to Efficiently Extract the Top K Interesting Target Ranks from Web Search Engines.* They tackle a rather specific problem, which however falls within a very important area of efficient Web searching. The paper shows also how research focus in Software Engineering shifts towards problems connected with Internet applications.

Finally, I should like to express my hope that readers will find the special issue interesting and inspiring. My thanks go to the following reviewers:

Pavel Brazdil	Mark Nissen
Davide Bugali	Jens Penberg
Mel Ó Cinnéide	Gabika Polčicová
Peter Dolog	Wilhelm Rossak
Amnon Eden	Zhongshi Shi
Ivan Kapustík	Mária Smolárová
Ann Macintosh	Gheorge Tecuci
Jerzy Nawrocki	Valentino Vranić
Roumen Nikolov	Stefan Wrobel

I wish to thank Professor Matjaž Gams for his perfect support in my work.

References

[1] Readings in Artificial Intelligence and Software Engineering, Morgan Kaufman, 1986.

[2] Návrat P. & Ueno H. (Eds.) (1998): Knowledge-Based Software Engineering. Frontiers in Artificial Intelligence and Applications, Vol. 48, IOS Press, Amsterdam, 334 pp.

[3] Hruška T. & Hashimoto M. (Eds.) (2000): Knowledge-Based Software Engineering. Frontiers in Artificial Intelligence and Applications, Vol. 62, IOS Press, Amsterdam, 342 pp.

[4] Joint Conference on Knowledge Based Software Engineering 2002: http://kbse.cs.shinshu-u.ac.jp/conf/conf.html

[5] Special Issue on Knowledge-Based Software Engineering, IEICE Transactions on Information and Systems, Vol. 83-D, No. 4,: 2000.

Understanding OO frameworks and applications: an incremental approach

Neelam Soundarajan and Stephen Fridella
Computer and Information Science
Ohio State University, Columbus, OH 43210
e-mail: {neelam,fridella}@cis.ohio-state.edu

Keywords: Object-oriented application frameworks, software reuse, formal specifications, verification

Received: March 18, 2001

Object-Oriented frameworks provide a powerful technique for developing groups of related applications. Individual designers may develop applications tailored to their specific needs by providing appropriate definitions for the hook *methods, while reusing the design built into the framework. One important problem in using this approach is the lack of suitable techniques for precisely documenting the behavior of such frameworks. If such techniques were available and were used to document framework behavior, application developers would be able to more precisely understand the framework, be able to more effectively build their applications from the framework, and be able to reason more reliably about the behavior of these applications. Our goal is to develop such techniques. The techniques we develop will allow the framework designer to specify the framework behavior precisely, and allow the application developer to 'plug-in' information about the behavior of the hook methods into the specification of the framework to arrive at the behavior of the application. We illustrate the technique by applying it to a simple case study.*

1 Introduction

Object-Oriented (OO) frameworks [8, 16] promise to dramatically reduce the time and effort needed to develop complete applications. The framework designer designs the main classes and methods, including key methods that direct the flow-of-control and call appropriate methods of various classes. These key methods are referred to as *template methods* in the design patterns literature [10], the methods they call being the *hook methods*. In order to implement a complete application based on such a framework, the application developer treats (some of) the classes of the framework as base classes and defines derived classes that include definitions for the hook methods that implement behavior appropriate to his[1] particular application. A different developer could produce a (possibly very) different application by providing an alternate set of definitions for the hook methods. In effect, the framework designer implements behavior that is common to the various applications that may be built on this framework, including such behavior as the patterns of calls implemented in the template methods, and the individual application developer enriches this common behavior in ways appropriate to his particular application by defining (or redefining) the appropriate hook methods. If the framework has been designed carefully and provides the right hooks, an entire new application can be developed with just the incremental effort involved in designing a new set of definitions for the hook methods.

But it is not sufficient to be able to *build* these applications, we also need to be able to *reason* about them so that we can understand and document their behavior. And the reasoning technique should itself be incremental. In other words, we should be able to reason just once about a given framework and arrive at a suitable specification for it; then, when a new application is built on this framework, we should be able to 'plug-in' appropriate information about the (behavior of the) hook methods defined in the application to arrive at a suitable specification of the behavior of the entire application. Unless the reasoning technique is incremental in this sense, the framework approach will be little more than a mechanism for code reuse. Our main goals in this paper are to develop an incremental reasoning technique for frameworks and applications, and to demonstrate it on a simple case study.

In our approach to documenting frameworks, we will specify the behavior of each method of each class in terms of a pre-condition and a post-condition [21]. The key problem we have to address is that of specifying the behavior of the template methods of the framework in a way that suits the needs of the application developer. Suppose, for example, that $t()$ is a template method of the framework. We will see the full details later, but the key is to include, in the specification of $t()$, information about which hook methods $t()$ invokes, the objects it invokes them on, the order it invokes the methods in, etc. In order to provide this type of information, we will make use of a *trace*, denoted by the symbol τ, to record the hook method calls that the template method makes. Our traces are somewhat similar to the traces used in reasoning about distributed systems [22] except that whereas we use them to record information about hook method calls, in dealing with distributed systems they are used to record information about communications be-

[1] Following standard practice, we use 'he', 'his', etc. in place of 'he or she', 'his or her' etc.

tween the processes. The specification of $t()$, in particular its post-condition, will give us information not only about the final values of the member variables but also information about which hook methods $t()$ invoked along the way, on what objects, etc. As we will see, the application developer will then be able to plug into this specification, the behaviors corresponding to his particular (re-)definitions of the hook methods, to arrive at the corresponding new behavior of $t()$.

There is one key requirement that hook method (re-)definitions must satisfy. Suppose $X.h()$ is one of the hook method invocations in $t()$, X being the object and $h()$ the method invoked. In arriving at the specification of $t()$ by analyzing its code in the framework, the (framework) designer would have made some assumptions about the effects of this call to $h()$. Typically, these would correspond to the behavior exhibited by the default definition of $h()$ provided as part of the framework, or if no default definition is provided, then the behavior that is expected to be common to the different definitions that will be provided in the different applications built on this framework. This default (or expected) behavior will be part of the specification of the framework. Unless the (re-)definition of $h()$ in the application satisfies this specification, the analysis of $t()$ may no longer be valid, and we would be forced to reanalyze the body of $t()$. Since we want to avoid such reanalysis of the framework code, we will require that the redefinition of $h()$ in the application satisfy its specification in the framework. Such a requirement is not new to our work; it is the key idea underlying the work on *behavioral subtyping* [1, 19, 20]. What is new about our work is that, if this requirement is satisfied for each of the hook methods, then the "plugging-in" process we outlined above will allow us to arrive at the *richer* behavior that $t()$ acquires as a result of the (re-)definition of $h()$. Since the entire notion of OO frameworks rests on this ability to enrich, in the application, the common behavior provided by the framework, it is essential that the reasoning system enable us to reason about this enriched behavior.

1.1 Related Work

The importance of proper documentation of frameworks has been widely recognized and a number of ways [2, 5, 9, 11, 13, 15, 18, 23, 24, 28] have been developed for the purpose. Some of these, for example, [2, 5, 18, 23, 24], are informal and focus on training application developers to use the framework in ways intended by the framework designer using cookbook-style patterns. Others are more formal and focus on developing techniques for specifying and reasoning about the behavior of the framework and applications built using the framework. Since our goal is to develop a formal reasoning method, we will restrict attention to this latter group. One important aspect of our work that distinguishes it from other approaches is that, we are interested in not just ensuring that the application meets the constraints imposed on it by the framework but also in arriving at the richer behavior resulting from the definitions (of the hook methods) adopted in the application, whereas most of the others focus on just the first point, i.e., ensuring that the application meets the framework's constraints.

In their work on documentation of frameworks, Campbell and Islam [4] consider what might be called *structural* questions, such as identifying which objects are related to which other objects, specifying constraints on which classes may be combined with which other classes, etc., whereas our focus is on behavioral questions. But for complete documentation of complex frameworks and applications, we must deal with both structural and behavioral questions, so our approach is in some sense complementary to that of [4].

Helm *et al.* [13] introduce the notion of *contracts* to capture important aspects of the interactions between objects in a system. A contract imposes conditions on what calls to certain *virtual*[2] methods must do and under what conditions, whereas we will not impose such constraints. We will, as already outlined, require hook-method redefinitions to satisfy their framework-level specifications, but these are *functional* requirements whereas *contracts* allow one to impose requirements on, for example, what other methods the called hook method may, in turn, call. But [13] does not deal with the question of how to combine the specific behaviors implemented by the hook methods as defined in the application with the specification of the framework to obtain the behavior of the complete application.

D'Souza's [7] *Catalysis* method focuses on the development of components and frameworks. It uses pre- and post-conditions as well as *statecharts* to specify transitions between key (abstract) states; external calls, including calls to hook methods, can be encoded in these statecharts. Although Catalysis provides a rich set of notations and tools to express behavior, it is not clear how it deals with the question of ensuring that the application developer abides by the framework's requirements nor how to plug-in the extra behavior provided by the application, to arrive at the application-specific behavior.

Froehlich *et al.* [9] use the term *hook* to denote some aspect of the framework that can be used to tailor the system to the application's needs. A hook method is a simple hook, but it is possible to have more complex hooks as well. Suppose, for example, two hook methods are intended to cooperate in some manner and if one of them is redefined in the application then the other one would also (very likely) have to be redefined in order to be true to the intent of the framework designer; these two methods will together be considered a hook. Froehlich *et al.* propose a formal syntactic notation for specifying such hooks but do not deal formally with questions of behavior.

Buchi and Weck's [3] work is closer to our approach. They note that pre- and post-conditions on just the values of member variables are inadequate when dealing with

[2]For concreteness, we occasionally use language-specific terminology although our reasoning technique applies to any class-based OO language; thus 'virtual' is in *C++*; in *Eiffel* or *Java*, we would say 'non-final'.

template methods. They introduce a formalism and a programming language-like notation using which some information about the trace of hook method calls can be specified. Although their use of traces is similar to ours, Buchi and Weck focus only on specifying conditions that the trace must satisfy. They do not address the question of establishing the richer behavior resulting from the redefinitions of the hook methods in the application.

Garlan *et al.* [12] develop a temporal-logic based approach to reasoning about systems involving *implicit invocation*. Calls to hook methods are essentially implicit invocations since the actual method invoked is defined in the derived class which may not even be implemented when the framework is designed. Conversely, event-based systems, the type of implicit invocation systems that [12] considers, can be rewritten using template methods and hook methods. Indeed, the approach of [12] requires us to reason about the *dispatcher* method which is responsible for handling the implicit invocations, and this method is analogous to standard template methods found in OO frameworks. One difference with our approach is that our specifications are non-temporal and hence somewhat easier to work with; on the other hand, our specifications involve traces explicitly whereas those of [12] do not. A more important difference is that a key goal of our approach is to first reason about the framework independently of any application that may be built on it and then, when an application is built by (re-)defining the virtual methods of the various base classes of the framework, to arrive at the behavior of the entire application by plugging information about the methods defined in the application, into the specification of the framework. By contrast, the approach of Garlan *et al.* is more of a top-down approach, so to speak; that is, given a system S (consisting of all the methods in all the classes) and a specification for it, they partition S into a number of groups (each consisting of a number of methods), arrive at a suitable specification for each group, show that each group satisfies its specification, and show that together the specifications of the individual groups, imply the original specification of S. For frameworks, since many applications may be developed on a given framework, our approach of reasoning independently about the framework and then plugging in application-specific information when the application is built, would seem preferable.

In an earlier paper [26], we considered a related problem. Our goal in that paper was to reason about a single base class and derived classes that may be defined from it. Although that problem is related to the problem of reasoning about frameworks and applications, there are also important differences. For example, in the case of frameworks, there are several different objects each with its own static type, and with its own dynamic type; by contrast, in the situation considered in [26], there is a single object whose static type is the base class in question and dynamic type is the particular derived class under consideration. Again, in the case of frameworks, the result of redefinitions of virtual methods in a derived class in the application is to enrich not only the other methods of the corresponding base class but also, and indeed more importantly, enrich the methods of the *controller* class of the framework. By contrast, in [26] the only enrichment resulting from redefinitions of virtual methods in a derived class is of other methods of the base class that invoke these virtual methods. Despite these differences, we were able, as in the current paper, to use a trace-based approach to address the problem considered in [26]. This should not be surprising; traces are a fairly powerful tool and have been used for reasoning about different types of systems ranging from sequential to concurrent to distributed and real-time systems. Indeed, a trace-based model also underlies the approach of Garlan *et al.* although traces do not explicitly appear in their reasoning technique. But "there is no free lunch", as the saying goes; for exploiting the power of traces, we pay a price in the form of somewhat more complex assertions in our specifications, as we will see in the case study. This complexity can be somewhat mitigated by defining suitable functions and predicates over traces for use in our specifications. We will return to this issue in the final section of the paper.

The main contributions of this paper may be summarized as follows:

- It identifies the problems involved in documenting precisely the behavior of frameworks and in reasoning about the behavior of applications built on frameworks.
- It develops a reasoning technique to allow the framework designer to specify the framework, and to allow the application developer to incrementally arrive at the specification his application from that of the framework.
- It illustrates the reasoning technique by applying it to a simple case study.

The rest of the paper is organized as follows: In Section 2 we present a simple model of frameworks and applications. In Sections 3 and 4 we develop our approach to specifying and reasoning about the behavior of frameworks and applications. In Section 5 we apply the approach to a simple application. In the final section we reiterate the importance of precise reasoning about framework behavior, summarize our solution to the problem, and identify some directions for future work.

2 Model Of Frameworks

We will say that a class A is *refineable* if at least one of its methods is virtual[3]. A class all of whose methods are

[3]Our refineable classes are very similar to *abstract classes*, the key difference being that at least one of the methods of an abstract class must be *pure virtual* (*deferred* in *Eiffel*), i.e., have no body defined for it in the class. In this case, the derived class would have to provide a definition for this method, else objects of this type cannot be created. Our approach deals with virtual and pure-virtual methods in very similar ways so we will generally not distinguish between the two. One language-specific point to note is that in C++, if a method is *virtual* (or *pure virtual*) in a given

non-virtual is a *concrete* class. A framework $\mathcal{F}$ will consist of the following classes:

- A concrete class C which we will also call the *Controller* class of $\mathcal{F}$. C will have a distinguished method named '*run*()'. As the name suggests, it is this method that primarily decides how control flows among the various methods of the various classes of $\mathcal{F}$.
- Zero or more other concrete classes $C_1, \ldots, C_m$.
- One or more refineable classes $A_1, \ldots, A_n$.

In addition to the *run*() method, many frameworks include a mechanism for initialization in order to allow some information about the actual application to be passed to the framework. Although details of the initialization are important in practice, in this paper we will ignore them so that we can focus our attention on reasoning about the *run*() method, the hook methods (of the refineable classes $A_1, \ldots, A_n$) that it invokes, and the possibilities that these provide to the application developer to enrich the behavior provided by the framework.

An application $\mathcal{A}$ developed using $\mathcal{F}$ will have, corresponding to each refineable class A_j of $\mathcal{F}$, one or more derived classes $CA_{j,k}$ each of which will provide definitions for some or all of the virtual methods[4]. $\mathcal{AC}$ will denote this additional code provided by the application builder. In order to use the application $\mathcal{A}$, we would (in the 'client program' using $\mathcal{A}$) create an object, let us call it d, of type C, initialize it, and then invoke *run*():

$d.run()$

Objects such as d will have components of type $C_1, \ldots, C_m$ as well as $A_1, \ldots, A_n$ (or rather $CA_{1,1}, \ldots$); the client program itself will not have any objects of these types.[5] Inside the *run*() method we will call the various methods defined in $C_1, \ldots, C_m$ and $A_1, \ldots, A_n$, some of which will be virtual methods, and hence will call the corresponding methods defined in the appropriate concrete classes defined in $\mathcal{AC}$, to act upon these components of the main object d. In this manner, the application will exploit the common framework design embedded in the *run*() method, as well as any other methods defined in $\mathcal{F}$, while at the same time exploiting the application-specific behavior defined in $\mathcal{AC}$.

We conclude this section with an observation. The main contribution that a framework makes is relieving the application developer of having to worry about how control should flow among the various methods of the various classes. In our model, this is highlighted by the *run*() method. None of the other methods of A_i is allowed to invoke the hook methods because if they were allowed to do so, that would amount to embedding some of the control flow logic into these other methods. Not every framework fits within our model. For example, it is possible that the control flow is in fact distributed among several methods, perhaps even methods of several classes. Nevertheless, our model captures the essential aspects of frameworks. The key ideas underlying the reasoning technique we develop in the next two sections focusing, as it does, on dealing with characterizing the control flow in the framework, apply also to frameworks that do not fit exactly within our model.

There is another, somewhat technical, restriction we impose on our frameworks: All the objects in our frameworks are *named*, i.e., there are no anonymous objects. In practice, frameworks often do have anonymous objects that are accessed via pointers or other references to them. In the final section we will briefly consider how we can use our approach to reason about such frameworks (and applications).

3 Reasoning About The Framework

In order to specify the framework, we need to provide specifications for each of the concrete classes $C_1, \ldots, C_m$, each of the refineable classes $A_1, \ldots, A_m$, and the controller class C. Since our approach is an extension of the standard approach, see for example [17], to reasoning about classes, we start with a brief summary of that: In reasoning about a class B, we first write down the specification of each method of B as pre- and post-conditions over the concrete state, i.e., the values of the member variables, of B, check (or formally verify) that the method body satisfies its concrete specification, define a conceptual model of B, write down the abstract specification of each method as pre- and post-conditions over the conceptual state, define a function (or relation) mapping the concrete state to the conceptual state, and finally check that if a method satisfies its concrete specification, then it must also, given our mapping from the concrete to the conceptual state, satisfy its abstract specification. Once this is done, the concrete specification is essentially forgotten since data abstraction, a basic tenet of the OO approach, requires the client of B to take an abstract view of the class and not worry about its internal details. However, for a designer of a derived class D of B, the concrete specification of B is important since the methods he (re-)defines in D will in general access the member variables of the base class B and thereby interact with the methods defined in B, so it is important for this designer to know how these methods manipulate the member variables[6] of B.

class, it remains virtual in derived classes; our intention is that all virtual methods will become, in *Java*-terminology, *finalized* in the applications.

We should also note that we are taking a bit of liberty with the language in using 'refineable' to mean 'capable of refinement'.

[4]We should note that $CA_{j,1}$, for example, might be the same as A_j (if A_j does not contain any pure virtual methods); for the sake of uniformity of notation, we will still refer to it as $CA_{j,1}$ in the application.

[5]This seems to be the standard pattern for using applications built on frameworks. In view of this, it may be useful for programming languages to provide *syntactic* mechanisms that 'bundle' the classes of $\mathcal{F}$ into a syntactic unit, identify the controller class C of $\mathcal{F}$, etc. That will make it clear that the 'client' program is only expected to have objects of type C, and that classes $C_1 \ldots, C_m$ etc. are of no direct relevance to the client.

[6]Some authors, see for example [14], have argued that the derived class should have no direct access to the member variables of the base class but

Consider the concrete class, C_i. The framework is a client of C_i. Hence, all we need is the conceptual specification of each method of C_i. This specification is not for use by the final client of the application since the class C_i is for use (only) by $\mathcal{F}$. Rather, when dealing with a call of the form $ci.f()$ in the body of *run*(), *ci* being a member variable of type C_i (of the controller class) of the framework and $f()$ one of the methods of C_i, we will appeal to the conceptual specification of $f()$ to understand the effect of this call. The situation is somewhat different for the refineable class A_i since the application developer will define one or more derived classes of A_i in which one or more of the hook methods of A_i will be (re-)defined. This means, following the comments in the last paragraph, we need the concrete specification of the behavior of the methods of A_i. If *ai* is a member of the controller class C, and the static type of *ai* is A_i, then in dealing with a call such as $ai.h()$ in the *run*() method, $h()$ being a method of A_i, we will appeal to the concrete specification of $h()$.

Most interesting is the controller class C, in particular its *run*() method. As noted earlier, it is this method that makes it possible to exploit the enriched behavior that the application developer implements in the application. For example, if the developer redefines the virtual method $h()$ in $CA_{i,j}$ and if the dynamic type of the object that *ai* refers to is $CA_{i,j}$, then the call $ai.h()$ in *run*() will be dispatched to this redefined method, $CA_{i,j}.h()$. In order to allow the developer to reason about the resulting enriched behavior, we need to record information about this call in the specification of *run*(). For this purpose, we associate a *trace* τ with *run*()[7] and record all calls it makes to hook methods such as $h()$.

What information about the call $ai.h()$ do we need to include in the trace τ and how do we record it? This call (and the corresponding return) will be recorded by appending the following element to τ:

$$(ai,\ h,\ \bar{x},\ ai.\bar{v},\ \overline{xx},\ ai.\overline{vv}) \qquad (1)$$

The first component identifies the object involved, here *ai*. The second is the hook method called, here $h()$. The third gives us the values of the arguments passed to $h()$. The fourth component is the state, i.e., the values of the member variables, of the object (*ai*) at the time of the call. The next component is the values of the arguments when $h()$ returns[8] The final component is the state of the object when $h()$ returns.

Let us consider each of these components in turn and see why it needs to be included. The first component, the identity of the object involved, is needed since otherwise we would not know which component of the framework will be enriched by the redefinition of $h()$ in the application. The second component is needed since without knowing which method was invoked, we cannot even know whether any enrichment is involved. The next two components are needed since the enrichment in question may depend on these values. For example in the banking framework considered in Section 5, one of the virtual methods of *Account*, one of the refineable classes of the framework, is *withdraw*(); in an application built on this framework, we may redefine *withdraw*() to keep a count of withdrawals of more than a specified amount; if we want to be able to reason about the resulting enrichment of the *run*() method, it is not sufficient to know which account objects *withdraw*() was applied on, but also the amount withdrawn, in other words the argument value. Similarly, if the enrichment depends on the state of the object, say the current balance in the account, we will need the information in the fourth component to reason about the resulting enrichment of *run*()'s behavior. The information in the last two components is needed in some unusual cases. In particular, if the framework-level specification of $h()$ does not prescribe *specific* values (depending upon the values of $\bar{x}$ and $ai.\bar{v}$) to be assigned to the various member variables or the arguments, but instead allows the method to assign a range of values, and if the subsequent behavior of *run*() depends on these values, then in order to express the relation between this behavior and the state of the object (or the values of the arguments) at the time of the return from the initial virtual method call, we would need to refer to this state in the specification.

The rule **R1** below is a standard procedure call rule, modified to take account of recording of the call in the trace τ:

R1. Hook Method Call Rule

$$\begin{array}{l} p \Rightarrow pre.A_i.h(\overline{y})[\overline{y} \leftarrow \overline{x},\ \overline{v} \leftarrow ai.\overline{v}] \\ \forall(\overline{xx}, \overline{vv}). \\ \quad \{\ (p \wedge post.A_i.h(\overline{y})[\overline{y}@pre \leftarrow \overline{x}, \\ \qquad \overline{v}@pre \leftarrow ai.\overline{v},\ \overline{y} \leftarrow \overline{xx},\ \overline{v} \leftarrow ai.\overline{vv}\,]\ ) \Rightarrow \\ \quad (\ (\ q[ai.\overline{v} \leftarrow \overline{vv},\ \overline{x} \leftarrow \overline{xx},\ ])\,[\tau \leftarrow \\ \qquad \tau\hat{}(ai, h, \overline{x}, ai.\overline{v}, \overline{xx}, \overline{vv})\,]\ )\ \} \end{array}$$

$$\{\ p\ \}\ ai.h(\overline{x})\ \{\ q\ \}$$

The first antecedent requires us to show that the precondition of $h()$ of the class A_i is satisfied with $\bar{x}$ playing the role of the formal parameter $\bar{y}$ and the state $ai.\bar{v}$ of the involved object *ai* playing the role of the member variables $\bar{v}$ of the class. '$\leftarrow$' denotes substitution of the indicated variables.

In the second antecedent the '*@pre*' notation [27] ap-

that instead, the base class should provide a special set of (*protected*) operations that would be available to the derived class but not to client code. These protected operations would permit the methods in the derived class to manipulate the variables of the base class but only in ways allowed by the protected operations. Even with this approach, we would still need to utilize two types of specifications; the first, an abstract specification, for use by clients of the (base) class; the second, a (more) concrete specification that includes information about the behavior of the protected operations, for use by the derived class designers.

[7]Recall that in our model, none of the other methods of A_i may invoke virtual methods such as $h()$ of A_i. If this restriction were relaxed, the redefinition of $h()$ in $CA_{i,j}$ will result in this other method of A_i also being enriched since the call to $h()$ in this method will also be dispatched to the redefined method. To deal with such enrichments, we would have to associate a trace also with any such method of A_i and record all such calls to virtual methods.

[8]We assume, for simplicity, that all arguments are of simple types such as *int* and are passed by value-result.

pearing in the post-condition of a method refers to the value, at the time the method was called, of the variable in question. Being able, in the post-condition, to refer to the values of variables at the time the method started, often simplifies the form of the post-conditions. "^" denotes appending an element to the trace. The second antecedent may be understood as follows: As far as the framework code is concerned, the effect of the call to $h()$ is essentially to assign new values to τ (by appending to τ an element representing the call), to $ai.\overline{v}$ (because of the new values that will in general be assigned by the body of $h()$ to the member variables of ai), and to the arguments $\overline{x}$ (since our parameters are passed by value-result). This may be represented as three assignments; the first of these is:

$$\tau := \tau \hat{} (ai, h, \overline{x}, ai.\overline{v}, \overline{xx}, \overline{vv}) \quad (2)$$

which appends, to τ, an element representing the call; $\overline{xx}$, $\overline{vv}$ represent respectively the final values of the arguments and the final state of ai when the method finishes execution. This is followed by the two assignments:

$$ai.\overline{v} := \overline{vv}\,; \quad \bar{x} := \overline{xx} \quad (3)$$

representing respectively the effect of the call on the state of ai and on the arguments $\bar{x}$. Thus the second antecedent of **R1** checks that the pre-condition p and the post-condition q of this call are appropriately related, the substitutions to q reflecting the effect of the three assignments. The quantifiers on $\overline{xx}$, $\overline{vv}$, and the post-condition (with appropriate substitutions) on the left side of the implication of this antecedent ensure that we consider all possible values (constrained by the requirement that they satisfy the conditions imposed by $postA_i.h()$) that the call may leave in the arguments and in the member variables of ai.

It is worth noting that although **R1** allows us to include complete information about each component of each element of τ in the specification of *run*(), doing so may often result in unwieldy specifications. In practice, the framework designer must cull this information and include only those aspects that he expects will be of use to the application developer in building actual applications. In doing so, the designer does run the risk of leaving out some information that a particular application developer could have profitably exploited. We will return to this point later in the paper.

4 Plugging In The Application

When an application is built on a framework, the most common type of enrichment is to add variables to one of the refineable classes A_i and redefine its virtual methods to update the new variables when these methods are invoked. For example in our banking framework considered in the next section, we will enrich the *Account* class of the framework by adding a member variable *tCount* which will keep count of the transactions performed on the account; correspondingly, the *withdraw*() and *deposit*() operations will be redefined so that they not only update the *balance* in the account but also increment *tCount* when they are invoked.

The first step in reasoning about the application is to write down (and, if necessary, verify) the new specifications for the redefined methods. The details of this will, of course, depend on the details of the redefinitions. Specifications are also written down for the inherited methods; these are identical to their framework-level specifications with an additional clause in the post-condition that asserts that the values of the new member variables are the same as their values when the method started, i.e., a clause of the form:

$$(nv = nv@pre) \quad (4)$$

for each new variable nv. There is no need to verify these specifications since the methods have been verified to satisfy their original specifications and the new clauses of the form (4) must necessarily be satisfied since clearly the original methods could not have accessed variables such as nv.

Next, we must check that the redefined methods also satisfy their framework-level specification, this being the behavioral subtyping requirement we discussed earlier. In other words, if $h()$ is one of the virtual methods of A_i that *run*() invokes, then in reasoning about *run*(), using the rule **R1** to deal with calls to $h()$, we would have appealed to its framework-level specification. So when the developer redefines this method in a derived class $CA_{i,j}$ of A_i, unless the redefinition satisfies its framework-level specification, our reasoning would become unsound if the object on which $h()$ is invoked is of dynamic type $CA_{i,j}$. In practice, this check is usually simple since the most common redefinitions of the hook methods are such that they do exactly what their framework-level bodies did as far as the member variables of A_i are concerned, and it is only these variables that can appear in the framework-level specification of $h()$; quite often, the redefined method is written as a call to the base class $h()$ followed or preceded by additional code to update only the new variables defined in $CA_{i,j}$. In this case, since the code that manipulates the framework-level variables is the same as before, it follows immediately that the redefined method satisfies the corresponding framework-level specification.

The final, and perhaps most important, step is to arrive at the enriched behavior of the *run*() method. Suppose the framework-level specification of *run*() is:

$$\{ p_f \wedge \tau = \varepsilon \}\ run()\ \{ q_f \} \quad (5)$$

where ε is the empty trace (at its start, *run*() has not made any hook method calls, so τ must be empty); p_f specifies information about the initial states of the objects that the framework manipulates; and q_f the final states of these objects as well as the final value[9] of τ. We will assume that the framework designer has established (5).

[9] We should note here that we are assuming that our *run*() method terminates. Many frameworks/applications are intended to run for ever. For dealing with such frameworks, we would have to use invariants; these invariants would be quite similar to our post-conditions and would be in terms of the states of the objects and the values of τ.

Suppose now we wish to arrive at the following application-level specification for *run*():

$$\{ p_a \wedge \tau = \varepsilon \}\ run()\ \{ q_a \} \quad (6)$$

where p_a, like p_f, specifies information about the initial states of the objects that the framework manipulates, but for each object, p_a will include information also about the values of the new member variables introduced in the derived class (which is the dynamic type of the object in question) defined in the application, whereas p_f would only contain information about the member variables of the corresponding base class defined in the framework. Similarly, q_a would specify information about the complete final states of the various objects, including the values of the member variables introduced in the derived classes, whereas q_f specifies only the 'framework portion' of the state of each object. Indeed, much of what we mean by enriched behavior is the information about the 'application portion' of the state of each object that q_a provides and q_f does not.

The key question is, how do we arrive at the enriched specification (6) by plugging the application-specific behavior of the hook methods into the framework-level specification (5)? We will first introduce a number of notations that will be of use in answering this question. Below, we will use σ, σ' etc. to denote the 'state of the application', i.e., the states of all the objects manipulated by the application; the current 'state of an object' is essentially the current values of all the member variables of the object; we will use c, c' etc. to refer to the individual objects.

$\sigma(c)$: The state of the individual object c in the (application) state σ.

$\sigma(c)\uparrow$: The 'framework portion' of the state of the object c.

$\sigma(c)\downarrow$: 'Application portion' of the state of the object c.

$\sigma\uparrow$: The framework portion of the state σ (i.e., for *each* object, we project out the framework portion of its state).

$\sigma\downarrow$: The application portion of σ.

$|\tau|$: Length of τ, i.e., the number of elements in τ.

$\tau[k]$: k^{th} element of τ.

$\tau[k].ob$: The first component of $\tau[k]$, i.e., the identity of the object involved in this hook method call.

$\tau[k].hm$: The second component of $\tau[k]$, i.e., the hook method called.

$\tau[k].ar$: The third component of $\tau[k]$, i.e., the argument values passed to the method.

$\tau[k].re$: The fifth component of $\tau[k]$, i.e., the results returned at the end of this call.

$\tau[k].is$: The fourth component of $\tau[k]$, i.e., the initial state of the object $\tau[k].ob$ at the time of the call.

$\tau[k].fs$: The sixth component of $\tau[k]$, i.e., the final state of the object $\tau[k].ob$ at the time of return.

$\tau\uparrow$: Sequence obtained from τ by retaining only the framework portions of each of the two states in each element of τ; similarly $\tau\downarrow$ is obtained by retaining only the application portion of the states.

Consider now the question of going from (5) to (6). The relation that must hold between the respective pre-conditions p_a and p_f is easy to see: since, in the application, we wish to reuse the framework-level reasoning, we must have:

$$p_a \Rightarrow p_f \quad (7)$$

The pre-conditions specify the effect of the initializations. (7) requires that the application-specific initialization ensure what has been assumed at the framework level.

The relation between the post-conditions is more involved. Suppose we have states σ, σ' that we expect are possible initial and corresponding final states of the application and τ is the value that will be accumulated into the trace during the execution of *run*(), in going from σ to σ'. Then we note:

1. The framework portions of σ, σ', τ must satisfy the framework-level specification (since otherwise they cannot arise according to that specification and hence they cannot also arise in the application since the redefinitions of the hook methods in the application are consistent, in the sense of behavioral subtyping, with their framework-level specifications).

2. The application portion of the state of each object does not change between hook method calls (since the framework-level code that executes between such calls cannot access this portion of the state).

3. Argument & result values, and initial & final states, recorded in any element of τ must satisfy the application level post-condition of the hook method involved.

It is the third item above that plays the critical role in letting us arrive at the enriched behavior; it allows us to appeal to the enriched application-specific behavior of the hook method and this is appropriate since it is to this method, depending on the run-time type of the object in question, that the call will be dispatched. Rule **R2** below formalizes these observations. We first introduce two predicates:

$asu(\sigma, \tau, \sigma')$: '*asu*' denotes 'application portion of state is unchanged'. This evaluates to *true* if for each k ($\leq |\tau|$): the application portion of the 'initial state' recorded in $\tau[k]$, i.e., $\tau[k].is\downarrow$, is the same as the application portion of the 'final state' recorded in the most recent preceding element of τ that involves the same object as $\tau[k]$, or if there is no such preceding element then the state of this object in σ; and, for each object c, the application portion of this object in the final state, i.e., $\sigma'(c)\downarrow$ is the same as the application portion of

the final state recorded in the last element of τ that involves c (or is the same as $\sigma(c)\downarrow$ if c is not involved in any element of τ). If either of these conditions is not satisfied, this predicate evaluates to *false*.

$sccs(\tau)$: '*sccs*' stands for 'state change (in each element of τ) is consistent with the (application-level) specification (of the method)'. This predicate evaluates to *true* if the arguments, results, initial, and final states recorded in $\tau[k]$ (for each k) satisfies the post-condition of the corresponding hook method in the appropriate derived class based on the dynamic type of the object in question, i.e., $\tau[k].ob$. Note that to evaluate *sccs*, we need to be able to determine the dynamic type of our objects. We will assume that such a facility is available.

The formal definitions of these two functions is straightforward, if tedious, so we will omit them.

R2. Enrichment Rule

$$\{ p_f \wedge \tau = \varepsilon \}\ \mathcal{F} : run()\ \{ q_f \}$$
$$p_a \Rightarrow p_f$$
$$\forall(\sigma, \sigma', \tau).$$
$$[\, [p_a \wedge q_f[\sigma@pre \leftarrow \sigma\uparrow,\ \sigma \leftarrow \sigma'\uparrow, \tau \leftarrow \tau\uparrow]$$
$$\wedge\ asu(\sigma, \tau, \sigma') \wedge sccs(\tau)\,] \Rightarrow$$
$$q_a[\sigma@pre \leftarrow \sigma, \sigma \leftarrow \sigma']\,]$$

$$\{ p_a \wedge \tau = \varepsilon \}\ \mathcal{A} : run()\ \{ q_a \}$$

The first antecedent requires the framework-level specification to have been established. Next is the relation (7) between the application-level and framework-level pre-conditions. The third antecedent requires us to show, for all σ, σ', τ, that if σ satisfies the application-level pre-condition, if their framework-portions satisfy the framework-level post-condition, if the only changes in the application portion of the states of the various objects is due to calls to hook methods (i.e., $asu(\sigma, \tau, \sigma')$ is *true*), and finally if the changes in the states, recorded in τ, are consistent with the appropriate specification of the corresponding hook methods (i.e., $sccs(\tau)$ is *true*), then σ, σ', τ must also satisfy the application-level post-condition of *run()* with σ being the initial state and σ' the final state. If these three antecedents are established, then by appealing to **R2** the application developer may establish the behavior of *run*() specific to his application.

5 Case Study

The *BANK* framework, in Figures 1 and 2, consists of a refineable class *Account*, and the controller class *Bank*. We use a *Java*-like syntax. All methods are virtual and can, unless flagged as *final*, be overridden in the derived class. We use ":=" to denote assignment, reserving "=" for equality (especially in our specifications).

Account has a member variable *balance* that maintains the current balance in the account. The constructor initializes it to 0. *deposit*() and *withdraw*() are defined in the

```
class Account {
  protected int balance; // current balance
  Account( ) { balance := 0; }
  public void deposit(int amt)
    { balance := balance + amt; }
  public void withdraw(int amt)
    { balance := balance − amt; }
  public string getInfo()
    { return string(balance); }
}
```

Figure 1: Account Class

obvious manner to update *balance*. These will be redefined in the application to provide richer behaviors. *getInfo*() returns, as a string, the current balance in the account; note that *string*() returns the string representation of the value of its (integer) argument. Even *getInfo*() will be redefined in the application; in fact, it is via this redefinition that we will be able to see, so to speak, the enriched behaviors of the other operations. It is to enable such enrichments that we have defined *getInfo*() to return a result of type *string* (rather than *int*).

The *Bank*, has (only) two accounts, *a1, a2*. *tRequests* will be initialized to contain all transaction requests. *outInfo* will be initialized to empty; it will hold the results of all *printInfo* transactions.

run() reads in the transaction requests (from *tRequests*) and performs each one. A request specifies the account, type of transaction, and amount (if any). The request maybe for *deposit*, *withdraw*al, or *printInfo*. To avoid I/O issues, we use two strings, *tRequests*, initialized to contain all the requests, and *outInfo*, initialized to empty, to hold all the outputs (of *printInfo* transactions).

run() repeatedly reads the next request[10] from *tRequest* and calls *processTrans*() to process it. If the transaction is "*deposit*" or "*withdraw*", *processTrans*() invokes the corresponding operation on *a1* or *a2*, depending on the account number specified. If transaction is "*printInfo*", it appends the name of the account to *outInfo* ("+ =" is *Java*'s string append operator), then calls *getInfo*() to get information concerning the account, and appends that to *outInfo*. This means that depending on the run-time type of the account in question, appropriate information about that account –as determined by the application developer and implemented in the (re-)definition of *getInfo*() in the corresponding derived class– will be appended to *outInfo*.

Let us now consider the specification for the framework. We will focus on the key points of the specs, omitting some

[10] We omit details of *NextTrans*() and *RestTrans*() which, given a string of transaction requests, return the first transaction and the string of all remaining transactions. Similarly, *AccNo*(), *TransName*(), and *Amount*(), given a transaction request, return account number, name of the transaction ("*deposit*", "*withdraw*", or "*printInfo*"), and amount (0 for "*printInfo*"). More importantly, the duplicate *if*-code that appears in *processTrans*() is because of our explicit naming of the account objects; in practice, (references to) these objects would be stored in an array, and we would simply iterate through them.

```
class Bank {
  private Account a1, a2;
  private string tRequests; // Requests
  private string outInfo; // Results
  public final void run( ) {
    while( tRequests ≠ ⟨⟩ ) {
      nextReq := NextTrans( tRequests );
      tRequests := RestTrans( tRequests );
      processTrans( nextReq ); } }
  private final void processTrans
        ( string nextReq ) {
    acc := AccNo( nextReq );
    trans := TransName( nextReq );
    amnt := Amount( nextReq );
    if( acc == 1 ) {
      if(trans=="deposit") a1.deposit(amnt);
      if( trans == "withdraw" )
        a1.withdraw(amnt);
      if( trans == "printInfo" ) {
        outInfo + = "<"; outInfo + = "a1: ";
        outInfo + = a1.getInfo();
        outInfo + = ">";} }
    elsif( acc == 2 ) {
      if(trans=="deposit") a2.deposit(amnt);
      if( trans == "withdraw" )
        a2.withdraw(amnt);
      if( trans == "printInfo" ) {
        outInfo + = "<"; outInfo + = "a2: ";
        outInfo + = a2.getInfo();
        outInfo + = ">";} } }
}
```

Figure 2: Bank Class (part of BANK framework)

of the formal details. The specs for *deposit*() and *withdraw*() of *Account* are simple and need only say that they update *balance* appropriately.

Next consider *getInfo*():

$$post.Account.getInfo() \equiv [(balance = balance@pre) \wedge (string(balance) \preceq result)] \quad (8)$$

This states that *balance* is unchanged and the *result* returned by *getInfo*() is such that (the string representation of) *balance* is a prefix of the result. This allows the application developer to return extra information in this result; behavioral subtyping will require that the first part of this result be *string*(*balance*).

The specification for *run*() is complex but the complexity is mainly notational, involving τ and the strings *tRequest* and *outInfo*. The pre-condition of *run*() will assert that:

> τ will be ε, the empty trace; *outInfo* will be empty; the balances in *a1*, *a2* will be 0; and, *tRequests* will be a sequence of (*legitimate*) transaction requests.

More formally,

$$pre.Bank.run() \equiv [(\tau = \varepsilon) \wedge (outInfo = \varepsilon) \wedge (LegitTRs(tRequests)) \wedge (a1.balance = 0) \wedge (a2.balance = 0)] \quad (9)$$

Consider now the post-condition. Let *trsp* be the value of *tRequests* at the start of *run*(); in other words, *trsp* is *tRequests@pre*. In essence, the post-condition will then state that (when *run*() finishes):

> *tRequests* will be empty; (10.1) (10)
>
> the balance in *a1* (*a2*) will be the sum of the amounts in all the *deposit* transactions in *trsp* involving the account *a1* (respectively *a2*) less the sum of the amounts in the *withdraw* transactions; (10.2)
>
> the number of elements in τ is the same as in *trsp* and each element of τ is a call to the hook method corresponding to the particular request in *trsp*, and the account object will be the one specified in the request; (10.3)
>
> *outInfo* will consist of a sequence of 'account information' strings consisting of the name of the account and other information; the number of these strings will be equal to the number of *printInfo* transactions in *trsp*; the 'other information' in each will be the result returned by the corresponding call to *getInfo*() (which will be the *.re* component of the corresponding element of τ). (10.4)

Note that information about the argument values passed in the hook method calls has *not* been included in this specification, although it could have been. The point is that it is a choice the framework designer must make depending on what types of enrichments he expects the application developer to make. With the choice reflected in our specification, the application developer will not be able to reason about any enrichment that depend on these argument values.

In the application in Figure 3, we have defined two derived classes of *Account*. Note that in *FeeAccount*, transaction fee are imposed only on withdrawals, not deposits. In each derived class, we have redefined *getInfo*() to provide information about the new variable (*tCount, tFee* respectively). The specs of these methods are straightforward and it is easy to check that they satisfy their framework-level specs. In particular, the specification of *getInfo*() will say that the result returned will be the current balance in the account followed by the current transaction count or current transaction fee, for *TCAccount* and *FeeAccount* respectively.

Consider now the enriched behavior. Suppose *a1* is of dynamic type *TCAccount* and *a2* of dynamic type *FeeAccount*; these will be fixed at initialization. By appealing to **R2**, in particular to *sccs*(), we can strengthen our specification of *run*(). In particular, in those cases where a call to *deposit*() or *withdraw*() on *a1* is recorded, the value of *a1.tCount*, because of the richer behavior of these methods in *TCAccount*, will be increased by 1. More precisely,

$$[post.Bank.run'() \wedge (\tau[k].ob = a1) \wedge sccs() \wedge (\tau[k].hm \in \{deposit, withdraw\})] \Rightarrow$$

```
class TCAccount extends Account {
  protected int tCount; // transaction count
  TCAccount( ) { tCount := 0; }
  public void deposit(int amt)
    { super.deposit(amt); tCount++;}
  public void withdraw(int amt)
    { super.withdraw(amt); tCount++;}
  public string getInfo() {
    res := string(balance);
    res + = "trans count: ";
    res + = string(tCount); return(res); }
}

class FeeAccount extends Account {
  protected int tFee; // transaction fee
  FeeAccount( ) { tFee := 0; }
  public void withdraw(int amt)
    { super.withdraw(amt); tFee++;}
  public string getInfo() {
    res := string(balance);
    res + = "trans fees: ";
    res + = string(tFee); return(res); }
}
```

Figure 3: NewBANK Application

$$(\tau[k].fs.tCount = \tau[k].is.tCount + 1) \quad (11)$$

Note that in (11) (and in what follows), we have used the notation *post.Bank.run'*() to denote the framework-level post-condition as specified informally in (10) with the substitutions, specified in the enrichment rule **R2**, to indicate that the state, trace, etc. referred to in this assertion are the framework-level ones. Thus (11) tells us that if the k^{th} element of τ corresponds to a call to *deposit* or to *withdraw* and if the object involved is *a1*, then the value of *tCount* in the final state (immediately after the return from this call) will be one greater than its value in the initial state (that existed immediately before this call).

Similarly, because of the richer behavior of *FeeAccount.withdraw*(), and the unchanged behavior of *FeeAccount.withdraw*(), the value of *a2.tFee* will be increased by 1 following *withdraw* transactions, but not following *deposit*s:

$$\begin{aligned}&[post.Bank.run'() \wedge (\tau[k].ob = a2) \wedge sccs()] \Rightarrow \\ &\quad [\,[(\tau[k].hm = deposit) \Rightarrow \\ &\qquad (\tau[k].fs.tFee = \tau[k].is.tFee)] \wedge \\ &\quad [(\tau[k].hm = withdraw) \Rightarrow \\ &\qquad (\tau[k].fs.tFee = \tau[k].is.tFee + 1)]\,] \end{aligned} \quad (12)$$

Finally, because of the enriched behavior of *getInfo*() in both *TCAccount* and *FeeAccount*, we can argue that the corresponding strings output in *outInfo* will consist of the balance and transaction count information for the *printInfo* requests corresponding to *a1*, and the balance and transaction fee for the *printInfo* requests corresponding to *a2*. Note that this depends upon the information in (10.4), that the strings making up *outInfo* are based on the results returned by the calls to *getInfo*(), i.e., the *.re* components of the corresponding elements of τ.

$$\begin{aligned}&[post.Bank.run'() \\ &\quad \wedge (\tau[k].hm = getInfo) \wedge sccs()] \Rightarrow \\ &\quad [(\tau[k].fs = \tau[k].is) \wedge \\ &\qquad [(\tau[k].ob = a1) \Rightarrow \\ &\qquad\quad (\tau[k].re = \\ &\qquad\qquad string(balance)\,\hat{}\, \\ &\qquad\qquad "trans\ count"\,\hat{}\,\tau[k].fs.tCount)] \wedge \\ &\qquad [(\tau[k].ob = a2) \Rightarrow \\ &\qquad\quad (\tau[k].re = \\ &\qquad\qquad string(balance)\,\hat{}\, \\ &\qquad\qquad "trans\ fees"\,\hat{}\,\tau[k].fs.tFee)]\,] \end{aligned} \quad (13)$$

I.e., the result returned by a call to *getInfo* is a string consisting of the *balance* in the account followed by either the string *"trans count"* and the *tCount* in the account if the account in question is *a1*, or the string *"trans fees* and the *tFee* in the account if the account in question is *a2*.

When the information in (13) and in (10.4) are combined, we can conclude that the results that appear in *outInfo* will indeed consist of the *balance* and *transaction count* information in the case of *a1* and *balance* and *transaction fee* information in the case of *a2*, and that this information will accurately reflect the states of these account objects, based on the preceding transactions, at the time that the information was added to *outInfo*.

6 Discussion

A framework implements behavior that is common to a range of applications. An application developer can build a new application with just the incremental effort of redefining the hook methods of the framework as needed by the particular application. Our goal has been to design a reasoning approach that would be similarly incremental so that we need to reason just once about the framework. When a new application is built on the framework, our approach allows the developer to arrive at the behavior of the complete application by plugging the behavior of the redefined virtual methods into the specification of the framework.

We had to address two issues in our work. The first was ensuring that the redefined methods satisfy their framework-level specifications. The second was establishing the enriched behavior resulting from the redefinition of hook methods. Much of the past work on reasoning about OO programs has focussed on the first issue, this being the essence of behavioral subtyping. For frameworks, the second is also critical since enrichment of the behavior of the entire framework by just redefining the hook methods is the whole point of the framework based approach. In our work, the inclusion, in the specification of the *run*() method, of information about the hook method calls it makes, enables the application developer to arrive at the application-specific behavior by appealing to the richer behavior of the redefined hook methods.

In order to keep our formalism relatively simple, we had to impose some restrictions on our model of frameworks. One of these restrictions was that only *run*() was allowed to

call hook methods. As noted earlier, we could relax this restriction but if we did so, we would have to associate a trace τ_f with each method $f()$ that may invoke hook methods. This is because, given that $f()$ invokes hook methods, redefinitions of these hook methods in the application will result in the behavior of $f()$ being enriched in the same manner as that of *run*(); therefore, to arrive at the application-specific behavior of $f()$ incrementally, i.e., without reanalyzing its code, we would need information about the sequence of hook method calls it makes etc., in other words just the sort of information we record in the trace. Correspondingly, our enrichment rule would have to be generalized to apply to all template methods like $f()$ rather than only *run*(). Another restriction in our model was that all the component objects of the framework/application were explicitly *named*, such as the accounts *a1* and *a2* in the case study. In practice, frameworks may have anonymous components with only references to them being stored, perhaps in an array. As long as there are no additional complications, such as *aliasing* between these various components, our approach needs only straightforward extensions to deal with such anonymous components. Aliasing presents more serious challenges as we note below.

One important problem with our approach is the complexity of the specification of *run*(). Even in the case of our relatively simple case study, a full formal specification of *run* would have been quite involved. The complexity arises mainly because of having to deal with inductive manipulations of the trace elements. For example, if we wanted to formalize the clause (10.2), we would have to refer to all preceding transactions involving a given account when specifying the balance in the account following any particular transaction. This complexity is mostly notational and can be mitigated by devising suitable notations, perhaps using regular expressions, or notations similar to those of [3]. This should also make the approach more accessible to system developers who may not have much formal training but are likely to be comfortable with such notations.

A more interesting problem, from a conceptual point of view, is that of *framework composition*. The question is how to extend the reasoning approach to deal with applications that are composed from several, rather than a single, framework. It should be relatively straightforward to extend the notion of our traces to deal with various hook method calls made by a framework composed from two or more frameworks. The more challenging question that will have to be addressed here is that of object *references*, i.e., one object containing a reference to another object. The problems in reasoning about such systems (even in the absence of hook methods) have been discussed for example, by Meyer [21]. The problem is essentially that of *aliasing* but while aliasing has in the past been considered a sign of poor design, many modern OO systems, including those built from frameworks, use object references to great advantage. Hence the reasoning system needs to provide appropriate ways to deal with object references. One approach to object references [6, 25] is to take explicit account of object *identity*. While this approach works for a given, specific pattern of referencing between objects in a given system, it remains to be seen whether it can be extended to deal with flexible patterns of references such as are likely to be allowed in frameworks. Perhaps something like our trace of hook method calls, but one that allows us to record information about the details of these references, would prove useful.

Acknowledgments: The authors would like to thank the anonymous referees for detailed comments on the first draft of the paper.

References

[1] P. America. Designing an object oriented programming language with behavioral subtyping. In *Foundations of Object-Oriented Languages, REX School/Workshop*, LNCS 489, pages 69–90. Springer-Verlag, 1991.

[2] K. Beck and R. Johnson. Patterns generate architectures. In *Proceedings of the Eighth ECOOP*, pages 139–149, 1994.

[3] M. Buchi and W. Weck. The greybox approach: when blackbox specifications hide too much. Technical Report TUCS TR No. 297, Turku Centre for Computer Science, 1999. available at http://www.tucs.abo.fi/.

[4] R.H. Campbell and N. Islam. A technique for documenting the framework of an OO system. *Computing Systems*, 6:363–389, 1993.

[5] Apple Computer. *MacAppII Programmer's Guide.* Apple Computer, 1989.

[6] F.S. de Boer. Speaking of objects, technical report. Technical report, Utrecht University, The Netherlands, 1995.

[7] D. D'Souza and A. Willis. *Objects, Components, and Frameworks with UML.* Addison Wesley, 1999.

[8] M.E. Fayad and D.C. Schmidt. Special issue on object oriented application frameworks. *Comm. of the ACM*, 40, October 1997.

[9] G. Froehlich, H. Hoover, L. Liu, and P. Sorenson. Hooking into OO application frameworks. In *Proc. of 1997 Int. Conf. on Softw. Eng.*, pages 141–151. ACM, 1997.

[10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: Elements of reusable OO software.* Addison-Wesley, 1995.

[11] D. Gangopadhyay and S. Mitra. Understanding frameworks by exploration of exemplars. In *Proc. of 1995 Int. Wkshp. on Computer Aided Softw. Eng*, pages 90–99, 1995.

[12] D. Garlan, S. Jha, D. Notkin, and J. Dingel. Reasoning about implicit invocation. In *Proc. of Foundations of Softw. Eng. (FSE-6)*, pages 209–221. ACM, 1998.

[13] R. Helm, I. Holland, and D. Gangopadhyay. Contracts: Specifying behavioral compositions in object-oriented systems. In *OOPSLA-ECOOP*, pages 169–180, 1990.

[14] C. Horstmann. *Mastering Object-Oriented Design in C++*. Wiley, 1995.

[15] R. Johnson. Documenting frameworks using patterns. In *Proc. of 1992 OOPSLA*, pages 63–76. ACM, 1992.

[16] R. Johnson and B. Foote. Designing reusable classes. *Journal of OOP*, 1:26–49, 1988.

[17] C. Jones. *Systematic Software Development Using VDM*. Prentice-Hall, 1990.

[18] G. Krasner and S. Pope. A cookbook for using the mvc interface paradigm in smalltalk-80. *J. of Object Oriented Programming*, 1(3):26–49, 1988.

[19] B. Liskov and J. Wing. A new definition of the subtype relation. In *ECOOP*, 1993.

[20] B. Liskov and J. Wing. A behavioral notion of subtyping. *ACM Trans. on Prog. Lang. and Systems*, 16:1811–1841, 1994.

[21] B. Meyer. *Object-Oriented Software Construction*. Prentice Hall, 1997.

[22] J. Misra and K. Chandy. Proofs of networks of processes. *IEEE Trans. on Software Eng.*, 7:417–426, 1981.

[23] W. Pree. Meta patterns: a means for capturing the essentials of reusable OO design. In *Proc. of* 8^{th} *ECOOP*, pages 150–162, 1994.

[24] H. Schmid. Creating architecture of a framework by design patterns. In *Proc. of OOPSLA '95*, pages 370–384, 1995.

[25] N. Soundarajan and S. Fridella. Reasoning about polymorphic behavior. In Ege, Singh, Meyer, Riehle, and Mitchell, editors, *Proceedings of TOOLS 26*, pages 346–358. IEEE Computer Society Press, 1998.

[26] N. Soundarajan and S. Fridella. Framework-based applications: From incremental development to incremental reasoning. In W. Frakes, editor, *Proc. of* 6^{th} *Int. Conf. on Softw. Reuse*, LNCS 1844, pages 100–116. Springer, 2000.

[27] J. Warmer and A. Kleppe. *The Object Constraint Langauge*. Addison-Wesley, 1999.

[28] D. Wilson and S. Wilson. Writing frameworks: capturing your expertise about a problem domain. In *Tutorial notes, Eighth OOPSLA*. ACM Press, 1993.

Towards software design automation with patterns

Ahmet Sıkıcı and N. Yasemin Topaloğlu
Ege University, Computer Engineering Department
Bornova, Izmir - 35100 Turkey.
ahmets@egenet.com.tr, yasemin@bornova.ege.edu.tr

Keywords: design patterns, software reuse, components

Received: March 23, 2001

Design patterns are powerful design and reuse tools in software development. However current usage of patterns seem to employ a small portion of the enclosed potential. Commonly, pattern usage is limited to the manual customization of just a few legacy patterns, or documentation of existing designs. We think that a more effective use of the patterns requires standardization and formalization of the pattern utilization. In this work, we propose a formalism based on a fully generalized and purely symbolic interpretation of patterns, which focuses on the representation of reusable knowledge.

1 Introduction

The reuse of design and code has been a major issue of debates in software community, with an increasing popularity in the last decade. Most of these reuse activities take place in the *ad-hoc reuse* (Prieto-Diaz 1993) category. However, in order to maximize the benefits of reuse, as reduced time to market, better quality and decreased development costs; it is a well agreed fact that a *systematic* approach to reuse is essential. Systematic reuse has been named as a "*paradigm shift*" in software development involving reuse support in all the phases of the software development life cycle (Frakes 1994).

One of the innovative approaches is the pattern-based reuse. In 1970's Christopher Alexander, an architect, introduced the pattern notion and proposed "*a generative pattern language*" which is composed of 250 patterns of architecture solutions (Alexander 1979). Alexander proposed to use combinations of a limited number of design patterns in order to build an unlimited number of solutions in the architecture domain.

Recently, the software community has adopted the pattern concept (Coplien 1997) as design templates and since the book on Design Patterns, by GoF (Gamma et. al. 1994), the notion has gained wide popularity. According to the popular view, a design pattern does not contain any implementation and only introduces the key aspects of a design. In state of the art pattern definitions, natural language usage is dominant and automation of patterns is out of consideration. Although the state of the art design patterns provide a common language for the exchange of design solutions and a documentation tool, we believe that the pattern-based reuse can be more beneficial through the realization of a formal pattern model. In the literature, there are several projects on the formalization of patterns (Eden et. al. 1998, Smolarova et. al. 1998a), which aim to formalize the instantiation procedure of the pattern usage. In our opinion, formalization of design patterns can be applied in a broader context, including the pattern production processes.

The purpose of our work is to provide a purely pattern-based model as a standard medium for the representation of reusable design knowledge. In this paper we present a general symbolism which is suitable for expressing any kind of design knowledge, without posing any semantic restriction or implication upon its content. Our discussion is based on a pattern model and a pattern arithmetic, which resemble object-oriented concepts and involve to define several operations like inheritance for design patterns. Prior to the introduction of our approach, we will attempt to orient our discussion with respect to the concepts and issues of the object-oriented paradigm and component-based technologies. For this purpose these concepts will need to be revised over abstract models and illustrated by simple examples.

The paper is organized as follows. In Section 2, our pattern-based approach is compared with the object-oriented and component-based approaches conceptually. Section 3 covers the description of the pattern model and it is followed by the description of the pattern arithmetic in Section 4. Type checking phenomenon and the advanced message interpretation for the patterns is discussed in Section 5 and in Section 6, respectively. Conclusion is given in Section 7.

2 Patterns as Components in Reusable Software Development

We claim that the pattern-based components can have certain advantages over their object-oriented counterparts in reusable software development. Before the discussion of the possible contributions of a pattern-orientation, the general concepts of component-based reuse will be reviewed.

2.1 Component-Based Reuse

In component-based reuse, components are bound in a way that they can cooperate for performing the required functionality. Component technology requires the presence of certain descriptive measures to define how components can combine, and some protocol definition facility to define how they can communicate. In the rest of the paper, these two tasks of components will be called *association* and *communication,* respectively.

In the literature, there are several definitions given for a component. Parrish et al. defines (Parrish et. al. 1999) a component as a software artifact consisting of a service interface, a client interface and an implementation. The service interface consists of the services that the component exports to the rest of the world, the client interface consists of those services used by this component exported from other components, and the implementation is the code necessary for the component to execute its intended functionality. The sum of the service and the client interfaces makes up the interface of the component.

Prieto-Diaz (Prieto-Diaz 1993), categorizes reuse from several aspects and according to this classification, reusing components as they are, is called *black box reuse* and reusing components with modification is called as *white box reuse.* It is our opinion that it can be more practical to use an analog value like the *transparency of reuse* instead of the bivalent parameter of being black or white. In this context, an ideal *black box* reuse can be considered to be totally opaque and so called *white box* reuse can be considered to be totally transparent.

Components rely on a notion called *wrapping* which hides the implementation of the component and provides an interface for the component. The separation of the implementation and the interface is a crucial aspect of component-based reuse since it provides the necessary structure for the intended *opacity.*

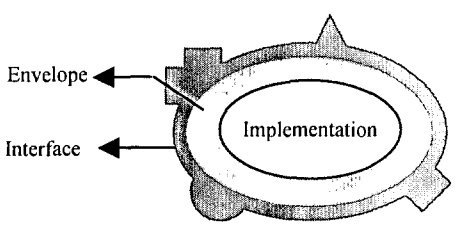


Figure-2. The abstract structure of a component.

In Figure-2, an abstract view of a component is described. The piece of code that performs the wrapping and separates the interface from the implementation is named as the *envelope.* A component's envelope manages *association* with other components at coding time and *communication* at run time. These two are the two major component roles so it can be argued that the implementation section of a component has not been referred by the component systems, resulting in poor support for *transparent reuse.* However, black box reuse is well supported by these models, theoretically over all domains since any code that can be wrapped, can be reused.

When patterns are considered, strict encapsulation is not feasible since transparency is encouraged and the distributed nature of patterns make them almost impossible to be encapsulated sufficiently by an envelope. Both the association and the communication of the pattern based components may require the implementation part to be involved. This means that no part of a typical pattern-based component can be idealized as statically private. Instead, every piece of knowledge that a pattern encloses should be theoretically reusable.

2.2 Objects vs. Patterns as Components

Parrish et al. (Parrish et. al. 1999) argue that it is disturbing to put objects and components into different categories from the beginning and that the relationships are strong and obvious. The facilities that the object-oriented paradigm grants to the component technology includes, support for the interfacing, encapsulation, easy deployment and privatization. Object-oriented type checking mechanism handles most of the component validation tasks and the communication between the components is carried out by the object-oriented messaging facility.

The conceptual roots of the object-oriented paradigm and those of the design patterns are distinct. Object-oriented programming makes use of our daily acquaintance with physical objects. On the contrary, a pattern is the representation of an idea, which is something purely within the designer's imagination (Sıkıcı & Topaloğlu 2000). This fundamental difference between the objects and patterns have many practical implications. One immediate result is that new methods should be employed to perform type checking or validation for the patterns. While type of an object can be defined by means of the properties and methods it possesses, for patterns the solution is not that straightforward.

The problem can be illustrated very well with a physical example; by the difference between defining what a car is and what a bridge is. A car can be easily visualized as an object, while a bridge is more like a pattern instance, so the type "car" can be implemented via a class but a pattern should be authored to provide a complete description of the notion "bridge". It is relatively easy to

program a computer to recognize a car, as it can be defined analytically. A car is a vehicle with four wheels, an engine, a steering wheel, an exhaust pipe, two headlights etc. On the other hand, a bridge can be in many different forms. Our discussion is not limited to the fact that a pontoon bridge, an arc bridge and a suspension bridge have very little in common, in terms of physical appearance. We also claim that if some engineer comes up with a new type of bridge that does not look like any of the known forms, most people still can tell at the first sight that it is a bridge. Although patterns do not have clear-cut boundaries and handles, human mind can easily cope with them. The kind of intelligence that lets us identify a bridge, can not be carried out with conventional type checking methods. More complex pattern recognition methodologies are needed.

The common habit prefers a different reuse medium at each level of granularity. It is argued that, unlike objects, which were too granular, components permitted reuse of implementation and related interfaces at medium granularity (D'Souza & Wills 1999). According to the current view about the patterns, pattern-based reuse takes place at an even higher level than the component-based reuse. Gamma et al. (Gamma et. al. 1994) state that design patterns describe commonly recurring structures of communicating components that solve general design problems within particular context.

However there is no theoretical restriction for the employment of patterns at a finer-grained level as object and components can be shown to be special instances of the patterns. In (Sıkıcı & Topaloğlu 1999) objects and components are viewed as basic star-shaped patterns. However in order to reach a broad band reuse medium, in terms of granularity, the model needs to be kept free from semantic content.

3 The Pattern Model

We base our discussion to pattern based components on a pattern model (Sıkıcı & Topaloğlu 1999) which is composed of , *nodes* and *links*, each having a name and a set of attributes. Every pattern is represented with a standard notation, which can be described as a special purpose semantic net. The most significant attribute of nodes and links distinguishes each as either a *hot* or a *cold element*, indicating whether the structure could have further details at that point. In Figure-3, a simple notation to denote the four types of *elements* is shown.

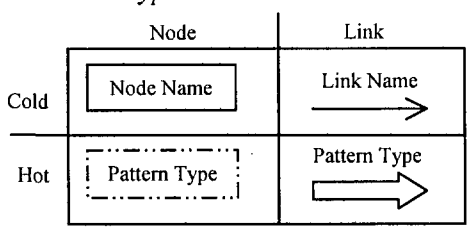


Figure – 3. Four types of elements of the pattern model

Each *hot element* can be replaced by a whole pattern, allowing the nesting of patterns. A *hot element* can be visualized as a gap in the pattern specification waiting to be filled by another pattern. *Hot elements* have restriction criteria related to the type of the replacing pattern in order to ensure the safety of the replacement. The name of a cold node, just as the type of a hot node, can be considered to be the only nesting interface of a single node.

Links have two additional properties within their nesting interface. Each link has two labels to match the nodes that it connects. These are named as the *source label* and the *destination label* as shown in Figure-4.

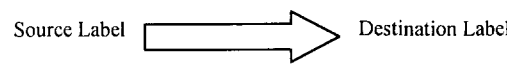


Figure – 4. The association interface of a link.

The source and the destination labels define a restriction on the relation provided by the link, by imposing type constraints on the nodes that are related. These labels make up the association interface of a link.

Our pattern model is meant to provide a base for higher systems that can associate each node and link type with the necessary semantic content, by enhancing the basic mechanical behavior.

4 Pattern Arithmetic

We consider several operations of the object-oriented technology like inheritance and aggregation as the elements of an *object-oriented arithmetic*. While automatic processing of the patterns is limited to the refinement operation, with the help of *object-oriented arithmetic*, objects can be reused to build more complex objects or components. We claim that with the suitable set of operations, a formal pattern model can be even more orthogonal than object-oriented component systems. Below we discuss a set of operations that can be executed on patterns. Together with the pattern model and the type checking mechanism, these operations constitute a *pattern arithmetic*.

4.1 Nesting the Patterns

Nesting is the most fundamental operation in the *pattern arithmetic*. Nesting is said to take place between two components when a component becomes a local property of the other. Although a component is nested inside another one through a single link, nesting a pattern into another pattern usually requires a lot of new bonds to be made. For this reason patterns naturally possess the potential to perform a stronger validation check, while attempting to nest one reuse entity into another. According to (Bosch 1997), since the designers of

reusable components are unable to make all but minimal assumptions about the context in which the component will operate, the binding of the acquaintances required by the component is often performed in an ad-hoc manner. On the other hand, patterns require that the nested pattern makes all of its association with its surrounding automatically and due to the structural representation of knowledge by the links that have been made, at least theoretically, no additional semantic contract is required.

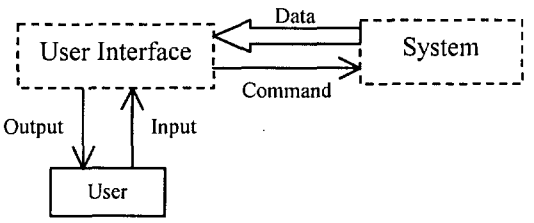


Figure-5. System Monitor Pattern.

In Figure-5, a basic pattern example is demonstrated by using the proposed pattern notation. This pattern is called *System Monitor Pattern* and it describes in a rather abstract way, how a system is monitored to a user. In the figure, the *elements*; "User Interface", "Data" and "System" are arranged as *hot elements* indicating that the details related to these parts are not known. The overall structure however is known and it indicates that the *System* directly or indirectly sends *Data* to *the User Interface* while it receives commands from it. The *User*'s interaction with the *User Interface* has been denoted with *cold elements*, indicating a static zone in the architecture.

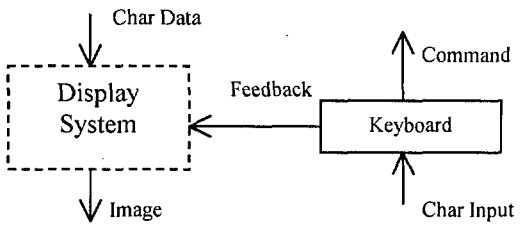


Figure-6. Basic Terminal Interface Pattern.

In Figure-6, a pattern called *Basic Terminal Interface* is given. This pattern defines a basic character-based user interface like that of a dummy terminal. If the type checking system verifies that *Basic Terminal Interface Pattern*, is an implementation of *User Interface Pattern*, we can nest it into the position of the hot *User Interface* node in Figure-5. The problem of performing type checking on patterns will be discussed in the next section.

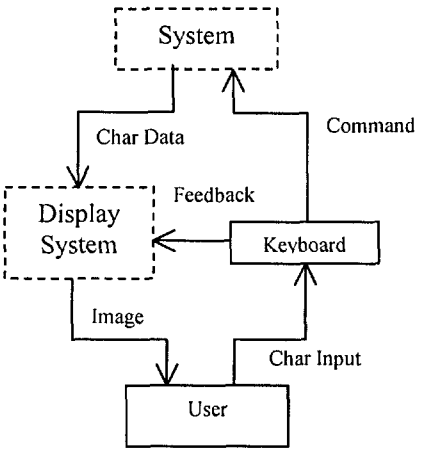


Figure-7. The resultant pattern after the nesting operation.

In Figure-7 the resultant pattern produced by the nesting operation can be observed. This new pattern describes a system monitoring pattern where the user interface accepts basic character-based input through the keyboard and displays character-based data from the system.

4.2 Abstraction of a Pattern

Sometimes it is beneficial to decrease a pattern's determinism. Büchi and Sekerinski (Büchi & Sekerinski 1997) state that non-determinism is a fundamental tool for specifications to avoid laying down unnecessary details. It is also useful in providing flexibility and purifying the reusable knowledge. To achieve this we employ an operation called *abstraction* in accordance with the conceptual framework defined by Smolarova et al. (Smolarova et. al. 1998a, 1998b).

The description of the abstraction operation is rather straightforward since physically the reverse of the nesting operation is performed. While in the nesting operation, a hot node is replaced by a pattern, in the abstraction operation a zone is replaced by a hot node. The operation is performed in two steps. In the first step a zone is marked over a pattern and in the second step that zone collapses into a single hot node.

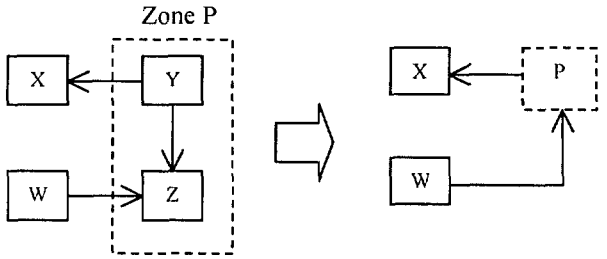


Figure-8. The abstraction operation.

In Figure-8 the operation has been illustrated on a plain example. A zone P is chosen over a pattern and all the details that are enclosed by that zone are cleared by the operation. The semantic content of the zone is now represented by a single identifier P, which is a *pattern*

type as shown in Figure-3. Under normal conditions the abstraction operation causes a loss of knowledge, resulting in a pattern that is not only more abstract but also more general.

4.3 Inheriting from a Pattern

In order to reach an understanding of inheritance for patterns, we first need a formal definition of the concept of inheritance. The conceptual definition of inheritance is very similar to the definition of reuse itself. According to (Weck 1997), in principle, inheritance is equivalent to copying and modifying source code. Obviously this definition aims inheriting from code but it can be generalized and strengthened to fit our definition of pattern inheritance. In our view, the most suitable definition of inheritance is that, it is the act of basing the construction of a new entity onto an existing *parent entity*, and preserving all the *key aspects* related to the parent entity.

From the pattern-oriented point of view, conceptually, the *key aspect* to be preserved is the knowledge content of the pattern. If we declare that a pattern inherits from another, this means that all the knowledge contained in the *parent pattern* is also possessed by the child *pattern*. Physical interpretation of this definition is that, when inheriting from a pattern, all the existing nodes and links should be preserved. In Figure-9, a new pattern, which was created by extending the *Basic Terminal Interface Pattern* is depicted. In this pattern, keyboard input has been preserved, and a mouse input has been added to the system.

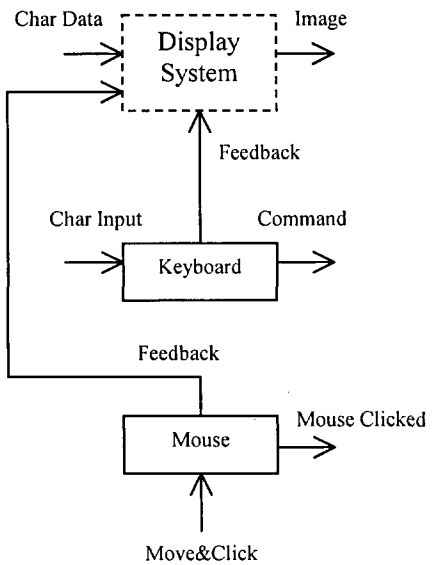


Figure-9. Extended Basic Terminal Interface Pattern.

We need to mention that inheritance does not always produce a more specific pattern in the semantic sense. In fact inheritance may change the generality and abstractness of a pattern in either direction. The semantic effect of inheritance depends on the semantic context which is imposed by higher systems and the physical *key aspects* that are preserved in the inheritance operation.

4.4 Superposition of Patterns

Superposition is probably best described in Christopher Alexander's words : "It is also possible to put patterns together in such a way that many patterns overlap in the same physical space: the building is very dense; it has many meanings captured in a small space; and through this density, it becomes profound" (Alexander et. al. 1977). We define *superposition* as a union of all the nodes and links of the two patterns. In this operation, two patterns are joined in a single scope, sharing their common entities and relations. Recurring elements are not repeated, but are re-referenced in the resultant pattern.

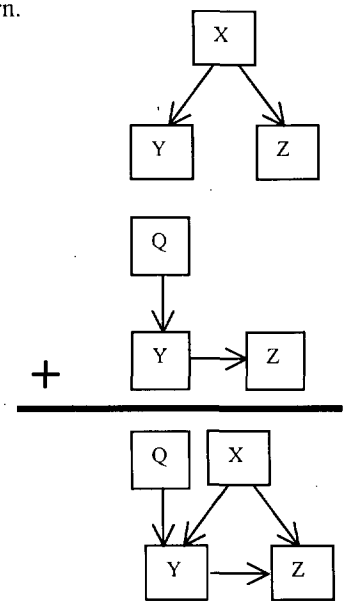


Figure-10. The *superposition* of patterns.

Figure-10 illustrates the superposition operation in an abstract way. Superposition is suitable when two patterns are physically different but are known to be describing the same phenomenon. In that case, two loose descriptions of the same design can be superposed into one solid description. Superposition can also be viewed as *multiple inheritance* for the patterns and it can be used to combine two ideas into one. In contrast with object-oriented multiple inheritance, superposition operation does not aim to produce specific instances of the operands. In parallelism with the pattern inheritance, preserving the enclosed knowledge is the main focus of superposition.

5 Type Checking

5.1 Component-Based Validation Check

In the idealized definitions of a component, a validation check is promised. For example Kozaczynski (Kozaczynski 1994) states that a component conforms to and provides a set of interfaces in the context of well-

formed system architecture. However, the situation of the existing component systems in terms of integrity validation seems to be quite problematic. (Schreyjak 1997) states that the replacement and addition of components are not supported in an adequate way by today's component systems due to the lack of the integrity checks, that can verify that all the required interfaces and preconditions are met.

Two innovative attempts to solve the validation problem of component usage will be introduced briefly. One is called a "port and link framework" (Wang & MacLean 1999) and the other is a "reuse contract" (DeHondt et. al. 1997).

A "*port and link*" framework is proposed to support component assembly for distributed systems in (Wang & MacLean 1999). In this model, a functional boundary is described for the component by four lists, namely services a component provides, services a component requires, events a component observes and events a component generates. In Figure-11, the knowledge content of such a definition has been illustrated. This model clearly extends the classical approach with event generating and observing. The local services and generated events are called *ports* and they correspond to the *service interface.* Observed components and called services are called as *links* and they correspond to the client interface. In such a system, validation of the association of components is implemented by matching the links of one component with the ports of the other and vice versa.

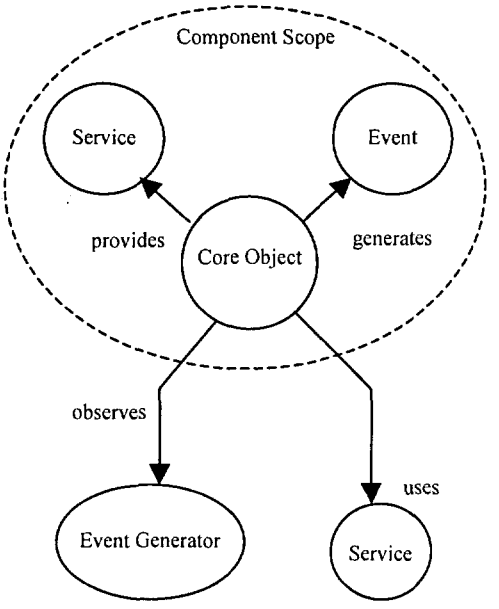


Figure-11. Content of an *architectural component definition.*

However (DeHondt et. al. 1997) demonstrates the insufficiency of a *port and link* approach and supports a "*reuse contract*" for the adequate definition of a component interface. A reuse contract keeps a track of the services that call each other within and across the component boundaries, providing an overview of the component's interactions. Figure-12 depicts a representation of a reuse contract.

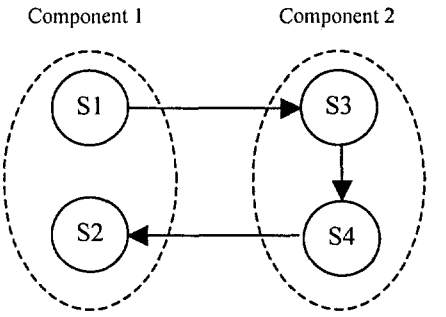


Figure-12. A reuse contract.

Since modifying or extending the system would probably break some of the existing links and construct new ones, some "*reuse operators*" were introduced to keep track of the modifications. Each "*derived contract*" is represented by a series of reuse operators and the association of the components across the derived contracts are validated by a consistency check between those series.

A reuse contract can be considered to be an instance of an architectural pattern and seems to be useful in its domain. However, the need to introduce the reuse operators suggests that the issue of extending the reuse contracts is problematic. It is an obstacle that a *common base contract* is a minimum requirement for checking the validity of the association of two components. In our opinion such a two phased method with its descriptive and procedural parts is not as promising as a purely descriptive type checking method.

5.2 Pattern-Based Validation Check

According to our approach, theoretically a separate structural type checking method can be defined for any semantic aim that can be attributed to type checking. If *association* is considered as in the case of component models, naturally type checking should focus on the external links of the pattern. The same is true for the nesting operation. A pattern's type according to association and nesting is defined by listing the incoming and outgoing links of the pattern.

Interface: +a, +b, -c, -d

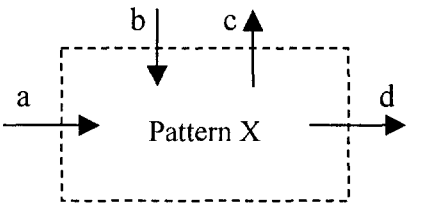


Figure-13. Natural association and nesting interface.

As indicated in Figure-13, the pattern can be considered as if it has been totally *abstracted* into a *hot node,* and its interface can be defined just as in the *port and link* method for components. Regardless of the implementation of the Pattern X, if its *elements* have the required external relations, it belongs to the equivalence class defined by the interface. This interface is called the *natural association and nesting interface* because it can be overridden at a higher layer in order to load additional semantic meaning to the pattern's type. This can be possible by defining a *pattern interface* (Sıkıcı & Topaloğlu 1999) explicitly. In that case, instead of matching the external links of the pattern with the links of the context, an explicitly specified portion of the pattern is tried to match with the context.

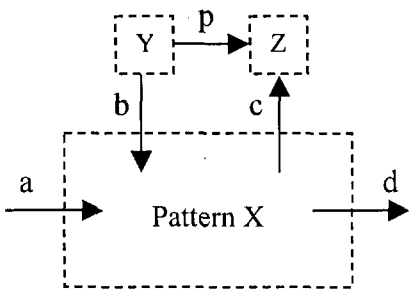


Figure-14. Explicit association and nesting interface.

In Figure-14, a basic example of an explicit association and nesting interface has been defined. The pattern that is shown in Figure-14 inherits from that of Figure-13. This time however an additional constraint should be met for a nesting or association to be possible. The source of the link "b" is linked to the target of link "c" with a relation "p". Only if such a relation is present within the context, a nesting of the pattern X into that context is allowed.

6 Advanced Message Interpretation

As it has been stated in Section 3.1, *communication* is one of the two main issues that component systems have to deal with. Most of the state of the art component systems rely heavily on object-oriented message sending mechanisms for *communication.* Little has been added by the component technologies on this issue. *Communication* from an object-oriented point of view is called "*message sending*" as objects are known to communicate by sending messages to each other. An object-oriented message is an invocation of an operation. It consists of the name of the operation, the identity of the recipient, and a set of arguments (D'Souza & Wills 1999). An object's method or operation is simply a procedure that has been defined with respect to that object's scope and a message is a procedure call. Below we give a representation of a message sending syntax in pseudo-code.

```
myRecipient.myOperation(arg1,arg2,...,argN)
```

Object-oriented components interpret messages in a rather straightforward way. The class definition contains code for the operations the object "understands", that is, the operations for which there are specifications, and hence clients could expect to send it (D'Souza & Wills 1999). The notions of object-oriented programming like polymorphism, overriding and overloading play the chief roles in the interpretation of the message. The tasks that are attributed to these phenomena, can be summarized as the smart matching process of the incoming messages with the defined operations, by using the recipient's name, the operation's name and the list of arguments.

Sometimes due to a wrapper object or a layer, the operation name or the order of arguments may be changed before the message is transferred to the private part of the component. An example for this case is a system called LayOm (Bosch 1997), which employs layers around classical objects in order to interpret messages. Further interpretation mechanisms are used by the components of distributed architectures where the main initiatives are providing compatibility and preserving security.

Pattern-based component model brings about the idea that a more intelligent message interpretation can be possible by using the knowledge enclosed by a pattern. One method takes the advantage of a parallelism between a syntax diagram and a pattern. The method that will be introduced here is based on the fact that, through a series of knowledge-preserving transformations, pattern graphs can be transformed into syntax diagrams (Sıkıcı & Topaloğlu 1999).

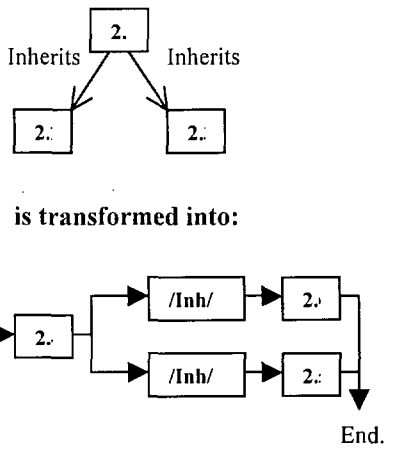


Figure 15. Transforming a pattern into a syntax diagram.

Figure-15 depicts such a transformation. The possibility of such a transformation shows that patterns can parse complicated messages that have a structural form, just like the sentences of a language. Each pattern can be seen as defining a little language through which it interprets the incoming messages.

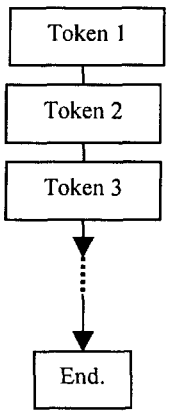


Figure-16. A pattern-based message.

In Figure-16 such a complicated message is represented. A pattern-based message is a train of tokens (Sıkıcı & Topaloğlu 2000). While it is being parsed, each token can be received by a separate node or link of the pattern. The receipt of each token by an *element* of the pattern is an equivalent of an object-oriented message transfer. The pattern's structure, and the message's content mutually determine a scenario that defines in what order the messages will be received by the nodes.

7 Conclusions

In this paper we have discussed the benefits and plausibility of using patterns as reusable components. We outlined an approach for defining the pattern-based components relying on a formal pattern model. The model does not deal with the semantic content of the patterns but it supports pattern-originated concepts from a higher level, with constructs to define names, types, attributes, relations and non-determinism.

Our model has been enhanced with a set of basic constructive operations, forming a *pattern arithmetic*. *Pattern arithmetic* enables to reuse the existing patterns in order to build new patterns.

In our discussion we made use of the object-oriented and component-based phenomena and tried to inquire their equivalents in the pattern paradigm. Patterns are problematic in certain areas like black box reuse, however concepts like inheritance can be transferred to the pattern terminology with minor adaptations. In certain aspects like validation, patterns perform better than objects and components, due to their descriptiveness and transparency. Although at this point in our work, it is hard to make statements about the message interpretation issue, it is evident that, patterns are promising in the sense that semantically complex messages can be interpreted by the patterns.

8 References

[1] Alexander C. (1979) *The Timeless Way of Building.* Oxford University Press, New York.

[2] Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. & Angel,S. (1977) *A Pattern Language.* Oxford University Press, New York.

[3] Bosch J. (1997) Adapting Object-Oriented Components. *Proceedings of the Second International Workshop of Component-Oriented Programming (WCOP'97)*, Jyvaskyla, Finland, June, 1997.

[4] Büchi M. & Sekerinski E. (1997) Formal Methods for Component Software: The Refinement Calculus Perspective. *Second International Workshop of Component-Oriented Programming (WCOP'97)*, Jyvaskyla, Finland, June,1997.

[5] Coplien J. O. (1997) On the Nature of The Nature of Order. Lecture in The Chikako Patterns Group, 5th August 1997 [notes taken by Brad Appleton]. http://www.enteract.com/~bradapp/docs /NoNoO.html.

[6] DeHondt K., Lucas C. & Steyaert P. (1997) Reuse Contracts as Component Interface Descriptions. *Second International Workshop of Component-Oriented Programming (WCOP'97)*, Jyvaskyla, Finland, June, 1997.

[7] D'Souza D. & Wills A. C. (1999) *Objects, Components and Frameworks with UML.* Addison Wesley.

[8] Eden A., Gill J.Y., Hirshfeld Y. & Yehudai A. (1999) Towards a Mathematical Foundation for Design Patterns. Technical report 1999-004, Department of Information Technology, Uppsla University

[9] Frakes W. (1994) Systematic Software Reuse: a paradigm shift. *Proceedings of Third International Conference on Software Reuse,* IEEE Computer Society Press, p 2-4.

[10] Gamma E., Helm R., Johnson R. & Vlissides J. (1994) *Design Patterns, Elements of Reusable Object Oriented Software.* Addison Wesley.

[11] Kozaczynski W. (1999) Composite Nature of Component. *2nd International Workshop on Component-Based Software Engineering, The 21st International Conference on Software Engineering (ICSE 99)*, Los Angeles, CA, USA, May 1999.

[12] Parrish A., Dixon B. & Hale D. (1999) Component-Based Software Engineering: A Broad Based Model is Needed. *2nd International Workshop on Component-Based Software Engineering, The 21st International Conference on Software Engineering (ICSE 99)*, Los Angeles, CA, USA, May 1999.

[13] Prieto-Diaz R. (1993) Status Report: Software Reusability. *IEEE Software*, May 1993, p 61-66.

[14] Schreyjak S. (1997) Coupling of Workflow and Component-Oriented Systems. *Second International Workshop of Component-Oriented Programming (WCOP'97)*, Jyvaskyla, Finland, June 1997.

[15] Sıkıcı A. & Topaloglu N.Y. (2000) Towards Building with Design Patterns. *The Fifteenth International Symposium on Computer and Information Sciences(ISCIS XV)*,Istanbul, Turkey, October, 2000, p 522-529.

[16] Sıkıcı A. & Topaloglu N.Y. (1999) Design Patterns as Building Blocks in Reusable Software Design. *The Fourteenth International Symposium on Computer and Information Sciences(ISCIS XIV)*, Kuşadası, Turkey, 1999, p 247-256.

[17] Smolárová M., Návrat P. & Bieliková M. (1998a) Abstracting and Generalising with Design Patterns. *The 13th Int. Symposium on Computer and Information Sciences (ISCIS'98)*, U. Gudukbay et al. (Eds.) IOS Press, 1998, p 551-558.

[18] Smolárová M., Návrat P. & Bieliková M. (1998b) A Technique for Modelling Design Patterns. *Knowledge-Based Software Engineering - JCKBSE'98*, Smolenice, Sept., 1998. P. Návrat and H. Ueno, eds, IOS Press, Amsterdam, p 89-97.

[19] Wang G. & MacLean H.A. (1999) Software Components in Contexts and Service Negotiations. *2nd International Workshop on Component-Based Software Engineering, The 21st International Conference on Software Engineering (ICSE 99)*, Los Angeles, CA, USA, May 1999.

[20] Weck W. (1997) Inheritance Using Contracts & Object Composition. *Second International Workshop of Component-Oriented Programming (WCOP'97)*, Jyvaskyla, Finland, June 1997.

Support of knowledge management in distributed environment

Ján Paralič*, Marek Paralič**, Marián Mach*
*Dept. of Cybernetics and Artificial Intelligence, **Dept. of Computer Science,
University of Technology in Košice, Letná 9, 042 00 Košice, Slovakia
Phone: +421 95 602 4128, Fax: +421 95 62 535 74
[paralic,paralicm,machm]@tuke.sk

Keywords: knowledge management, knowledge modelling, mobile agents, agent collaboration, e-Democracy

Received: March 11, 2001

An original approach to support of knowledge management within an organization is presented in this article. This approach has been designed, implemented in form of the KnowWeb system and tested on various pilot applications. Special attention is paid to organizations with distributed environment. For this purpose an experimental system for support of mobile agents that combines the power of high-level distributed programming with the mobile agent paradigm has been proposed and is presented here as well. Finally, experiences from two of the KnowWeb pilot applications as well as further application possibilities in the area of e-Democracy are sketched.

1 Introduction

Knowledge can be simply defined as actionable information (Tiwana 2000). That means (only) relevant information being available in the right place, at the right time, in the right context, and in the right way.

The knowledge life cycle defined in (Nonaka & Takeuchi 1995) hinges on the distinction between *tacit* and *explicit* knowledge. Explicit knowledge is a formal one and can be found in documents of an organization: reports, articles, manuals, patents, pictures, images, video, sound, software etc. Tacit knowledge is personal knowledge given by individual experience.

A considerable amount of explicit knowledge is scattered throughout various documents within organizations. It is quite often that this knowledge is stored somewhere without being retrieved and reused any more. As a result, most knowledge is not shared and is forgotten in relatively short time after it has been invented or discovered. Therefore, it has become very important for advanced organizations to make the best use of information gathered from various document sources inside companies and from external sources like the Internet. On the other hand, tacit knowledge of the documents' authors provides important context to them, which cannot be effectively intercepted.

Knowledge management (KM) generally deals with several activities that appear in knowledge life cycle (Abecker et al. 1998): identification, acquisition, development, dissemination (sharing), use and preservation of organization's knowledge. From these activities, dissemination (sharing) is crucial. Knowledge that does not flow does not grow and eventually ages and becomes obsolete and useless. By contrast, knowledge that flows, by being shared, acquired, and exchanged, generates new knowledge (Borghoff & Pareschi 1998).

Our approach to knowledge management supports most of the activities mentioned above. Based on this approach, KnowWeb[1] toolkit has been designed, implemented and tested on 5 pilot applications. Firstly, it provides tools for capturing and updating of tacit knowledge connected with particular explicit knowledge inside documents. This is possible due to ontology, which is used for representation of organization's domain knowledge. Section 2 describes in more detail domain knowledge modelling in general and our approach to it in particular.

Secondly, intelligent retrieval is enabled making use of both kinds of knowledge linked together within the organisational memory. How organisational memory in our approach is organised and what functionality it offers presents section 3.

As next, efficient communication and support for distributed groups to share knowledge and exchange information efficiently is provided (section 4). For these purposes an experimental framework for mobile agents has been designed and implemented and is introduced in section 5. Experiences from two of the KnowWeb pilot applications as well as further application possibilities in the e-Democracy context are sketched in section 6.

[1] EC funded project ESPRIT 29065 "Web in Support of Knowledge Management in Company (KnowWeb)"

2 Domain knowledge modelling

2.1 General

Theoretical foundations for the research of domain modelling can be found in the works (Chandrasekaran et al. 1999; Gruber 1993; Wielinga et al. 1997), and others on ontologies and knowledge modelling. Ontology is a term borrowed from philosophy where it stands for a systematic theory of entities what exist. In context of knowledge modeling, Gruber introduced the term ontology as a set of definitions of content-specific knowledge representation primitives consisting of domain-dependent classes, relations, functions, and object constants. The ontology represents formal terms with which knowledge can be represented and individual problems within a particular domain can be described. Ontology in this sense determines what can 'exist' in a knowledge base. Chandrasekaran understands ontology as a representation vocabulary typically specialised to some domain. He suggests basically two purposes why ontologies may be used:

- to define most commonly used terms in a specific domain, thus building a skeleton,
- to enable knowledge sharing and re-using both spatially and temporally - see also (Motta & Zdrahal 1998).

Ontology with syntax and semantic rules provides the 'language' by which KnowWeb(-like) systems can interact at the *knowledge level* (Newell 1982). Ontology allows a group of people to agree on the meaning of few basic terms, from which many different individual instantiations, assertions and queries may be constructed. Once there is a consensus on understanding what particular 'words' mean, knowledge represented by these words can be adapted for particular purposes. Knowledge must be defined unambiguously because different people in the organisational structure of an organization need to use them with the same meaning. Thus, it is possible to re-use and share the knowledge thanks to understanding of its representation.

Common understanding of the meaning of notions used in a given domain (the understanding may be domain-specific) results in the definition of *concepts*. Concepts are more or less abstract constructs on which a relevant part of the world is built, or better, which can be used to describe this relevant part of the world. Since concepts can differ in their character, several types of concepts, namely *classes*, *relations*, *functions* or *procedures*, *objects*, *variables* or *constants*, can be distinguished. These primitive constructs can be represented differently in different applications but they have the same meaning in all applications – i.e. when someone wants to communicate with somebody else, he/she can do it using constructs from the ontology of shared concepts.

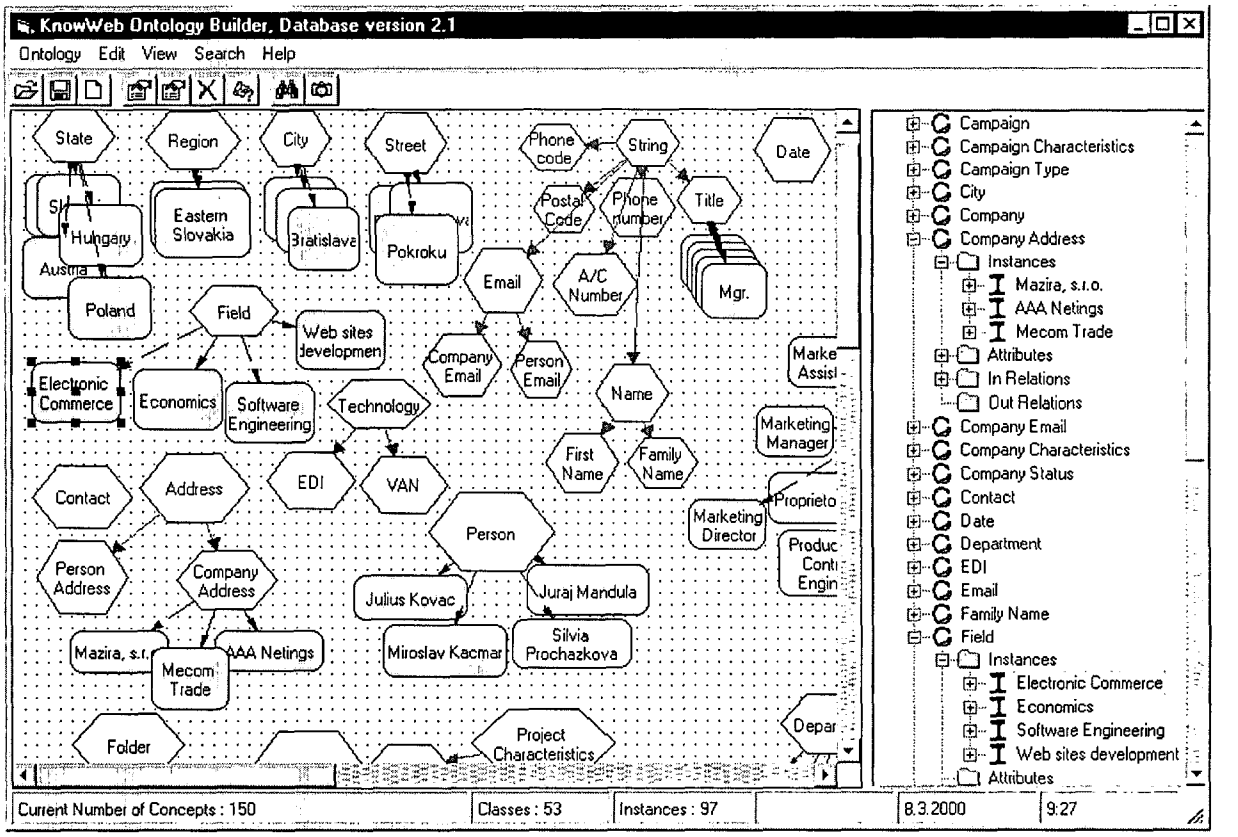


Figure 1. Sample of a domain knowledge model (ontology).

Ontology with syntax and semantic rules provides the 'language' by which KnowWeb(-like) systems can

The concepts create usually a very complicated hierarchical, network (or tree)-like structure. However,

even a complex structure covers only a specific part of the world, e.g. a narrow world of an organization and its activities. This structure models the world from a certain point of view. And here emerges the notion from the title of this section – *Domain Knowledge Modelling*, as the concepts are usually highly domain-dependent or subject-dependent, and can be meaningfully used only in the frame of the particular domain. In other words, what is acceptable and important, for example, for a property management company may be not suitable for a company dealing with distance delivery of educational courses.

2.2 Our approach

Based on the needs analysis of several pilot applications, two types of concepts have been identified as necessary and satisfactory as well. They can be either generic (type *class*) or specific (type *instance*). Both of them have attributes. Concepts and relations among them are used to construct domain model. Formally, a relation in KnowWeb is an oriented link between two concepts. Two basic types of relations can be distinguished: *subclass_of* for relations between classes and *instance_of* for relations between classes and their instances. These two relation types enable inheritance of attributes and their values. The inheritance is an important mechanism for the development of a hierarchical ontology. Also multiple inheritance is supported i.e. a class concept can inherit its attributes from several parent class concepts. Figure 1 represents an example - a very small part of a domain knowledge model.

What shall be included in the domain model? The simple but vague answer is - everything what is relevant and important to describe a particular domain. In case of a company such model may conceptually describe the company specific concepts, such as its activities, projects, customers, employees etc., as well as relations among these concepts.

Each organization has some knowledge already gathered in the form of various databases and/or documents containing information about various technologies, products, customers, suppliers, projects or employees. Each company has usually some internal procedures how to perform specific tasks. Simply said – knowledge exists in an established environment. This knowledge is traditionally called organization's *goals* and *know-how*. From the knowledge modelling perspective a repository of know-how, goals etc. may be addressed as an *organisational memory* or a *corporate memory*.

3 Organisational memory (OM)

3.1 Conceptualisation and retrieval

The KnowWeb system enables author of any document to store his/her background knowledge together with the document attaching the relevant concepts from ontology – i.e. document is stored with its context. Context can be attached to the document as a whole or to a specific part(s) of a document called text fragment(s). *Text fragment* is a continual part of text within the document (e.g. a sentence or paragraph). In present version, the KnowWeb toolkit is able to process MS Word documents where no restrictions are given on text fragments and HTML documents where text fragment cannot cross any HTML tag. In order to cope with these documents the system provides a set of tools. They differ in their functionality but together they enable documents' authors and users to manage knowledge in a company in an easy and user-friendly way.

First, in order to place a document in the organisational memory, it is necessary to attach context knowledge (i.e. a piece of tacit knowledge) to it. This context can be in the form of a conceptual description (CD). By conceptual description is meant a set of links between a document (or its marked text fragments) and concepts in the domain knowledge model. Conceptual descriptions will enable to refer not only to explicit knowledge contained in the document but also to make use of tacit knowledge. In such a way easy sharing of knowledge in the future is enabled. The CD links can be created manually or semi-automatically User can select a text fragment and link this fragment to the domain knowledge model. The linking can be done directly to ontology concepts or to some template (for semi-automatic linking). Association links are of many-to-many type. It means that it is legal to link a document (or a text fragment) to several concepts and obviously, a concept can be linked to several documents (and or text fragments).

When a document with its description is available (after manual or semi-automatic linking within the KnowWeb system or after receiving the document with its description from outside and consecutive automatic linking), it can be incorporated into the organisational memory represented by a KnowWeb server. The description of this document is stored in the KnowWeb server as well. Another possibility is to store a conceptual description of a document without storing the document (the document can be located on other KnowWeb server in a distributed company or somewhere else, e.g. on Internet). If an author is not satisfied with the conceptual description of a document stored in the organisational memory, he/she has the possibility to modify it and subsequently upload the document with the modified description into the organisational memory.

The aim of storing document in the organisational memory is to access the right knowledge in the right time or situation. In order to express requirements on documents, which should be retrieved from the organisational memory, the user has to formulate a query. To formulate a query he/she can use concepts (or their attributes) from the domain model. They can be composed into a more complex structure using various operators (e.g. logical connections). In general, the concepts specified in the query will be used to search conceptual descriptions of documents.

3.2 Implementation

The *Conceptualisation tool* (*CT tool*) is built on the top of three modules, namely *DocView*, *OntoView*, and *TemplateEditor* (see Figure 2). The *CT tool* serves as an "envelope" or integrator for these modules. The *CT tool* is able to deal with two types of documents. First of all, it is possible to open a new document, which will be linked to the corresponding concepts in the domain knowledge model, and a copy of this document will be stored on the local KnowWeb server. Two kinds of association links are available for this kind of documents.

- Association links between the whole document and (a) specific concept(s) from the domain knowledge model
- Association links between a text fragment of the document and one or more concepts from the domain knowledge model

The same may be done for documents that are already stored on the local KnowWeb server and can be already (at least partly) linked to some concepts from the domain model. Existing association links can be edited/removed and/or new links can be added.

Another sort of documents are those, which will not be stored locally (e.g. documents accessed by remote retrieval function) and in principle can be located on any web server on Internet. We refer to the documents of this kind as *referenced documents*, because only their URLs are provided and serve as references to such documents. For this kind of documents, it is possible to define only the association links between a document as a whole and (a) concept(s) from domain knowledge model. The same applies for other then MS Word and HTML types of documents, where text fragments cannot be defined.

In order to create association link(s), the user uses the drag-and-drop functionality of the *CT tool*. It means that it is possible to "grab" a concept from the domain model, drag and drop it on a highlighted text fragment or header of a document currently opened in *DocView* window.

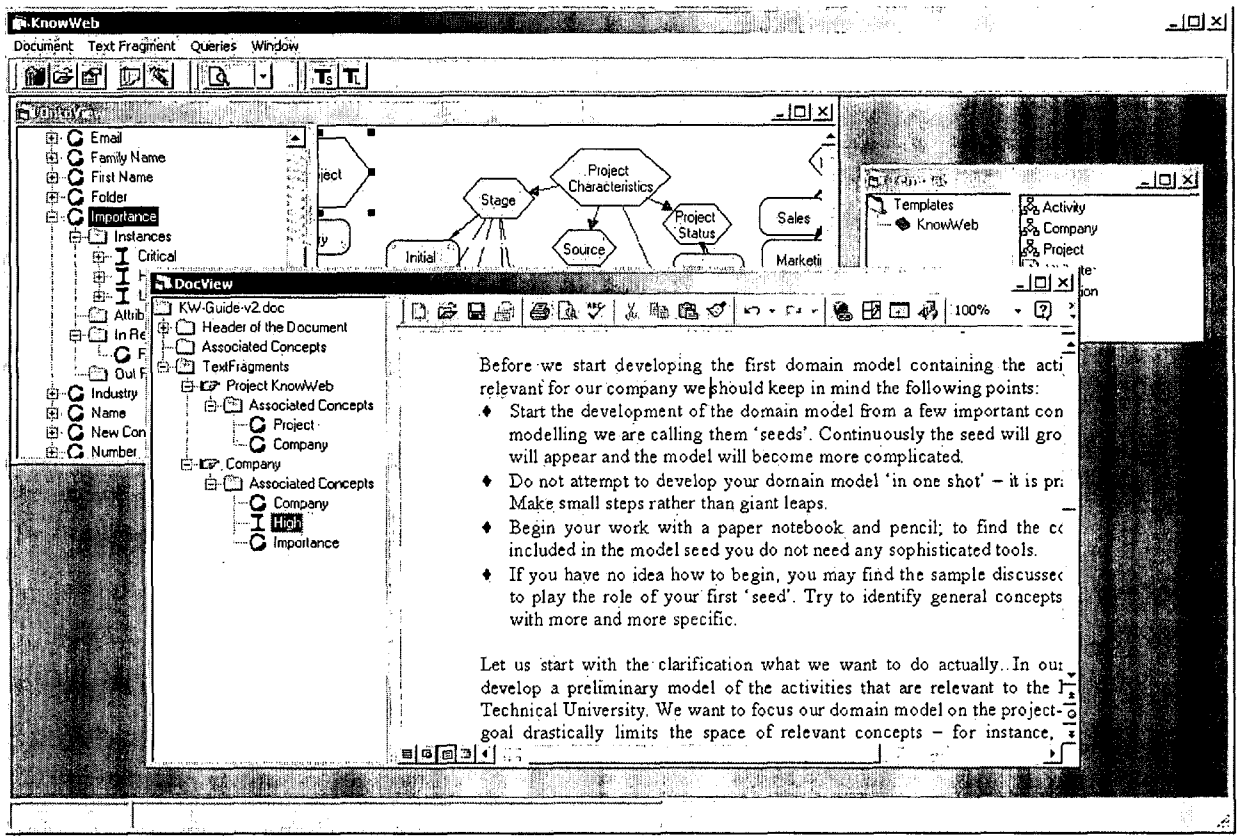


Figure 2. KnowWeb client interface to organizational memory.

The purpose of the *DocView module* is to provide users with possibility to preview documents with option of highlighting those text fragments, which are linked to the domain knowledge model. It also supports definitions and modifications to the annotation structure of the document, which results in the modified definitions of text fragments. Assignment of specific attributes to these text fragments is also possible.

The analysis of pilot applications by our industrial partners (e.g. application in the retail sector) identified a special requirement for "automatic" conceptualisation of documents with rigid structure (e.g. a structure of daily reports in a retail chain generated each day in each shop is fixed). This requirement is fulfilled by so-called "quiet mode" of the *CT tool*. In this case no visual component is

started and the documents are linked automatically to predefined concepts.

TemplateEditor module is a good supporting tool for both "quiet" as well as manual modes of *CT tool*. The purpose of this module is to give the users a tool for definition of special rules for selection of appropriate concepts. Linking a document (or its text fragment) to a template means that association links from this document (or text fragment) to all concepts resulted after application of particular template to present ontology will be created automatically.

In general, all documents have (possibly empty) headers, which represent various properties of the documents. For example, they can contain information regarding the document name, date and time of its creation, authors of documents, comments, etc. The set of attributes applicable for a header definition is application-dependent. Some properties may be compulsory while others are optional. Within a template three basic kinds of concept references are available.

- A concept can be *directly referenced* (only target concept must be given).
- A concept can be a result of an *"if – then" rule* application to values of document properties.
- A concept can be *referenced by a document property*. In this case, the name of a concept is determined by the value of a document property.

The first two kinds of references are especially useful for manual linking, but can be used for automatic linking as well. The last type is well suited for the purposes of the quiet mode of the *CT tool*, i.e. for automatic linking of whole documents. In this particular case, the target concepts are given indirectly and depend on the document property values as given in the document header.

Moreover, means for automatic creation of instances in the domain knowledge model (in quiet mode) are provided as well. This was another user-defined requirement.

4 Agent-based support for distributed organisational memory

As already mentioned in previous sections one of the most important requirements for success of OM is a support for the distributed environment of an organization. The OM should be flexible enough to fit different organization network settings and also it should remain open to the external sources such as Internet. The

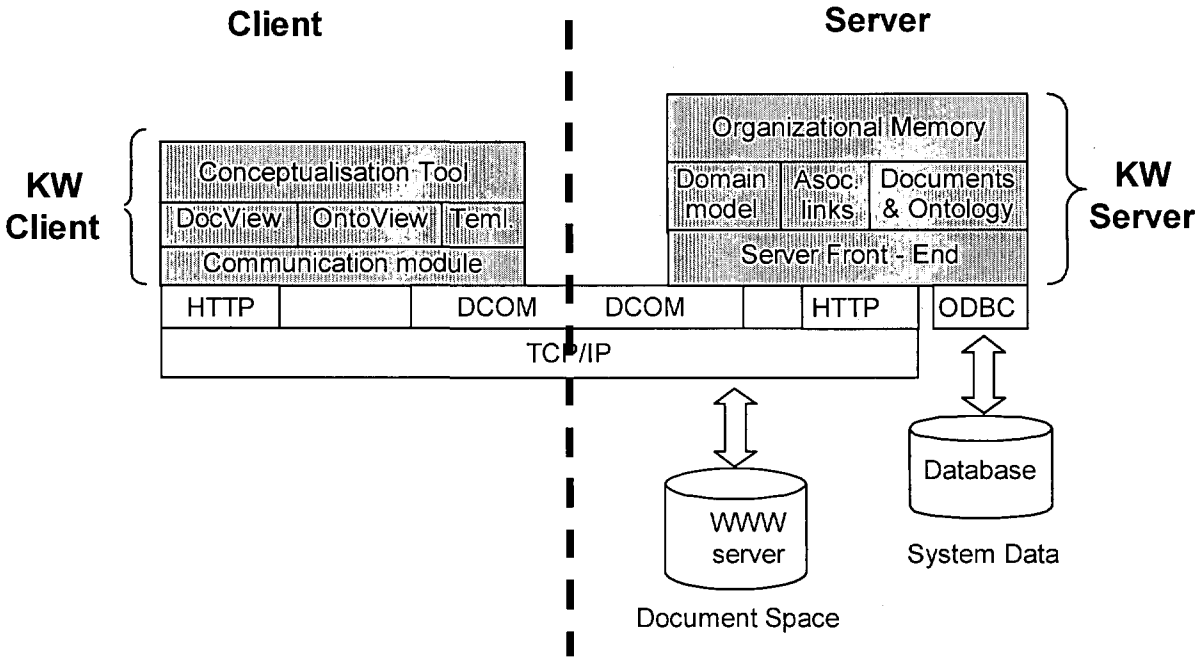


Figure 3. Distributed architecture of the KnowWeb environment – intranet solution.

state-of-the-art solution to the first problem lies in the utilisation of the distributed objects. The most popular architectures for supporting distributed objects are *CORBA* (Common Object Request Broker Architecture) and *DCOM* (Distributed Component Object Model) (Pritchard 1999).

Objects in these architectures can capture the high-level logic (so-called business logic) of a distributed application and are accessible for processes outside the computer running them. Present implementations of these standards are based on commonly used communication protocols (e.g. TCP/IP). In the research

prototype version of the KnowWeb toolkit a 3-tiered architecture was designed and implemented. We are distinguishing the following tiers:

- data source tier represented by the relational database,
- middle tier (called 'Server Front-End') that offers services to clients independently from the database implementation, and
- KnowWeb clients.

The relations among these three tiers are depicted in Figure 3. The document store and OM functionality is mapped to the KnowWeb server. Server offers key services – (i) acquisition and modification of the domain knowledge model state, (ii) association links of the documents, and (iii) retrieval in the document or ontology (domain model) space. All these operations are at the implementation level carried out as operations upon a relational database.

Solution as described above fulfils the requirement of flexibility in the corporate intranet settings and partially addresses the openness to the external sources. KnowWeb server is usually separated on an own local area network, contains (mostly one) domain knowledge model and typically serves many clients. The KnowWeb client can communicate with more than one KnowWeb server, potentially outside of the intranet scope.

implemented. Mobile agent (MA) technology is perhaps the most promising paradigm that supports application design for dynamically changeable, networked environment with distributed information and computation resources. The most significant properties of mobile agents regarding their role are *autonomy* and *mobility* (Rothermel *et al*, 1997). MAs are autonomous because of their capability to decide which locations in a computer network they will visit and which actions they will take in these locations. This ability is embodied in the source code of every agent (implicit behaviour) or by the agent's itinerary, which can be dynamically modified (explicit 'orders'). Mobile agents can move between different locations in a network. A location is the basic environment for execution of mobile agents' code and therefore an abstraction of the underlying computer network and operating system.

Usual benefits of mobile agents are (i) reduced network load, (ii) overcome of the network latency, (iii) encapsulation of different protocols, (iv) asynchronous and autonomous execution, and (v) natural heterogeneity (Harrison *et al.* 1995; Lange & Oshima 1999). We claim that robust and scaleable OM systems can profit from these features; this will be especially visible in the companies with a world-wide and/or distributed structure. Modified distributed structure of the KnowWeb system is depicted on Figure 4.

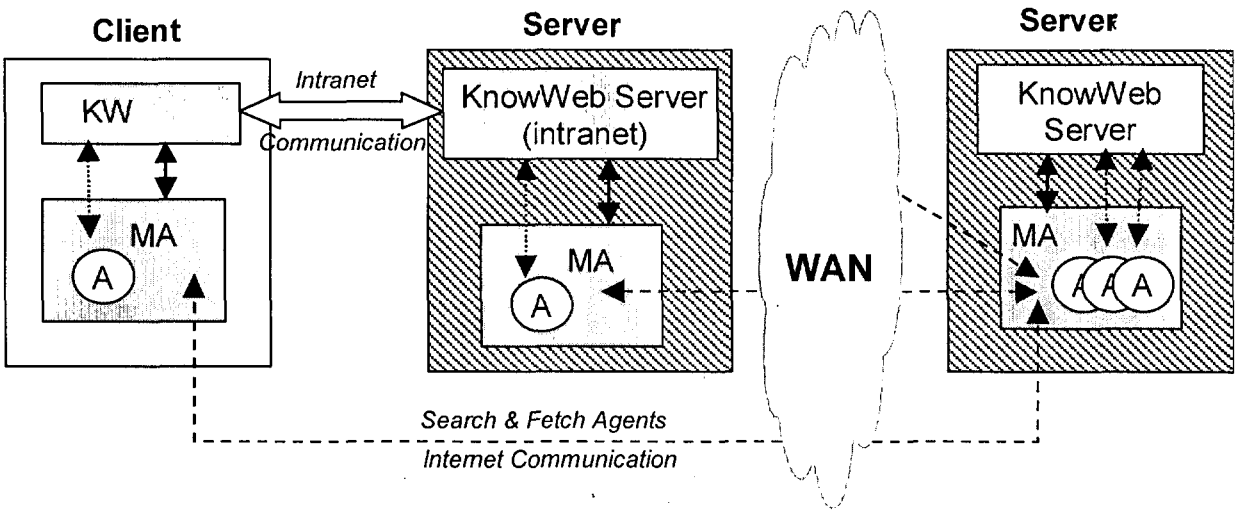


Figure 4. Mobile-agent based distributed structure of the KnowWeb system – supports also less reliable and low speed WANs

However, the accessibility of KnowWeb servers outside the fast intranet networks assumes a reliable network connection with guaranteed throughput especially when users want to introduce new documents into OM and browse already introduced documents. Unfortunately, this is not the case with most portable computers (notebooks etc.) or organization branches connected through a low-speed dial-up network connection.

To solve the major problem with the throughput the mobile agent-based solution has been proposed and

Dedicated mobile agents can do the most critical operations such as retrieval and gathering of documents at non-intranet servers. In retrieval operation client can formulate the query locally and either send an appropriate searching agent directly or demand the sending of an agent by a local KnowWeb server. The situation with gathering an already introduced document (accessible through other KnowWeb server) is similar. The main advantage of this functionality is the possibility for a client to get off-line for the period of operation

execution. By the next re-connect the results of ordered operations will be presented to the user. Such an approach brings significant savings to the communication between distributed KnowWeb servers containing a distributed OM.

To enable the work with mobile agents the *Mobile Agent Environment* (MAE) must be available on each concerned computer (client or server). MAE offers the following functionality: (i) creation of a mobile agent with a unique identity, (ii) transport of an agent, (iii) sending a message to an agent (possibly on another host), and (iv) getting the status information about any agent. MAE used in the research prototype of the KnowWeb toolkit is described in the following section.

5 Experimental framework for mobile agents

Basic functions of mobile agent environments (in today's mobile agent systems represented by agent servers) are identified by the Mobile Agent System Interoperability Facility (MASIF) (Milojcic et al. 1998) and include: (i) transferring an agent, which can include initiating an agent transfer, receiving an agent, and transferring classes, (ii) creating an agent, (iii) providing globally unique agent names, (iv) supporting the concept of a region, (v) finding a mobile agent, and (vi) ensuring a secure environment for agent operation.

A MA-based application usually requires to be programmed in two separate parts: mobile agent and location like *context* in Aglets (Oshima et al. 1998), *location services* in Ara (Peine & Stolpmann 1997), *place* in Gypsy (Jazayeri & Lugmayr 2000) or *service bridges* in Concordia (Wong et al. 1997). Mobile agent has a predetermined interface and restricted sources to communicate with on each visited location.

Mobile agent based distributed architecture of the KnowWeb system use the ESMA toolkit – developed and implemented at the Dept. of computers and informatics at the TU Košice (Paralič M. 2000). ESMA - the experimental system for support of mobile agents combines the power of high-level distributed programming with the mobile agent paradigm. As implementation platform the Mozart system (The Mozart Programming System) was chosen. It implements the Distributed Oz (DOz) programming language and offers simultaneously the advantage of a *True Distributed System* and the means for building a *Mobile Code System* (Picco 1998). In the following subsections mobile agent environment built from servers and its services and the methodology of how to build an agent-based application in our framework are shortly introduced.

5.1 Mobile agent environment

The basic functions offered by a mobile agent environment are transport and management of agents. In today's agent systems these services are offered by servers, which must be installed and running on every host computer that should be available for the mobile agents. Similarly our experimental framework in DOz offers basic functionality mentioned above.

MAE in the current implementation of the system can be started only once per host computer. Every agent created on a local MAE is a *home agent* for this MAE. On all other sites it will visit, it gains the status of a *foreign agent*. The MAE stores information about all its home agents and current available foreign agents in local database. The information about foreign agents is stored in the system only during time between the successful receiving from previous and successful sending an agent to the next host.

For the programmer of a mobile agent based application, the mobile agent environment is represented by the MAE module, which must be imported. Importing a MAE functor causes either the launching of a new environment with initialization values from persistent database for home and foreign agents and already connected other MAE servers or getting a reference to an already started MAE local server. This process is realized not only by launching a new local mobile-agent based application but also by resuming every incoming mobile agent. Thus, an agent gets access to the key services of the mobile agent environment. The possibility of dynamic loading and installation of first-class modules (Duchier et al. 1998) is thereby very important.

The communication between MAEs is realized in two layers: the first layer uses TCP sockets for exchanging of tickets for Oz ports. Oz ports then build a second, high-level communication layer, which can take advantage of Oz space data transparency. The Oz space offers the possibility to transparent copying of stateless entities and creating references for worldwide unique stateful entities. These possibilities can be fully utilized especially by the inter-agent communication.

5.2 Mobile agent based applications

Creating a MA-based application in the proposed framework is straightforward and requires only the following steps:

1. Identifying all fixed, not transferable resources needed by the application (i.e. their type) by means of abstract names and identifying parts of the transferable agent state.
2. Design and implementation of application-specific classes that are derived from the *MobileAgent* class and deal with agent state and other resources through their abstract names.
3. Designing and implementing an application that creates one or more instances of mobile agents, specifies their itinerary, sends them away, waits until they finish their jobs (or until the owner stops their work), and processes the results.

4. Design and installation of special environment modules (functors) in a compiled form. They map the abstract sources of MA to the real local resources of the host computer that should enable the execution of the MA based application.

5.3 KnowWeb and ESMA

To make the distributed structure of the KnowWeb system more flexible and suitable not only for fast intranet networks two special agent classes were created. The first one is *KW_SearchAgent* and the other one is *KW_FetchAgent*. Both mobile agent classes were proposed and implemented according to the methodology in section 5.2. The main task of the *KW_SearchAgent* is to get a query from the KnowWeb client and walk according to the travel plan of the KW servers in order to get the list of relevant documents – i.e. their exact addresses. Based on this list a KW client can send one ore more agents from the class *KW_FetchAgent*, which gather the whole document and return back with it.

At the client side are mobile agents created automatically after storing a query or fetching requirement in special form at local file system in predefined directory. At the server side *KW_FetchAgent* communicates directly with the WWW server that maintains the document space in the KnowWeb system. *KW_SearchAgents* store their queries at the local file system in predefined directory and are waiting for the answers, which can read also from the local file system.

6 Applications

6.1 KnowWeb pilot applications in BRD

Botnia Retail Data Inc. (BRD), located in Finland, has developed two of the five existing pilot applications based on the KnowWeb system. Main business activity of the BRD group is software development and consultancy for retail and industry. Their main product is WINPOS® - A Point-of-Sale (POS) software package.

In BRD felt that the best way to understand how to use the KnowWeb system is to first use the software within their organization with a domain knowledge model suitable for them. The second phase was the simulation of a retail environment with a domain knowledge model suitable for retail chains.

The database being used was an SQL Server database running under Windows NT to which documents are stored when they are finalised. Each user, who belongs to the personnel of BRD, uses Windows NT workstations on which KnowWeb clients are operating. The documents entered by a user shall typically be linked to a group of contexts. In order to quickly find the right contexts where to link the documents they have created a number of individual concept conditions. They are handling mainly documents of the MS Word or HTML types. They often scan in news articles and other printed material, which is first, pasted into a MS Word document and attached with explanations before the document is stored into the KnowWeb space.

The advantages identified in BRD after a couple of months of active KnowWeb using are the following.

1. **Internal efficiency improved**. They have been able to organise themselves much better than before. Now anyone in BRD can access relevant documents anytime without having to hunt for a document and wasting the time of his/her colleagues.
2. **Faster customer response**. Support personnel are now able to on-line check all information related to various WINPOS® versions and features.
3. **Exact and broader feedback to development**. They are tracing competitor information as well as feedback and feature requests from their end-users and dealers of their WINPOS® software product. Before they did not systematically trace this information. With the new system the development engineers are receiving much better information as a base for decisions about further enhancements to the product range.
4. **Improved marketing**. Through their systematic entering of competitors' information and their own marketing material as well, combined with the better feeling for what features customers are actually asking for, they have the feeling the system has helped them in improving their marketing activities.

On the other hand the following disadvantages have been reported:

1. **User interface** of the KnowWeb system prototype version seems to be too complicated to use for ordinary users.
2. There is a **time overhead in inserting the documents into the KnowWeb system**. During a busy day this is simply not being done.
3. The **success** of the introduction of the KnowWeb system to a quite large extent **depends on the discipline of the staff** using it.

Main motivation of the second BRD's pilot application was to structure all the events that within a retail chain cause disturbance to the present information system. These exceptions are in modern network environments most often reported in form of emails and documents sent between retail head office departments and the shops; today often without structure and forgotten after some time - thus leaving incorrect information in various databases as reports of present POS information systems. When planning a campaign or when making judgments e.g. regarding profitability on certain article groups, the marketing departments are often relying on historical information, which may go several years back. If the figures are unreliable, the decisions taken may be incorrect - the problem today is that many of the systems in the market today are not able to trace disturbances that have happened in the past.

Having this in mind, in BRD have linked their WINPOS® Point-of-Sales software to the KnowWeb system within the second pilot application. The end-of-day routines in the WINPOS® Point-of-Sales package (normally run at the end of the day when the shop has closed) are in this pilot application automatically generating shop-specific html documents, which are automatically introduced to the KnowWeb space. This means that the html-reports are automatically inserted in the Know-Web database and automatically linked to some predefined concepts via templates.

Moreover, certain date related information leads to dynamic generation of new instances in the domain model. As examples of documents being stored automatically we can mention for example shop-report, that is: sales, profit and payment media information summarised for a shop. Another example is the so-called department report, which contains sales amount, quantities and profit for the main article groups of the company. The advantages the customer will have from this system are:

1. **Internal efficiency improved.**
2. **Faster customer response.**
3. **Improved marketing.**
4. **Sales and profit analysing related to disturbances.** Automatically created and linked POS reports on one side are combined within the KnowWeb system with information about events causing various disturbances in retail chain (see above) providing realistic view on calculated numbers in context of occurred events.

6.2 Webocrat system

Another, very interesting application domain is electronic democracy (Paralič & Sabol 2001). An architecture of a Web based system Webocrat[2] is being designed with aim to empower citizens with innovative communication, access and voting system supporting increased participation of citizens in democratic processes and increase transparency and accessibility of public administration (PA).

Basic scheme of the proposed WEBOCRAT system is depicted in Figure 5. The system is based on knowledge modelling technology. Ontological knowledge models are employed in order to index all the information present in the system. Therefore first layer - *knowledge model* is depicted in the core of the scheme on Figure 5.

From the technical point of view, the system will be based on results achieved within the KnowWeb project focused on organising information using domain knowledge models. Their employment enables precise annotation of information based on its content, which results in efficient and powerful information retrieval capability.

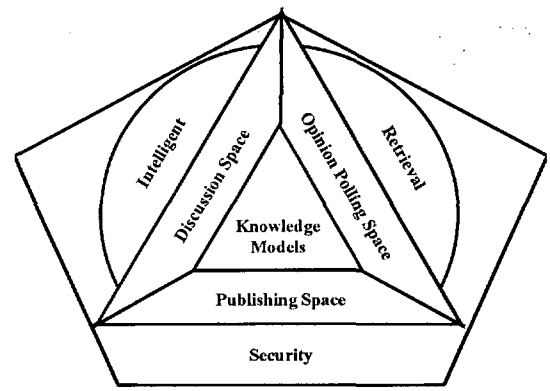


Figure 5. The principal scheme of the Webocrat system.

The second layer is represented by *publishing, discussion* and *opinion polling spaces* providing means for storing, updating and managing (in principle three slightly different types of) documents and their relations among themselves, as well as their relations to the knowledge model in the WEBOCRAT system.

The *Discussion Forum* (DF) module will support intelligent communication processes between public authorities, citizens and their elected representatives. DF will be responsible for documents in *discussion space.*

The *Web Content Management* (WCM) module will support publication of documents on the Internet, i.e. it will be responsible for documents in the *publishing space.*

The *Opinion Polling Room* (OPR) will enable electronic opinion polling featuring also with support for authentication and voter's privacy. OPR will be responsible for documents in the *opinion polling space.*

The third layer is composed from two retrieval-focused modules supporting retrieval capabilities that need only read access to the three above-mentioned spaces as well as to the knowledge model. The *Information Desk* (ID) will retrieve relevant documents of different types that are stored in the system.

The *Reporter/Summary* (REP) module will provide means for calculation of a variety of statistics as well as some more sophisticated approaches based on, e.g. data mining techniques. Moreover, this module will provide also alerting functionality, which has been required by user partners.

User registered in the system as an individual entity (i.e. not anonymous user) is provided with a personal access page ensuring him/her an individual access to the system. This page is built in an automatic way and can contain several parts. Some of them can be general and the other are person specific.

[2] EC funded project IST-1999-20364 Webocracy (Web Technologies Supporting Direct Participation in Democratic Processes)

The former can serve as a starting point for browsing all published documents accessible to the user, all conferences he/she is allowed to participate in, all running polls for which he/she is eligible, using search facilities of the system, read hot information, etc. The latter parts are devoted to user's personal newsletter, links to documents and conferences topics of which match the user's area of interest.

The personal access page hides division of the system into modules. Terms 'publishing space', 'discussion space', and 'opinion polling space' do not confuse users. The personal access page enables user to access all functionality of the system that he/she is allowed to access in a uniform and coherent way.

7 Acknowledgents

This work is done within the Webocracy project, which is supported by European Commission DG INFSO under the IST programme, contract No. IST-1999-20364 and project No. 1/8131/01 "Knowledge technologies for information acquisition and retrieval" supported by the Grant Agency of Ministry of Education and Academy of Science of Slovak Republic.

The content of this publication is the sole responsibility of the authors, and in no way represents the view of the European Commission or its services.

8 References

[1] Abecker A., Bernardi A. Hinkelmann K. Kühn, O. & Sintek M. (1998): Toward a Technology for Organizational Memories, *IEEE Intelligent Systems*, 13, May/June, p.40-48

[2] Borghoff U. M. & Pareschi R. Eds. (1998) Information Technology for Knowledge Management. Springer Verlag

[3] Chandrasekaran B., Josephson J.R. & Benjamins V.R. (1999) What Are Ontologies and Why Do We Need Them. *IEEE Intelligent Systems*, 14, p. 20-26

[4] Duchier D., Kornstaedt L., Schulte Ch. & Smolka G. (1998) A Higher-order Module discipline with separate Compilation, Dynamic Linking, and Pickling, *Technical Report*, Programming Systems Lab, DFKI and Universität des Saarlandes

[5] Gruber T.R. (1993) A Translation approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5, 2

[6] Harrison C.G., Chess D.M. & Kershenbaum, A. (1995) *Mobile Agents: Are they a good idea?* Research Report, IBM Research Division

[7] Jazayeri M. & Lugmayr W. (2000) Gypsy: A Component-based Mobile Agent System, Proceedings of the *8th Euromicro PDP'2000*, Rhodos, Greece

[8] Lange, D.B. & Oshima, M. (1999) Seven Good Reasons for Mobile Agents. *Communications of the ACM*, 42, 3, p. 88-89

[9] Mach M., Sabol T., Paralič J. & Kende R. (2000): Knowledge Modelling in Support of Knowledge Management. *Proceedings of the CRIS'2000 Conference*, Espoo-Helsinki, Finland, p. 84-88.

[10] Milojcic D., Breugst M., Busse I., Campbell J., Covaci S., Friedman B., Kosaka K., Lange D., Ono K., Oshima M., Tham C., Virdhagriswaran S. & White J. (1998) MASIF: The OMG Mobile Agent System Interoperability Facility. *Proc. of the Second International Workshop, Mobile Agents '98,* Springer-Verlag, p. 50-67

[11] Motta E. & Zdrahal Z. (1998) A principled approach to the construction of a task-specific library of problem solving components. *Proceedings of the 11th Banff Knowledge Acquisition for KBS Workshop*, Canada

[12] Newell A. (1982) The Knowledge Level. *Artificial Intelligence*, 18, p. 87-127

[13] Nonaka I. & Takeuchi H. (1995) *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation.* Oxford Univ. Press

[14] Oshima M., Karjoth G. & Ono K. (1998) Aglets Specification (1.1), *IBM Corporation*

[15] Paralič M. (2000) Mobile Agents Based on Concurrent Constraint Programming. *Lecture Notes in Computer Science*, Vol. 1897, pp. 62-75, Springer Verlag.

[16] Paralič, J. and Sabol, T. (2001) Implementation of e-Government Using Knowledge-Based System. Paper accepted for the 2nd Int. DEXA Workshop "On the way to Electronic Government", Munich, September 2001

[17] Peine H. & Stolpmann T. (1997) The Architecture of Ara Platform for Mobile Agents. *Proc. of the First International Workshop on Mobile Agents, MA'97*, LNCS 1219, Springer Verlag

[18] Picco G.P. (1998) Understanding, Evaluating, Formalizing, and Exploiting Code Mobility, *PhD thesis*, Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy

[19] Pritchard J. (1999) *COM and CORBA Side by Side: Architectures, Strategies, and Implementations.* Addison-Wesley

[20] Rothermel K., Hohl F. & Radouniklis N. (1997) Mobile Agent Systems: What is Missing? *Proc. of International Working Conference on Distributed Application and Interoperable Systems DAIS'97*

[21] The Mozart Programming System, Available at http://www.mozart-oz.org/

[22] Tiwana A. (2000) *The Knowledge Management Toolkit.* Prentice Hall

[23] van Heijst G., Schreiber A.T. & Wielinga, B.J. (1997) Using explicit ontologies in KBS development. *Int. Journal of Human-Computer Studies*, 46, p. 183-292

[24] Wong D., Paciorek N., Walsh T., DiCelie J., Young M. & Peet B. (1997) Concordia: An Infrastructure for Collaborating Mobile Agents, *Proc. of the First International Workshop on Mobile Agents, MA'97*, LNCS 1219, Springer-Verlag

An efficient approach to extracting and ranking the top K interesting target ranks from Web search engines

Chien-I Lee
Institute of Information Education
National Tainan Teachers College, Tainan, Taiwan, ROC
leeci@ipx.ntntc.edu.tw

Cheng-Jung Tsai
Dept. of Computer & Information Science
National Chiao Tung University, Hsinchu, Taiwan, ROC
tasicj.cis89g@nctu.edu.tw

Keywords: search engine, term weighting function, similarity function; ranking; Boolean expression

Received: April 2, 2001

Recently, several search engines have been established to help people find interesting information among the rapidly increasing number of web pages over the Internet. To obtain useful and reasonable searching results, users may submit queries with more than one query terms combined by a Boolean expression, supported by all existing search engines. However, these search engines all put the same emphasis on each query term combined by the Boolean expression. That is, for the identical queries, different users would obtain the same searching results. This contradicts the fact that different users usually have different searching interests even with the same queries. In other words, a useful search engine nowadays should allow users to emphasize each query term unequally to get the more reasonable and individual searching results. In this paper we propose an efficient approach, named Extreme Score Analysis method (ESA method), to solve this problem. ESA method uses web pages' original scores to derive users' top K web pages when each query term is assigned with a different weight. Moreover, we improve ESA method and further propose Extreme Score Analysis Inverse method (ESIA method), which can efficiently find users' top K interesting target ranks when these users assign different weights to each query term.

1 Introduction

As the fast process of computer and network technologies, computers connected over the Internet will soon become the indispensable electrical appliances of our daily life. Through the Internet, people can get the information they need much easier and quicker. Meanwhile, the rapidly growing data carried over the Internet really make people very difficult to search and filter the appropriate information they want. In recent years, several *search engines* [Brin & Page 1999, Buttler et. al. 2001, Ikeji & Fotouhi 1999, Kruger et. al. 2000, Lee & Yuwono 1996a, Lee & Yuwono 1996b, Marchiori 1997, Mauldin 1997, Schwartz 1998, Sheldon et. al. 1996] had been proposed to reduce such overloads.

In general, in order to support a full-text information retrieval, a search engine filters terms in the contents of web pages to establish a corresponding *inverted index* [McGill & Salton 1983, Yu et. al. 1999]. When users submit a query with some proper terms (these terms are called *query terms* in the rest of this paper) as keywords, the search engine looks these query terms up through the inverted index to find out the corresponding web pages and ranks these web pages by its predetermined *term weighting function* and *vector similarity function*. Finally, the search engine returns all related web pages in sequence. Based on the ranking results, users can read these returned web pages efficiently and get more relevant information.

Usually, to enlarge or narrow the searching interests, users may submit a query with more than one query terms and use a *Boolean expression* to combine these query terms. Such a query is called a *combined query*. Although the search engines nowadays all support the Boolean expression, they put the same emphasis on all query terms combined by the Boolean expression. For example, when a user submits a combined query with two query terms, "job and salary", a typical search engine will return the web pages that contain both the two terms and all returned web pages are ranked according to their own scores (We call such a rank of each web page *original rank*). Each web page's score could be the sum of the individual *contributed value* for the two query terms in its own context. (Note that the contributed value for a query term denotes the value returned from the search engine's term weighting function.) However, users may put more emphasis on one query term rather than on the other. For instance, an

unemployed man may want to look for a job through the web pages, containing "job" information and "salary" as well. In this case the user might put more emphasis on "job" rather than on "salary". That is, for the identical queries, different users could obtain the different searching results because they usually have different searching interests. For quantify such a case, users should be allowed to assign different weights to each query term in the Boolean expression, e.g. "(0.7) job and (0.3) salary", which means the degrees of importance of the two query terms are in a ratio of 7:3.

To fulfill such a requirement for different weights of query terms, it is necessary to re-calculate the score of each web page according to the new given weights and then re-rank them (in this paper, the new rank of each web page is called *target rank*). Finally, users can get the real ranking results. However,a typical search engine only returns matched web pages with final scores, ranks, etc., but not the individual contributed value for each query term. To re-calculate the scores, there are two problems needed to be resolved. First, we must know the scoring function of the search engine. Usually, due to the commercial secret, a search engine's scoring function is unavailable, then we can solve this problem by adopting another scoring function, which has been pre-defined by our system. The second problem is that we have to scan every returned web page's content to get each query term's contributed value. Such a re-scanning task is too time-consuming to be practicable.

In a real situation, most users usually only view the top K web pages from all HN returned web pages [Bergman et. al. 2000, Chaudhuri & Gravano 1999] ($K << HN$, where HN is the total number of returned web pages) for the sake of time or lacking of patient. Therefore, the second problem mentioned above can be resolved efficiently by focusing on the top K issue.

Based on the top K issue, to avoid the overhead of re-calculating the whole returned web pages, we will propose an efficient *Extreme Score Analysis method* (*ESA method*). ESA method, without re-scanning all returned web pages to get each query term's contributed value of each page, retrieves web page's original scores and applies the numerical analysis to find out users' top K interesting target ranks.. It would inform users that their top K interesting target ranks will be among the top R original ranks ($K \leq R << HN$). In the follwing performance study in Section 4, we find that the value of R is close to that one of K. For example, if one user submits a combined query with two query terms and request the top K web pages according to his/her weights (assume 0.6 and 0.4, respectively), R is approximately equal to $1.5K$. If the cost for re-scanning the top R web pages is allowable by users, our approach can provide users with their real top K target ranks.

The rest of the paper is organized as following. In Section 2, we survey a typical search engine's components and discuss the fundamental of term weighting function and vector similarity function. Section 3 proposes the basic idea Extreme Score Analysis method (ESA method) and a system prototype with this method. Section 4 presents an experimental evaluation of ESA method. In Section 5, we further improve the efficiency of ESA method, named *ESIA method* (*Extreme Score Inverse Analysis method*). The mathematical analysis is presented in Section 6. Finally, Section 7 is the conclusion.

2 The components of a search engine

When a user attempts to locate relevant information within a corpus in the Web, a web search engine system is the most common choice. A typical search engine system, composed of four components, is shown in Figure 1.

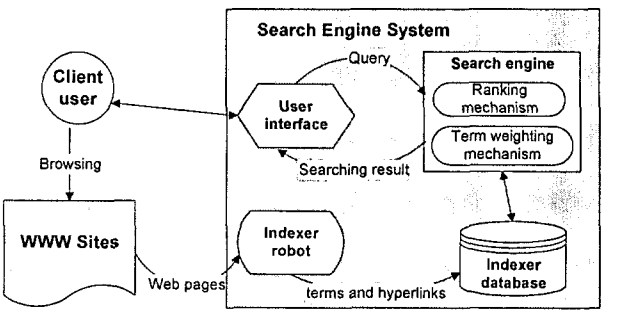


Figure 1. A typical search engine

a.) **User interface**
The user interface of a web search engine is a HTML (hypertext markup language) form, which can be invoked by standard WWW client programs such as Internet Explorer and Netscape, and manages the interaction between a user and a search engine. It accepts a user's query, dispatches the query to search engine, and finally displays the searching results to the user.

b.) **Indexer Robot**
The indexer robot [Cho et. al. 1998, Introna & Nissenbaum 2000] is an autonomous WWW browser, which communicates with WWW servers using HTTP (Hypertext Transfer Protocol). It visits a given WWW site, traverses hyperlinks in a breadth-first manner, retrieves the web pages, extracts terms and hyperlink data from the pages, and inserts these extracted data into an indexer database. A list of target sites is given to the indexer robot for creating the index initially.

c.) **Indexer database**
The Indexer robot reads web pages and sends these web pages to the indexer database (is also named inverted index) to crate indexing records. If a change of a web page is made, the inverted index should be also updated. The update will not be made in the inverted index but for the indexer robot has revisited this web page again. To reduce storage overhead, each web page in the inverted index is represented by a set of proper terms.

d.) **Search engine**
This part is mainly composed of two important mechanisms:

I.) **Term weighting mechanism**
This mechanism adopts a term weighting function to assign a specific weight to each term of each web page. This weight indicates the importance of this term in representing this web page. As a result, a web page in the indexer database can be regarded as a vector $p = (p_1, \ldots, p_m)$, where p_i is the weight of the ith term in representing the web page. A well-known term weighting function is named *tf×idf* [Kantrowitz et. al. 2000, McGill & Salton 1983], which assumes that term importance is proportional to the standard occurrence frequency of each term t_j in each web page H_i (that is, $FREQ_{ij}$) and inversely proportional to the total number of web pages in the web page collection (that is, $HOPFREQ_i$) to which each term is assigned. Then, a general form of term weighting function tf×idf can be

$$WEIGHT_{ij} = FREQ_{ij} \times [\log_2(n) - \log_2(HOPFREQ_k) + 1],$$

where n is the total number of web pages in the collection.

There are many others proposed term weighting functions, such as *Signal-Noise Ratio* and *Term Discrimination Value* [McGill & Salton 1983, Ricardo & Berthier 1999]. Although every term weighting function has its own properties, all of them propose the same hypothesis that term importance is proportional to the standard occurrence frequency of each term in each web page as the one in tf×idf. Nevertheless, tf×idf is superior to the others in the aspects of efficiency and cost [McGill & Salton 1983, Ricardo & Berthier 1999]. Furthermore, since the web pages are written in hypertext markup language (HTML), each search engine may take some related factor (e.g. hyperlink, number of times the keywords occur in the document title, etc.) into account to provide more suitable ranking results [Aguiar & Beigbeder 2000, Birmingham et. al. 1999]. Consequently, most term weighting function applied by the existing search engines are derived from tf×idf.

II.) **Similarity mechanism**
In order to rank all related web pages, a search engine assign relevance scores to them [Marchiori 1997]. The scores indicate the similarities between a given query and these web pages and this ranking task is implemented by a vector similarity function. Although there were many kinds of vector similarity functions, they all exhibited one common property, namely that the similarity value increases when the weight of the common properties in two vectors increase [Ricardo & Berthier 1999]. *Dot-product* function, the basic of all vector similarity functions, has been widely used for the evaluation of retrieval functions [Fan et. al. 2000, Liu et. al. 1999]. Even the well-known *Cosine* function is simply the dot-product function with the weights of terms computed in a specific manner. By dot-product function, the similarity between a query (is also represented as a vector $q = (q_1, \ldots, q_m)$, where q_i is the weight of the ith term t_i) and a web page could be represented as follows.

$$Similarity(q, p) = \sum_{i=1}^{m} q_i \times p_i,$$

where p_i is the weight of the ith term in representing a web page, q_i is the weight of the ith term in representing a query, and m is the number of terms in representing a web page.

3 Extreme Score Analysis method and the system prototype

3.1 Extreme Score Analysis method

As discussed in Section 2, we can assume that when a user submits a combined query with n query terms, the score KS_i (in this paper, we call it *original score*) that a search engine assigns to each web page H_i (i denotes the original rank of this web page among the returned matched web pages) will be

$$KS_i = \sum_{j=1}^{n} W_j (\, b((FREQ_{ij} + a)/HOCFREQ_j) + c\,),$$

where $1 \le i \le HN$, HN is the number of all returned web pages.

In the above equation, W_j is the weight of term j given in the user's combined query and $b((FREQ_{ij} + a) / HOCFREQ_j) + c$ is the contributed value of term j in the original score KS_i. Variables a, b, and c are used by each search engine to improve the tf×idf under the consideration of the web page's properties as mentioned in Section 2. Nevertheless, what our research concern is only about the ranking result of returned web pages. For the same search engine, whatever the values of Variables a, b, and c will be, the ranking result will never change. Furthermore, since all current search engines put the same emphasis on every query term (that is, $W_j = 1 / n$), the original score can be simplified as

$$KS_i = \sum_{j=1}^{n} (1/n) \times ConV_{ij},$$

where $1 \le i \le HN$, HN is the number of all returned web pages, and $ConV_{ij} = b((FREQ_{ij} + a) / HOCFREQ_j) + c$.

Obviously, if users assign each query term's weight unequally, each web page's original score may be changed. We call the score that is recomputed from the original score as *target score*

$$KT_i = \sum_{j=1}^{n} W_j \times ConV_{ij},$$

where $\exists\, j, W_j \neq 1/n$.

For example, suppose that a user submits a combined query with two query terms, term p and term q, and the user wants to assign 0.7 and 0.3 to each query term's weight respectively. Then, the original score of the ith matched web page is

$$KS_i = 0.5ConV_{pi} + 0.5ConV_{qi}.$$

And, the target score of the ith matched web page will be

$$KT_i = 0.7ConV_{pi} + 0.3ConV_{qi}.$$

First, to be convenient to explain our method, we consider the case with two query terms, term p and term q. Without loss of generality, we divide $ConV_p$ and $ConV_q$ by their maximum respectively to limit the value of them between 0 and 1, i.e., $0 \le ConV_p, ConV_q \le 1$. And there are two lemmas and three theorems, which are applied in our ESA method, needed to be stated and proved as follows.

Lemma 1. *Let x-axis and y-axis denote $ConV_p$ and $ConV_q$ respectively in the coordinate plane. Then a linear equation, which passes through the origin and is orthogonal to the straight line $SC = W_p \times ConV_p + W_q \times ConV_q$, will be $ConV_q = (W_q / W_p) \times ConV_p$, where $0 \le W_p, W_q \le 1$.*

Proof. According to *Slope Theorem*, the product of the slopes of two straight lines is –1, if the two lines are orthogonal to each other. Therefore, we can obtain the slope of the straight line, which is orthogonal to the straight line SC, is W_q / W_p. Moreover, because this straight line passes through the origin, we can get its linear equation is $ConV_q = (W_q / W_p) \times ConV_p$. □

Lemma2. *Let $SC_1 = W_p \times ConV_p + W_q \times ConV_q$ and $SC_2 = W_p \times ConV_p + W_q \times ConV_q$ represent two parallel lines in the coordinate plane and intersect the straight line $ConV_q = (W_q / W_p) \times ConV_p$ in the point A and point B respectively. If the length of line segment $\overline{OA}$ is longer than the length of line segment $\overline{OB}$, then the value of SC_1 is larger than the value of SC_2.*

Proof. First, we can obtain the coordinates of A is $((SC_1 \times W_p) / (W_p^2 + W_q^2), (SC_1 \times W_q) / (W_p^2 + W_q^2))$, and that of B is $((SC_2 \times W_p) / (W_p^2 + W_q^2), (SC_2 \times W_q) / (W_p^2 + W_q^2))$. Then, we can obtain the length of the two line segments are

$$\overline{OA} = SC_1 \Big/ \sqrt{W_p^2 + W_q^2}$$

$$\overline{OB} = SC_2 \Big/ \sqrt{W_p^2 + W_q^2}.$$

Clearly, since the length of line segment $\overline{OA}$ is longer than the length of line segment $\overline{OB}$, we can easily infer that the value of SC_1 is larger than that of SC_2. □

We use Figure 2 and Figure 3 to show the major concept of the above two lemmas. In both figures, x-axis and y-axis represent the contributed values of the query terms p and q, respectively. The point $(ConV_{pi}, ConV_{qi})$ in the coordinate plane denotes the web page H_i returned by a search engine. Every dashed straight line, intersecting the point $(ConV_{pi}, ConV_{qi})$, denotes the score of the web page H_i. In particular, Figure 2 represents the distribution of original scores of returned web pages, while Figure 3 represents the distribution of target scores. From the comparison of the two figures, we can observe that the dashed line, which denotes the score of web page H_i, will alter when the weights of the two query terms vary. From Lemma 1 we can know the reason is that the original score considers the weights of both query terms are equivalent (that is, $W_q / W_p = 1$), but target score does not (that is, $W_q / W_p \neq 1$). As well, from Lemma 2 we can obtain that the further the distance between the dashed straight line and the origin is, the higher the score denoted by the dashed straight line will be.

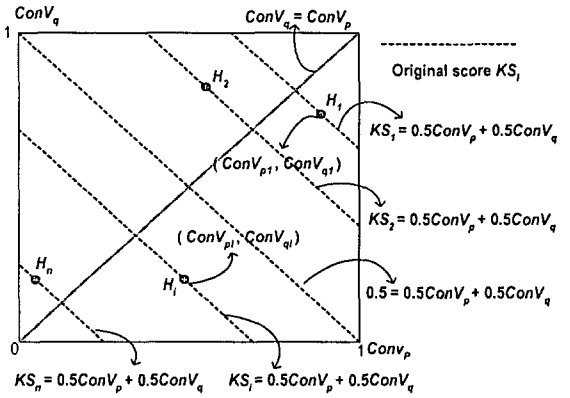


Figure 2. The distribution of the original scores in the coordinate plane.

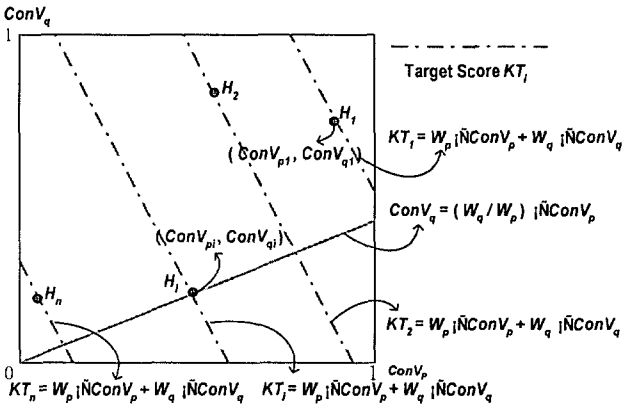


Figure 3. The distribution of the target scores in the coordinate plane.

Theorem 1. *As shown in Figure 4, let the original score $KS_i = 0.5ConV_p + 0.5ConV_q$ denotes a straight line in the coordinate plane, where $0 \le ConV_p, ConV_q \le 1$, $1 \le i \le HN$, HN is the number of all returned web pages, and $0 < W_q < W_p < 1$. Then, we can obtain an infinite number of straight lines $KT_i = W_p \times ConV_p + W_q \times ConV_q$, which*

intersect KS_i and are orthogonal to the straisght line $ConV_q = (W_q / W_p) \times ConV_p$.(Note that these straight lines KT_i denote the probable target score of the original score KS_i.) Therefore, for each original socre KS_i, we can obtain a pair of maximum target score KT_{maxi} and minimum target score KT_{mini}, which satisfy $KT_{maxi} \le KT_i \le KT_{mini}$.

Proof. Solve simultaneously the set of Equations (1) and (2):

$$\begin{cases} KS_i = 0.5ConV_p + 0.5ConV_q, & (1) \\ KT_i = W_p \times ConV_p + W_q \times ConV_q. & (2) \end{cases}$$

By (1) × $2W_p$ - (2), we can obtain

$$ConV_q \times (W_p - W_q) = 2W_p \times KS_i - KT_i.$$
$$ConV_q = (2W_p \times KS_i - KT_i) / (W_p - W_q). \quad (3)$$

By (1) × $2W_q$ - (2), we can obtain

$$ConV_p \times (W_q - W_p) = 2W_q \times KS_i - KT_i.$$
$$ConV_p = (2W_q \times KS_i - KT_i) / (W_q - W_p).$$

From the definition of Theorem 1 we can obtain

$$0 \le ConV_p, ConV_q \le 1. \quad (5)$$
$$1 < W_p < W_q < 0. \quad (6)$$

By solving simultaneously Equations (3), (5), and (6), we can obtain

$$0 \le (2W_p \times KS_i - KT_i) / (W_p - W_q) \le 1.$$
$$0 \le (2W_p \times KS_i - KT_i) \le W_p - W_q.$$
$$W_q + W_p (2KS_i - 1) \le KT_i \le 2W_p \times KS_i.$$
$$W_q + W_p (2KS_i - 1) \le KT_i \le W_p + W_p (2KS_i - 1). \quad (7)$$

Similarly, by solving simultaneously Equations (4), (5), and (6),we can obtain

$$W_q + W_q (2KS_i - 1) \le KT_i \le W_p + W_q (2KS_i - 1). \quad (8)$$

Finally, by solving simultaneously Equations (7) and (8), we can obtain the maximum of KT_i is

$$KT_{maxi} = Min (W_p + W_q (2KS_i - 1), W_p + W_p (2KS_i - 1)),$$

where the function *Min* is to get the minimum value among the set.

Similarily, the minimum of KT_i is

$$KT_{mini} = Max (W_q + W_q (2KS_i - 1), W_q + W_p (2KS_i - 1),$$

where the function *Max* is to get the maximum value among the set.

Therefore, for each original score KS_i, we can obtain a pair of maximum target score KT_{maxi} and minimum target score KT_{mini}, which satisfy $KT_{maxi} \le KT_i \le KT_{mini}$. □

Theorem 2. *Continuing with Theorem 1, let $KS_i = 0.5ConV_p + 0.5ConV_q$ and $KS_j = 0.5ConV_p + 0.5ConV_q$ represent two straight lines in the coordinate plane. If $KS_i > KS_j$, then $KT_{mini} > KT_{minj}$ and $KT_{maxi} > KT_{maxj}$.*

Proof. From Equations (7) and (8) in Theorem 1, we can get that if the values of term weights W_p and W_q are fixed, then the larger KS_i is, the larger KT_{maxi} and KT_{mini} will be. Therefore if $KS_i > KS_j$, then $KT_{mini} > KT_{minj}$ and $KT_{maxi} > KT_{maxj}$. □

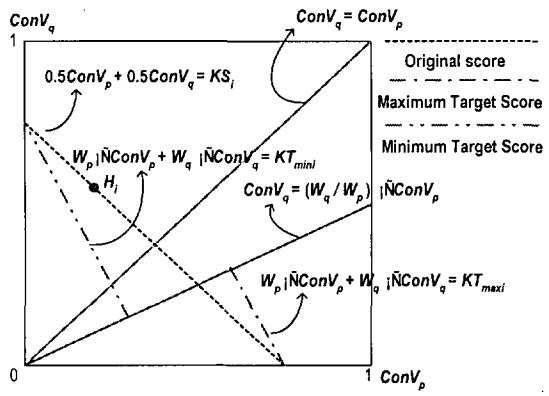


Figure 4. The relation between the original score and the target score.

Theorem 3. *By Theorem 1, we obtain that each original score KS_i has a pair of maximum target score KT_{maxi} and minimum target score KT_{mini}, which satisfy $KT_{maxi} \le KT_i \le KT_{min}$ $(1 \le i \le HN)$. By Theorem 2, for all original scores KS_i, we have $KT_{max1} \ge KT_{max2} \ge ... \ge KT_{maxi} \ge ... \ge KT_{maxHN}$, and $KT_{min1} \ge KT_{min2} \ge ... \ge KT_{mini} \ge ... \ge KT_{minHN}$. Suppose there is an original score KS_j, whose maximum target score KT_{maxj} is larger or equal to KT_{mink}, then its corresponding target score KT_j may be one of the top K of all target score KT_i.*

Proof. Suppose there is an original score KS_j $(j > K)$, whose maximum taget score KT_{maxj} is larger or equal to KT_{minK}, but KT_j should not be one of the top K of all target score KT_i. However, as asserted above, because we have $KT_{min1} \ge KT_{min2} \ge ... \ge KT_{minK-1} \ge KT_{minK}$, the top K of all target score KT_i will be $KT_1, KT_2, KT_3, ..., KT_K$ in turn when $KT_i = KT_{mini}$, where $1 \le i \le K$. Nevertheless, since KT_{maxj} is larger or equal to KT_{minK}, KT_j may be larger than KT_K and becomes the Kth rank of all KT_i when KT_j is equal to KT_{maxj}. As a result, our previous assumption is not true. Consequently, for each original score KS_j,if its maximum target score KT_{maxj} is larger or equal to KT_{mink}, then its corresponding target score KT_j may be one of the top K of all target score KT_i. □

As stated in Section 1, since the search engines do not return the individual contributed values $ConV_{pi}$ and $ConV_{qi}$ for each returned web page H_i, the coordinates of web page H_i, $(ConV_{pi}, ConV_{qi})$, may locate at anywhere on the straight line $KS_i = 0.5ConV_{pi} + 0.5ConV_{qi}$. As a result, we can not obtain the real target score KT_i for H_i. In Theorem 1, we have proved that the target score for each original score KS_i will be between *minimum target score* KT_{mini} and *maximum target score* KT_{maxi}. That is, for given term weights W_p and W_q, which are assigned by

a user, we can derive a value interval of target score from each original score. Furthermore, in Theorem 2 we have proved that the distribution of KT_{maxi} and KT_{mini} decrease gradually with the decrease in original score. Figure 5 explicitly shows such a case. Finally, according to Theorem 3, in order to support the real top K interesting target ranks from the returned web pages, every web page H_i whose $KT_{maxi} \geq KT_{minK}$ should be a choice. Therefore, instead of viewing the whole HN returned web pages, the user only has to view those web pages, whose $KT_{maxi} \geq KT_{minK}$, to get their real top K target ranks.

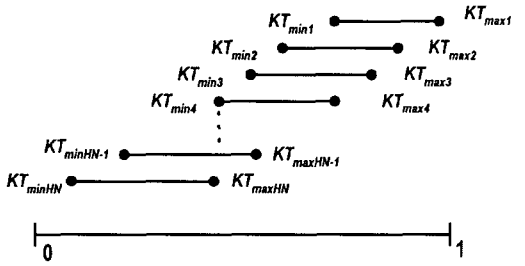


Figure 5. The distribution of minimum target score and maximum target score.

We have presented the basic idea of our ESA method by considering the case of two query terms in the above. Here, we will further extend it to the cases of n query terms. Without loss of generality, we let $W_1 \geq W_2 \geq \ldots \geq W_{n-1} \geq W_n$ and $0 \leq ConV_i \leq 1$, where $1 \leq i \leq n$, W_i denotes the weight of users' ith query term, and $ConV_i$ denotes the contributed value of the ith query term.

Suppose users submit a query with n query terms, the original score for each returned web pages H_i is

$$KS_i = (1/n) \times ConV_1 + (1/n) \times ConV_2 + \ldots + (1/n) \times ConV_n. \quad (9)$$

And, each returned web page's target score is

$$KT_i = W_1 \times ConV_1 + W_2 \times ConV_2 + \ldots + W_n \times ConV_n. \quad (10)$$

By (10) – $W_i \times n \times$ (9) for each i, where $1 \leq i \leq n$, we can obtain n number of equations as follows.

$$\begin{cases} KT_i = W_1 \times n \times KS_i - (W_1 - W_2) \times ConV_2 - (W_1 - W_3) \times ConV_3 - \ldots - (W_1 - W_n) \times ConV_n. \\ KT_i = W_2 \times n \times KS_i - (W_2 - W_1) \times ConV_1 - (W_2 - W_3) \times ConV_3 - \ldots - (W_2 - W_n) \times ConV_n. \\ \vdots \\ KT_i = W_n \times n \times KS_i - (W_n - W_1) \times ConV_1 - (W_n - W_2) \times ConV_2 - \ldots - (W_n - W_{n-1}) \times ConV_{n-1}. \end{cases}$$

Since $W_1 \geq W_2 \geq \ldots \geq W_{n-1} \geq W_n$ and $0 \leq ConV_i \leq 1$, from each equation above, we can obtain an interval of possible values of the target score KT_i. Therefore, for each web page H_i, we have n intervals of possible values of its target score KT_i as follows.

$$\begin{cases} W_1 \times n \times KS_i - (W_1 - W_2) \times ConV_2 - (W_1 - W_3) \times ConV_3 - \ldots - (W_1 - W_n) \times ConV_n \leq KT_i \leq W_1 \times n \times KS_i. \\ W_2 \times n \times KS_i - (W_2 - W_3) \times ConV_3 - (W_2 - W_4) \times ConV_4 - \ldots - (W_2 - W_n) \times ConV_n \leq KT_i \leq W_2 \times n \times KS_i - (W_2 - W_1) \times ConV_1. \\ \vdots \\ W_n \times n \times KS_i \leq KT_i \leq W_n \times n \times KS_i - (W_n - W_1) \times ConV_1 - (W_n - W_2) \times ConV_2 - \ldots - (W_n - W_{n-1}) \times ConV_{n-1}. \end{cases}$$

In other words, we can obtain a set of candidates for the maximum target score KT_{maxi} (the set is denoted by $MaxKT_i$) and a set of candidates for the minimum target score (the set is denoted by $MinKT_i$). The minimum value in the $MaxKT_i$ is the maximum target score KT_{maxi} of the web page H_i, and the maximum value in the $MinKT_i$ is the minimum target score KT_{mini}.

$$KT_{maxi} = Min(MaxKT_i)$$
$$KT_{mini} = Max(MinKT_i)$$

Now, we can generally consider the case with n (≥ 2) query terms in our ESA method. Figure 5 shows the major steps in ESA method; and it also shows how to evaluate the required top R web pages ($R \geq K$), which have to be viewed by the user to get his/her real top K target ranks. ESA method requires five kinds of input data, including W_j, K, n, HN, and KS_i, where W_j denotes the weight of each query term j assigned by the user, K is the number of the user's interesting target ranks, n denotes the number of a user's query terms, HN is the number of related web pages returned by a search engine, and KS_i is the original score of each returned page H_i. In Step 1, ESA method outputs each web page's maximum target score KT_{maxi} and minimum target score KT_{mini}, and then these bounds are used in Step 2 to derive the value of R. Without loss of generality, we let $W_1 \geq W_2 \geq \ldots \geq W_{n-1} \geq W_n$. Besides, to simply the expressions, we define two functions in Figure 6, which are

$$S_1(j, n) = \begin{cases} 0 & \text{if } n < j+1 \\ \sum_{m=j+1}^{n} W_m & \text{if } n \geq j+1 \end{cases}, \text{ and}$$

$$S_2(1, j) = \begin{cases} 0 & \text{if } j < 2 \\ \sum_{m=1}^{j-1} W_m & \text{if } j \geq 2 \end{cases}$$

Figure 5. Extreme Score Analysis method

Step1:
/* In this step, the agent with ESA method will compute the KT_{maxi} and KT_{mini} for every returned web pages.*/
Input data W_j, K, n, HN, KS_i
Output data KT_{mini}, KT_{maxi}

For i = 1 to HN
  For j = 1 to n
    $MinKT_j = W_j \times n \times KS_i - (n - j)\ W_j + S_1(j, n)$;
    $MaxKT_j = W_j \times n \times KS_i - (j - 1)\ W_j + S_2(1, j)$;
  End for
  $KT_{mini} = Max(MinKT_1, MinKT_2, \ldots, MinKT_n)$;
  $KT_{maxi} = Min(MaxKT_1, MaxKT_2, \ldots, MaxKT_n)$;
End for

Step2:
/* The top K interesting web pages will exist in the front of the top R original ranks of all HN returned web pages. */
Input data HN, KT_{minK}, KT_{maxi}
Output data R

For i = 1 to HN
  If $KT_{maxi} < KT_{minK}$ Then
    $R = i - 1$;
    Exit;
  End if
End for

Take Table 1 as the example, we consider the case with n = 2 and suppose the search engine returned 20 (= HN) related web pages. We also suppose a user wants to respectively assign 0.3 and 0.7 to each query term's weight, and the user requests the top 3 interesting web pages. That is, n = 2, HN = 20, W_1 = 0.3, W_2 = 0.7, and K = 3. To understand and verify the result ESA method, we used a generator to generate $ConV_q$ and $ConV_p$ randomly for these 20 web pages and then compute each web page's original score KS_i. The actual value of each target score KT_i also can be obtained as shown in Table 1. (Note that in a real situation, $ConV_q$, $ConV_p$, and KT_i are unavailable.) In Step 1, the KT_{maxi} and KT_{mini} for every returned web page are derived. Then, Step 2 derives the total number of web pages (i.e., the value of R) that the user has to view. In other words, Step 2 tries to find the web pages whose $KT_{maxi} \geq KT_{mink\,(k=3)}$ (= 0.854). In Table 1, we can find that KT_{max7} (= 0.866) > $KT_{minK\,(k=3)}$, but KT_{max8} does not. As a result, the user just needs to view the top R = 7 returned web pages (i.e., H_1, H_2, H_3, H_4, H_5, H_6, H_7) to get the real top 3 interesting web pages with the new given term weights. Additionally, a scanning mechanism can be also offered to avoid users to sieve the real interesting target ranks from a lot candidates when the value of K is large. Afterward, the user can directly get the real top 3 interesting target ranks (i.e., H_1, H_3, H_5). More noticeably, with ESA method, instead of all the HN (= 20) returned web pages, only the top 7 web pages are scanned.

Table 1. An example of ESA method (n = 2, HN = 20, W_1 = 0.3, W_2 = 0.7 and K = 3)

Page	*Ogr*	KS_i	KT_{mini}	KT_{maxi}	$ConV_p$	$ConV_q$	KT_i
H_1	1	0.934	0.908	0.96	0.920	0.948	0.928
H_2	2	0.906	0.868	0.944	0.870	0.942	0.892
H_3	3	0.896	0.854	0.938	0.980	0.812	0.930
H_4	4	0.860	0.804	0.916	0.780	0.94	0.828
H_5	5	0.857	0.800	0.914	0.950	0.764	0.894
H_6	6	0.811	0.735	0.887	0.770	0.852	0.795
H_7	7	0.776	0.686	0.866	0.760	0.792	0.770
H_8	8	0.726	0.616	0.836	0.990	0.462	0.832
H_9	9	0.648	0.507	0.789	0.713	0.583	0.674
H_{10}	10	0.622	0.471	0.773	0.406	0.838	0.536
H_{11}	11	0.579	0.411	0.747	0.666	0.492	0.614
H_{12}	12	0.542	0.359	0.725	0.997	0.087	0.724
H_{13}	13	0.532	0.345	0.719	0.746	0.318	0.618
H_{14}	14	0.519	0.327	0.711	0.538	0.500	0.527
H_{15}	15	0.429	0.257	0.601	0.200	0.658	0.337
H_{16}	16	0.349	0.209	0.489	0.090	0.608	0.245
H_{17}	17	0.279	0.167	0.391	0.340	0.218	0.303
H_{18}	18	0.114	0.068	0.16	0.220	0.008	0.156
H_{19}	19	0.060	0.036	0.084	0.091	0.029	0.072
H_{20}	20	0.056	0.034	0.078	0.01	0.102	0.038

*Ogr denotes the original rank of a web page

3.2 A system prototype with ESA method

Figure 6 illustrates the data flow among the user, our system prototype, and a web search engine. There are three main agents in our system prototype, which will be discussed as follows.

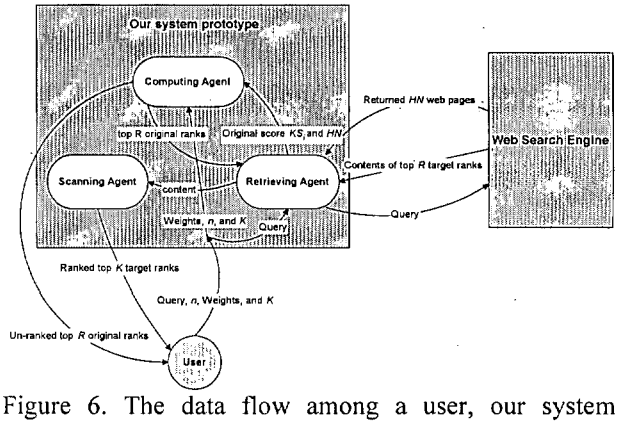


Figure 6. The data flow among a user, our system prototype, and a web search engine

a.) **Retrieving agent**
The retrieving agent accepts a user's query and dispatches this query to a web search engine. The original scores of all returned web page are retrieved and then conveyed to the second component, the computing agent. It would retrieve some web page's content and dispatch them to the scanning agent if such a task is necessary.

b.) **Computing agent**
This agent adopts ESA method and is responsible for deriving the candidate set of a user's top K target ranks. Its input data contain the original scores KS_i and the number of returned web pages HN from the retrieving agent; and the number of query terms n, the weights W_j of each query term, and the value of K from the user. When a scanning task is required, the computing agent sends the top R target ranks to the retrieving agent to ask the content of these web pages, otherwise it returns this un-ranked result to the user.

c.) **Scanning agent**
The scanning agent is composed of two pre-determined functions: the term weighting function and the similarity function, and two mechanisms: the full-text scanning mechanism and the ranking mechanism. It is an optional agent and works only when the user want to get his/her real top K target ranks. Initially, it accepts the content of the top R original ranks from the retrieving agent and then computes the contributed value of each query term in each retrieved web pages by the full-text scanning mechanism and the term weighting function. The similarity scores of these candidates are further computed by the similarity function. Finally, the ranking mechanism sorts these top R target ranks and the scanning agent returns the ranked top K target ranks to the user. The simple pseudo code of this agent is presented as follows.

```
Get the content of the top R original ranks from the
  retrieving agent
    For i = 1 to R
      Scan H_i and compute the contributed value
        of each query term j;
      Score the web page H_i according to the new
        computed contributed values and a new
        similarity function, which is
        predetermined by the system;
    End for
Rank H_i according to the target scores and return the top
  K target ranks to the user;
```

4 Evaluation of the experiment simulation model

4.1. Experiment simulation

In this section, to evaluate the efficiency of ESA method, we establish a simulation model for studying the effects based on different distribution of original scores by applying a generator to randomly generate several sets of the original scores. There are two performance measures in this simulation model. One is called *Per*, and the other is called *Mul*. *Per* denotes the percentage of web pages that a user need to view to get his real top K target ranks (i.e., $Per = 100\times(R / HN)$ %). *Mul* denotes the proportion of the number of a user's top K interesting target ranks to the number of returned web pages they have to view (i.e., $Mul = R / K$). Each *Per*, *Mul*, and R is obtained by averaging the results of simulating 100 times.

4.2. Results

The results of experiment simulation are presented in Table 2. The average of all *Mul* is 4.41 (that is, $R \approx 4.41K$), which means that on average users only have to view about 4.41 times the number of K to get their real top K interesting web pages. Even in the worst case *Mul* = 10.8 (where $W_p = 0.9$, $W_p = 0.1$, $HN = 5000$, and $K = 10$), the number of web pages users have to view is about 10.8 times the total of their top K (=10) interesting web pages. In such a condition, users can request the system to implement the re-calculated task. In this case, the total of returned web pages is 5000, and our system has only to re-calculate 108 (= 5000 × 2.16 %) returned web pages to provide users with the real top K (=10) target ranks, which is much more efficient than the way by re-calculating all 5000 returned web pages.

When users allow the system to re-calculate the real top K target ranks, the small *Per* in most cases has shown the efficiency of our method again. The average of all *Per* is 8.97%, which means that on average our system just need to re-calculate about 8.97% of all returned web pages to provide users with their interesting top K target ranks. *Per* will become much larger only when K is close to HN. However, a search engine usually returns much more web pages than a user's interesting top K web pages. That is, HN is usually much larger than K.

Moreover, we have some observations from the simulation results:

- For the same set of the query terms' weights and the same K, the larger the HN is, the less the *Per* will be.
- For the same HN and K, the less the difference between each query term's weight is, the less the *Per* will be.
- For the same set of the query terms' weights and the same HN, the smaller the K is, the smaller the *Per* will be.

5 Extreme Score Inverse Analysis method

As shown in the previous example in Table 1, ESA method has to first compute the maximum target score KT_{maxi} and minimum target score KT_{mini} for each returned web page H_i. When a search engine returns a large number of web pages, such a task is very time-consuming. In order to solve this problem, we propose an improved method as shown in Figure 7, called Extreme Score Inverse Analysis method (ESIA method). ESIA method only has to derive a predetermined *original score threshold* KS_0, while ESA method must calculate KT_{mini} and KT_{maxi} for each returned web page H_i. There is the major advantage in Step 1 of ESIA method compared to ESA method.

Table 2. The result of experiment simulation

W	*HN*	*K*	*R*	*Per*	*Mul*
	5000	10	17	0.33%	1.7
		20	33	0.66%	1.65
		30	47	0.924%	1.57
	1000	10	17	1.62%	1.7
$Wp = 0.6$		20	32	3.10%	1.6
		30	46	4.57%	1.53
$Wq = 0.4$	500	10	16	3.01%	1.6
		20	31	6.07%	1.55
		30	46	9.01%	1.53
		30	45	44.3%	1.5
	5000	10	26	0.51%	2.6
		20	50	1.00%	2.5
		30	74	1.47%	2.47
$Wp = 0.7$	1000	10	25	2.45%	2.5
		20	47	4.68%	2.35
$Wq = 0.3$		30	72	7.19%	2.4
	500	10	25	4.85%	2.5
		20	48	9.49%	2.4
		30	71	14.14%	2.37
	5000	10	48	0.95%	4.8
		20	93	1.86%	4.65
		30	122	2.43%	4.01
$Wp = 0.8$	1000	10	41	4.08%	4.1
		20	81	8.01%	4.05
$Wq = 0.2$		30	125	12.50%	4.17
	500	10	42	8.37%	4.2
		20	78	15.54%	3.9
		30	122	24.37%	4.01
	5000	10	108	2.16%	10.8
		20	201	4.01%	10.1
		30	280	5.59%	9.33
$Wp = 0.9$	1000	10	93	9.21%	9.3
		20	179	17.88%	8.95
$Wq = 0.1$		30	279	27.84%	9.3
	500	10	96	19.02%	9.6
		20	177	35.22%	8.85
		30	245	48.91%	8.17
Average				8.97%	4.41

Here, we will go into details to describe how our ESIA method derives the original score threshold KS_0 and how this threshold works. In Theorem 3, we have proved that the web page H_i may be one of the top K target ranks, if its corresponding $KT_{maxi} \geq KT_{mink}$. Therefore, we can regard KT_{mink} as a threshold of maximum target score (we use KT_{max0} to denote this threshold) and find its corresponding original score, KS_0. By ESA method in Figure 5, we can obtain that for the web page H_0

$KT_{max0} = Min\ (MaxKT_1, MaxKT_2, \ldots, MaxKT_n)$,
where n is the number of users' query terms, $MaxKT_j = W_j \times n \times KS_i - (j - 1)W_j + S_1(j, n)$, $1 \leq \theta \leq HN$, $1 \leq j \leq n$, and HN is the number of all returned web pages.

Let $KT_{max0} = KT_{mink}$, and we can get

$$Min\ (MaxKT_1, MaxKT_2, \ldots, MaxKT_n) = KT_{mink}.$$

Step1:
/*In this step, the agent with ESIA method will first derive the KT_{minK} and then compute the original score threshold KS_0. */
Input data W_j, K, n, HN, KS_K
Output data KS_0

For $j = 1$ to n
  $MinKT_j = W_j \times n \times KS_K - (n - j)\ W_j + S_1(j, n)$;
End For
$KT_{minK} = Max(MinKT_1, MinKT_2, \ldots, MinKT_n)$;
For $j = 1$ to n
  $KS_j = (KT_{minK} + (j - 1)\ W_j - S_1(j, n)) / (n \times W_j)$;
End For
$KS_0 = Max(KS_1, KS_2, \ldots, KS_n)$;

Step2:
/* The top K interesting web pages will exist in the front of the top R original ranks of all HN returned web pages. */
Input data HN, KS_i
Output data R

For $i = 1$ to HN
  If $KS_i < KS_0$ then
    $R = i - 1$;
    Exit;
  End If
End For

Figure 7. Extreme Score Inverse Analysis method.

Since we are not sure which $MaxKT_j$ is the minimum, we expand the above equation and can obtain n values of the original socre

$KS_j = (KT_{minK} + (j - 1)\ W_j - S_1(j, n)) / (n \times W_j)$,
where $S_1(j, n)$ is defined in ESA method, n is the number of users' query terms, and $1 \leq j \leq n$.

This means each original score KS_j can derive a candidate $MaxKT_j$ for its maximum target score, and these candidates' values all are equal to KT_{minK}. However, KT_{max0} is the corresponding maximum target score for the original score KS_0. In Theorem 2, we have proved that the larger the original score is, the larger the maximum target score will be. Therefore, we can obtain

$$KS_0 = Max\ (KS_j).$$

That is, when a web page's original score is larger than KS_0, this web page may be one of users' top K target ranks.

By using the same example as shown in Table 1, we explain proposed ESIA method. Step 1 derives KS_0 (in this example, $KS_0 = 0.757$) and then, Step 2 finds out the total of web pages that the user has to view (i.e., web pages whose original score $\geq KS_0$). We can observe that the result is identical to the one by ESA method. Most

notably, without calculating KT_{mini} and KT_{maxi} for each returned web pages H_i, ESIA method can be much more efficient than ESA method.

6 Evaluation of the mathematical analysis model

6.1. Mathematical analysis

Unlike the evaluation in Section 4, since the original scores do not have a fixed distribution format, to proceed with some mathematical analysis in this section, we assume the original scores returned by the search engine distribute uniformly in the interval (0,1). (Note that a web page with original score = 0 will not be returned by the search engine.)

Similar to Section 4, in this mathematical model we still use the two performance measures, *Per* and *Mul* as stated in earlier. To distinguish the evaluation in this section from that in Section 4, we use the superscript * in *Per**, *Mul**, and *R**.

Without loss of generality, we let $W_p > W_q$. From Equation (7) and (8) in Theorem 1 we can obtain that

If $KS_i \geq 0.5$ then

$$\begin{cases} KT_{mini} = W_q + W_p\,(2KS_i - 1), & (11) \\ KT_{maxi} = W_p + W_q\,(2KS_i - 1); & (12) \end{cases}$$

If $KS_i \leq 0.5$ then

$$\begin{cases} KT_{mini} = W_q + W_q\,(2KS_i - 1) & (13) \\ KT_{maxi} = W_p + W_p\,(2KS_i - 1). & (14) \end{cases}$$

Furthermore, from Theorem 3 and Equations (11), (12), (13), and (14), we can get that

If $KS_0 > 0.5$ and $KS_K > 0.5$ then

$$\begin{cases} KT_{max0} = W_p + W_q\,(2\,KS_0 - 1), \\ KT_{minK} = W_q + W_p\,(2KS_K - 1). \end{cases}$$

So we can get

$$KS_0 = (W_p \times KS_K - W_p + W_q) / W_q. \quad (15)$$

If $KS_0 \leq 0.5$ and $KS_K > 0.5$

$$\begin{cases} KT_{max0} = W_p + W_p\,(2\,KS_0 - 1), \\ KT_{minK} = W_q + W_p\,(2KS_K - 1). \end{cases}$$

So we can get

$$KS_0 = (2W_p \times KS_K - W_p + W_q) / 2W_p. \quad (16)$$

If $KS_0 > 0.5$ and $KS_K < 0.5$ then

$$\begin{cases} KT_{max0} = W_p + W_q\,(2\,KS_0 - 1), \\ KT_{minK} = W_q + W_q\,(2KS_K - 1). \end{cases}$$

So we can get

$$KS_0 = (2W_q \times KS_K - W_p + W_q) / 2W_q. \quad (17)$$

If $KS_0 \leq 0.5$ and $KS_K < 0.5$ then

$$\begin{cases} KT_{max0} = W_p + W_p\,(2\,KS_0 - 1), \\ KT_{minK} = W_q + W_q\,(2KS_K - 1). \end{cases}$$

So we can get

$$KS_0 = W_q \times KS_K / W_p. \quad (18)$$

On the assumption that original scores returned by the search engine distribute uniformly, the set of original scores will be an *arithmetic sequence* whose maximum is 1 and *common difference* is 1 / *HN*. Then, we can obtain

$$KS_i = 1 - ((i - 1) / HN),$$

and the *term T* of KS_0 in this *arithmetic sequence* is

$$T = HN\,(1 - KS_0) + 1.$$

Finally, we can get

$$Per^* = 100 \times (T / HN)\,\% = 100\,(1 + (1 / HN) - KS_0). \quad (19)$$

After substituting Equations (15), (16), (17), and (18) into Equation (19), we can get *Per** as following.

If $KS_0 > 0.5$ and $KS_K > 0.5$ then

$$Per^* = 100 \times (1 + (1 / HN) - (((W_p \times KS_K - W_p + W_q) / W_q))\,\%. \quad (20)$$

If $KS_0 \leq 0.5$ and $KS_K > 0.5$ then

$$Per^* = 100 \times (1 + (1 / HN) - ((2W_p \times KS_K - W_p + W_q) / (2W_p)))\,\%. \quad (21)$$

If $KS_0 > 0.5$ and $KS_K < 0.5$ then

$$Per^* = 100 \times (1 + (1 / HN) - ((2W_q \times KS_K - W_p + W_q) / (2W_q)))\,\%. \quad (22)$$

If $KS_0 \leq 0.5$ and $KS_K < 0.5$ then

$$Per^* = 100 \times (1 + (1 / HN) - (W_q \times KS_K / W_p))\,\%. \quad (23)$$

After computing *Per**, we can get

$$R^* = \text{Round}\,(Per^* \times HN), \text{ and}$$

$$Mul^* = R^* / K\,.$$

6.2. Result

The results of mathematical analysis are presented in Table 3. From the comparison of Table 2 and 3, we can find that the results of the experimental simulation model and the mathematical analysis model are close to each other. As a result, according to the results of the mathematical analysis, if users allow some degrees of errors, our approach can inform users how many returned web pages they have to view without any calculation. Similarly, our proposed method can decide how many web pages he has to retrieve and re-scan to provide users with their real top *K* target ranks without any calculation.

Table 3. The result of mathematical analysis

W	*HN*	*K*	*R**	*Per**	*Mul**
	5000	10	15	0.29%	1.5
		20	30	0.59%	1.5
		30	45	0.89%	1.5
Wp = 0.6	1000	10	15	1.45%	1.5
		20	30	2.95%	1.5
Wq = 0.4		30	45	4.45%	1.5
	500	10	15	2.90%	1.5
		20	30	5.90%	1.5
		30	45	8.90%	1.5
	5000	10	22	0.44%	2.2
		20	46	0.91%	2.3
		30	69	1.37%	2.3
Wp = 0.7	1000	10	22	2.20%	2.2
		20	46	4.53%	2.3
Wq = 0.3		30	69	6.87%	2.3
	500	10	22	4.40%	2.2
		20	46	9.07%	2.3
		30	69	13.73%	2.3
	5000	10	37	0.74%	3.7
		20	77	1.54%	3.85
		30	117	2.34%	3.9
Wp = 0.8	1000	10	37	3.70%	3.7
		20	77	7.70%	3.85
Wq = 0.2		30	117	11.70%	3.9
	500	10	37	7.40%	3.7
		20	77	15.40%	3.85
		30	117	23.40%	3.9
	5000	10	82	1.64%	8.2
		20	172	3.44%	8.6
		30	262	5.24%	8.73
Wp = 0.9	1000	10	82	8.20%	8.2
		20	172	17.20%	8.6
Wq = 0.1		30	262	26.20%	8.73
	500	10	82	16.40%	8.2
		20	172	34.40%	8.6
		30	253	50.44%	8.43
Average				8.58%	4.02

7 Conclusion

The existing search engines all put the same emphasis on submitted query terms combined by a Boolean expression. However, users may put different emphasis on each query term. That is, users could be allowed to assign different weights to each query term. For such a case, a system must re-calculate the score of each web page according to the new given weights to provide users with their real ranking results. Besides, in reality, most users usually view the top K web pages from those HN returned web pages only.

Nowadays, a typical search engine does not return sufficient information for the system to immediately re-calculate the new score. The system must rescan all returned web pages to provide users with their top K interesting web pages; however, such a re-calculating task is very time-consuming. Based on the top K issue, we have proposed an efficient ESA method, which adopts the original score to derive the candidates of users' top K target ranks, to solve this problem. We, furthermore, improve ESA method and propose ESIA method in Section 6. Through the use of the proposed method, our system does not have to rescan any returned web pages but can inform users that their top K interesting web pages will be among the top R web pages ($K \leq R << HN$). When users want to get the real ranking result of their top K interesting target ranks, instead of re-calculating the whole HN returned web pages, our system jut re-calculates the top R returned web pages. The evaluated results in Section 4 have proved the efficiency of our ESA method. By the mathematical analysis in Section 6, our system can inform users how many web pages they have to view without any calculation.

However, in order to find the best ranking results, a user might be unable to decide the weights of each query term combined in a Boolean expression. For this problem, a "feedback agent" is considered in our future work. The feedback agent records users' feedback of a web page (relevant or not from the users' viewpoint) and analyzes the page's content to establish a user model. By the user model, the most suitable set of weights for each query is pre-determined instead of assigning by users. By incorporating this feedback mechanism, our system could provide the best, ranked top K target ranks. Furthermore, since a single search engine only provides an incomplete snapshot of the Web, we will extend our approach to *metasearch* [Aslam & Montague 2000, Dreilinger & Howe 1997, Etzioni & Selberg 1995, Gauch et. al. 1997, Gravano & Papakonstantinou 1998, Lawrence & Lee 1999, Liu 1999, Mishra et. al. 2000, Yu C. & Meng et. al. 1999]. The personal factor [Birmingham et. al. 1999, Fan et. al. 2000] will be also taken into account to further improve the efficiency of our system.

8 References

[1] Aguiar F. & Beigbeder M. (2000) Discovering the Context of WWW Pages to Improve the Effectiveness of Local Search Engines. *FQAS 2000*, p. 517-527.

[2] Aslam J. A. & Montague M. H. (2000) Bayes Optimal Metasearch: A Probabilistic Model for Combining the Results. *SIGIR 2000*. p. 379-381.

[3] Bergman L. & Castelli V. & Chang Y. C. & Li C.S. & Lo M.L. & Smith J.R. (2000) The Onion Technique: Indexing for Linear Optimization Queries. *Proceedings of the 2000 ACM SIGMOD on Management of Data*, p. 391 – 402.

[4] Birmingham W. P. & Glover E. J. & Lawrence S. & Lee C. G. (1999) Architecture of a Metasearch Engine that Supports User Information Needs. *Proceedings of the Eighth International Conference on Information Knowledge Management*, p. 210 – 216.

[5] Brin S. & Page L. (1999) The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of* the Seventh International Web Conference, Brisbane, Australia.

[6] Buttler D. & Liu L. & Pu C. & Paques H. (2001) Omini Search: An Internet Search Engine for Dynamic Web Pages. *SIGMOD 2001.*

[7] Chaudhuri S. & Gravano L. (1999) Evaluating Top-K Selection Queries. *VLDB'99*, p. 397- 410.

[8] Cho J. & Molina H. G. & Page L. (1998) Efficient Crawling Through URL Ordering. Proceedings of the *Seventh International Web Conference*, Brisbane, Australia.

[9] Dreilinger D. & Howe A. E. (1997) Experiences with Selecting Search Engines Using Metasearch. *ACM Transaction on Information Systems*, 15, 3, p. 195-222.

[10] Etzioni O. & Selberg E. (1995) MultiService Search and Comparison Using the MetaCrawer. *Proceedings of the* Fourth International Web Conference, Boston, USA.

[11] Fan W. & Gordon M. D. & Pathak P. (2000) Personalization of Search Engine Services for Effective Retrieval and Knowledge Management. *Proceedings of the Twenty-First International Conference on Information Systems*, p. 20-34.

[12] Gauch S. & Gomez M. & Wang G. (1997) Profusion: Intelligent Fusion from Multiple Distributed Search Engines. *Journal of Universal Computing*, 2, 9, p. 637-649..

[13] Gravano L. & Papakonstantinou Y. (1998) Mediating and Metasearching on the Internet. *IEEE Bulletin of Data Engineering*, 21, 2, p. 28-36.

[14] Ikeji A. C. & Fotouhi F. (1999) An Adaptive Real-time Web Search Engine. *Proceedings of the Second International Workshop on Web Information and Data Management*, p. 12-16.

[15] Introna L. & Nissenbaum H. (2000) Defining the Web: The Politics of Search Engines. *IEEE Computer*, 33, 1, p. 54-62.

[16] Kantrowitz M. & Mohit B. & Mittal V. O. (2000) Stemming and Its Effects on TFIDF Ranking. *SIGIR 2000*, p. 357-359.

[17] Kruger A. & Giles C. L. & Coetzee F. M. & Glover E. & Flake G. W. & Lawrence S. & Omlin. C. (2000) DEADLINER: Building a New Niche Search Engine. *Proceedings of the Ninth International Conference on Information Knowledge Management*, p. 272-281.

[18] Lawrence S., Lee C. G. (1999) Accessibility of Information on the Web. *Nature*, 400, p. 107-109.

[19] Lee D. L. & Yuwono B. (1996a) A World Wide Web Resource Database System. *IEEE Transaction on Knowledge and Data Engineering*, 8, 4, p. 548-554.

[20] Lee D. L. & Yuwono B. (1996b) Search and Ranking Algorithm for Locating Resources on the World Wide Web. *Proceedings of the Twelfth Int'l Conference in Data Engineering*, p. 164-171.

[21] Liu K. L. & Meng W. & Rishe N. & Wu W. & Yu C. (1999) Estimating the Usefulness of Search Engines. *Proceedings of the Fifteenth International Conference on Data Engineering*, p. 146 –153.

[22] Marchiori M. (1997) The Quest for Correct Information on the Web: Hyper Search Engines. *Proceedings of the Sixth International WWW Conference*, Santa Clara, USA.

[23] Mauldin M. L. (1997) Lycos: Design Choices in an Internet Search Service. *IEEE Expert*, p. 8-11.

[24] McGill M. & Salton G. (1983) *Introduction to Modern Information Retrieval.* New York: McGraw-Hill Book Company.

[25] Mishra R. K. & Prabhakar T. V. & Yantra K. (2000) An Integrated MetaSearch Engine with Classification, Clustering and Ranking. *IDEAS 2000*, p. 122-133.

[26] Ricardo B. Y. & Berthier R. N. (1999) *Modern Information Retrieval.* Addison-Wesley Longman.

[27] Schwartz C. (1998) Web Search Engines. *Journal of the American Society for Information Science*, 49, 12, p. 973-982.

[28] Sheldon M. A. & Vélez B. & Weiss R. (1996) HyPursuit: A Hierarchical Network Search Engine that Exploits Content-link Hypertext Clustering. *Proceedings of the Seventh ACM Conference on Hypertext*, p. 180 – 193.

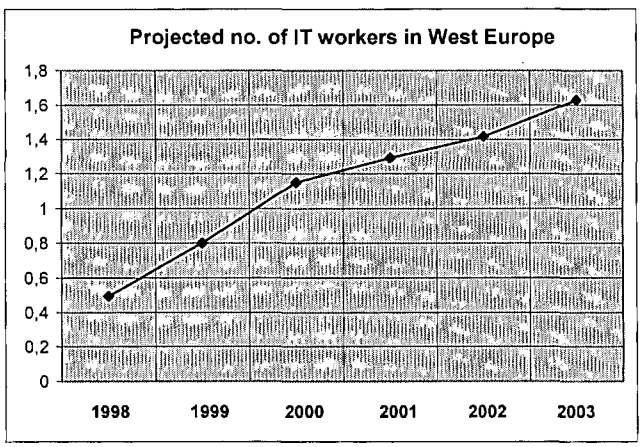
[29] Yu C. & Meng W. & Liu K. L. & Wu W. & Rishe N. (1999) Efficient and Effective Metasearch for A Large Number of Text Databases. *Proceedings of the Eighth International Conference on Information Knowledge Management*, p. 217 – 224.

Information technologies (Introduction to special issue)

Special Issue Editors: Cene Bavec, Matjaž Gams

Despite recent temporary seatbacks, information technologies (IT) are revolutionizing the way we understand and manage the environment in which we live and work. The exponential growth projected in the number of IT workers as suggested in Figure 1 declined slightly while other parameters - such as high-tech markets and shares - lost nearly half of their value in just a couple of months. Some other parameters, such as basic information society laws regarding the number of Internet hosts or growing speed of computer capacities, remained practically intact. The shocking events of September 11th have introduced another kind of mentality, but even those events have not severely hampered the progress of information society. For example, in the tragic days the Internet functioned nearly normally with some small exceptions, and e-mails are regarded much safer as before.

Figure 1: IT jobs.



The development of information society and the new economy is very fast with radical consequences on all parts of our society. Information society and the new economy are the greatest challenge for the human civilisation, countries, and above all the individual. Among major challenges we face today are:

- An incredible speed of changes and their apparent unpredictable nature;
- The lack of understanding of the impact that information technology has on all segments of our society, particularly at the top of decision making hierarchies;
- Limited social ability to absorb new technologies and information services, and inertia of humans and organisations.

This special issue is based on carefully selected papers from the multiconference Information Society 2001. Each of the selected papers was additionally extended and modified to meet the Informatica Journal criteria.

The papers deal with information technologies and their applications, which already play the crucial role in the economic and social development of Europe and Slovenia. But the gap between the most developed countries and others is widening even further and Slovenia unfortunately also lags behind the European average. That was one of the reasons for hosting a scientific meeting in the form of a multi-conference, which consisted of several conferences with specific themes covering some essential topics for the development of the information society.

The fourth international Information Society multiconference (http://is.ijs.si) was held in Ljubljana, Slovenia from October 22 to 26, 2001. There were over 100 lecturers and over 300 participants. The conference consisted of nine stand-alone independent conferences with their own agenda, their own programming committee and their own contributions:

- Collaboration and information society
- Data mining and warehouses
- Development and reengineering of information systems
- Education in information society
- Intelligent systems
- Management in information society
- Cognitive neuroscience
- Data mining and decision support in action
- New information technologies in fine arts.

The multiconference was organized together with INFOS, a major computer and informatics event in Slovenia. This demonstrates our intention to bring together science, applied research and business. Although the conference is of scientific nature with clear emphasis on scientific methods, the applied aspects of information society are clearly present.

The objectives of the Information Society multiconference were:

- Contribution and promotion of knowledge about the emerging information society and information technology;
- Interdisciplinary view on information technologies and information sciences;
- Promotion of key issues in forthcoming social and economic changes to politicians, government officials, industry and general public.

The central theme was the question: "What can we learn from computer and information science, and how can information and computer scientists support the transition to the information society through the search for new paradigms, new theories and new models?" We still have to strengthen the dialogue between managers, politicians and researchers, and we sincerely believe that the Conference and presented papers are a valid contribution to these efforts.

We thank all contributors, all program chairs and all committees for their efforts and achievements. We also thank our main sponsors, the Ministry of Education, Science and Sport, and the Ministry of Information Society.

Bibliography: Bavec, Cene (Ur.), Bernik, Mojca (Ur.), Bohanec, Marko (Ur.), Domanjko, Tomaž (Ur.), Dragan, Srečo (Ur.), Gams, Matjaž (Ur.), Grobelnik, Marko (Ur.), Heričko, Marjan (Ur.), Mladenić, Dunja (Ur.), Rajkovič, Vladislav (Ur.), Rozman, Ivan (Ur.), Solina, Franc (Ur.), Škrjanc, Maja (Ur.), Trček, Denis (Ur.), Urbančič, Tanja (Ur.). Information Society IS'01 : Proceedings A of the 4th International Multi-conference, 22-26th October 2001, Ljubljana, Slovenija. Ljubljana: Institut Jožef Stefan, 2001, str. 5-8. [COBISS-ID 16248103]

Classificatory challenge-data mining: a recipe

Steve Moyle and Ashwin Srinivasan
Oxford University Computing Laboratory,
Wolfson Building,
Parks Road,
Oxford OX1 3QD,
United Kingdom.
{sam,ashwin}@comlab.ox.ac.uk

Keywords: Data Mining, Challenge Data Mining, ROC Analyses, Optimal Models, Near-optimal Models

Received: November 2, 2001

A tested recipe for executing academic Classificatory Data Mining challenges is described. The recipe includes a way of assessing the quality of all submissions using ROC curves. The challenge methodology is focused towards selecting the best techniques *for the problem domain offered in the challenge. Modifications to the recipe so that the focus is on obtaining the best* model(s) *are discussed. The modified scheme could be used as the basis for a form of "collaborative" data mining.*

1 Introduction

Practices for evaluating the techniques used in Machine Learning (and also Data Mining) have continually evolved. These have included the widespread use of first *test sets*, then *cross validation*, and more recently *ROC curves* [6]. In the classification setting of machine learning such evaluation techniques have focused on assessing the accuracy of different algorithms or approaches – often with respect to some commonly used academic data sets [3].

The most important test of any data mining process is its application to real-world problems. Real world problems made publicly available for the purpose of comparing alternative data mining techniques are becoming common and are known as *challenge* problems (for example [8], [9], and [4]). This paper presents the challenge methodology that has been developed over the series of Predictive Toxicology Challenges (PTC) [8], which was recently employed in the PTC 2000–2001 [2].

The PTC Challenge recipe is composed of five main steps. The steps contain tasks which are common in other data mining approaches (for example *CRISP-DM* [1]). The aim of the challenge methodology was to evaluate data mining techniques for analysing toxicology data. The recipe can be used in other data mining challenge problems to allow the data mining problem owner to assess which is the "best" of the data mining techniques from those that are submitted to the challenge.

This paper is organised as follows. Section 2 outlines the PTC Challenge methodology as a step-by-step recipe. Section 3 discusses the recipe with respect to its aims as well as those of the CRISP-DM methodology. In section 4 an extension to the recipe is proposed to alter the framework from a *competitive* approach to a *collaborative* approach.

2 The PTC Challenge recipe

The setting that follows only relates to *classification* style data mining problems – that is those problems where the objective is to generate a *model* that is used to predict a discrete outcome from a set of input-attribute values. The agents that are involved in such a challenge setting are: the *challenge organisers* (or organisers), the *challenge participants* (or participants), and the *domain experts*.

- **Step 0:** ***Data Collection***
  The fist task it to find a willing problem owner with potentially interesting problems, and that these problems might be amenable to solution by data mining techniques. Having found such a situation a domain expert (or a panel of experts) is consulted to assist in the selection of a challenge problem. Once a problem is selected, then a problem specification must be produced, and an initial data set prepared.

- **Step 1:** ***Data Engineering***
  Data Engineering prepares the materials that are to be provided to all participants.

  1.1 Clean existing data sets. This task is the responsibility of the the challenge organisers.

  1.2 Separate *all* the cleaned data sets into *Training* and *Test* data sets (performed by the organisers).

  1.3 The data sets are augmented with new features. These features may be contributed by both the challenge organisers and the participants.

  Once this point in the challenge has been reached the training data-sets are *frozen* and then made available to all the registered challenge participants, along with all background and supporting materials.

- **Step 2:** ***Model construction***
  The challenge participants may apply any technique or techniques to the *training* data to produce classificatory models. The organisers make available to the participants the *test data set*. All challenge participants must submit one or more entries. An entry is considered as the set of the predicted outcomes (from the models) for the records of the *test* set.

- **Step 3:** ***Model Evaluation***
  The challenge organisers make some assessment of the entries.

  3.1 The organisers plot all submissions in ROC space (based on the test data). All optimal models M_{good} are identified. These will be those models that are on the convex hull of the ROC curve, or those models that are near-optimal.

  One way of defining *near-optimal* is those models that are within one standard deviation of the sample variance σ of the convex hull. For a binary classification model i where the number of observations to be classified is n, and the accuracy of the model's classifications with respect to the true classifications is p_i, and the error of the classifications is q_i then

$$\sigma_i = \sqrt{\frac{p_i q_i}{n}} = \sqrt{\frac{p_i(1-p_i)}{n}}$$

  3.2 Proposers of the models in M_{good} are asked to provide a text description of their models. These explanations are then evaluated by the domain experts.

- **Step 4:** ***Model Dissemination***
  The challenge organisers arrange for the dissemination of the results.

  4.1 A workshop is organised that is open to all interested parties. At the workshop the participants which were proposers of "good" models present the data mining approaches that were used to produce the models.

  Furthermore, and more importantly, the domain experts have an opportunity to provide their opinions.

  4.2 The results of the workshop are prepared for publication. Typically, the results would appear in a special issue of some journal in the domain from which the problem *originated*.

3 Discussion

3.1 PTC 2000–2001

The challenge methodology presented in the previous section was most recently used for *The Predictive Toxicology Challenge for 2000-2001* (see `http://www.informatik.uni-freiburg.de/~ml/ptc/`). The classificatory tasks were aimed towards predicting which chemicals behaved as carcinogens in four different biological settings: male and female mice, and male and female rats. At the conclusion of the data engineering step there were close to seven thousand attributes available.

The elapsed time for each of the steps performed for the PTC 2000–2001 were as follows: For *Step 0: Data Collection* approximately six months was taken; for *Step 1: Data Engineering* a further six months was allowed; *Step 2: Model construction* was restricted to three months; while *Step 3: Model Evaluation* and *Step 4: Model Dissemination* used two and a half months, and half a month respectively. The total duration of the challenge was eighteen months, with a sixth of the time allocated to model construction.

As an example consider the single task of predicting chemical carcinogenesis for male rats. There were fourteen participants, submitting a total of 31 entries. Each of the entries is plotted in ROC space in figure 1. The workshop to present and discuss the results was held in September 2001.

Overall, the results of PTC 2000–2001 were not particularly encouraging [5], with no clearly appropriate data mining techniques (from those submitted) that were suited to the problem domain. Feedback from the chemical toxicology domain expert(s) was not available at the time of writing.

This style of challenge is not really best suited to finding the best possible models for the problem domain. At best it indicates the techniques which are most likely to produce solutions to classificatory problems in the field of chemical carcinogenicity.

3.2 The Challenge methodology and CRISP-DM

The challenge methodology has some steps which are similar to the emerging data mining standard proposed in in the *CRISP-DM* methodology [1]. The *CRISP-DM* methodology specifies the following six steps: 1) Business Understanding; 2) Data Understanding; 3) Data Preparation; 4) Modelling; 5) Evaluation; and 6) Deployment. With respect to the challenge methodology it can be viewed that the challenge organisers have performed the tasks of *Business Understanding* and *Data Understanding* by selecting a problem and producing an initial data set in *Step 0: Data Collection*. There are further analogies – the effort expending in the challenge *Step 1: Data Engineering* is similar to that required in CRISP-DM's *Data Preparation* phase. The challenge *Step 2: Model construction* is clearly similar to the CRISP-DM *Modelling* phase.

The main contribution of the Challenge Methodology is the clearly defined evaluation criteria. Although evaluation is mentioned as part of CRISP-DM, specific evaluation

methodologies are not prescribed. In particular, CRISP-DM does not provide guidance on how to evaluate multiple solutions to the data mining problem.

For the challenge methodology the *Step 4: Model Dissemination* differs from the aims of the CRISP-DM *Deployment* phase. The *Deployment* phase is focused on how the models will be used within the business processes from which the data mining problem was chosen. The challenge methodology differs in this respect due to the nature of trying to assess *techniques* rather than specific *models*.

3.3 Using ROCCH to select *near-optimal* classifiers

The challenge *Step 3: Model Evaluation* described in section 2 mentions the selection of the optimal and *near-optimal* models. The objective of this process is to reduce the set of models to those that are likely to perform well in the problem domain. The interest is to discount the techniques that produce models that are unlikely to generate "good" predictive models. This section presents in more detail the use of ROC Convex Hulls (ROCCH) and the sample variance of each model to assess the *near-optimality* of the submissions.

For the "male rat" classification problem [2] participants were required to use their models to classify 185 instances from the test data set. The actual distribution of these instances was 133 chemicals that did not cause cancer in male rats (true negatives, n^-); while the remaining 52 did (n^+) did cause cancer. Clearly, the test set distribution was heavily skewed towards non-carcinogens.

One method of defining *optimality* is to consider the sample variance of each proposed model i with respect to the test set. It is possible to calculate this variance with respect to the both the true positive rate (produced by the model) σ_i^{TP} and the false positive rate σ_i^{FP}.

The variance of the false positive rate for model i is given by the following.

$$\sigma_i^{FP} = \sqrt{\frac{p_i q_i}{n^-}} = \sqrt{\frac{p_i(1-p_i)}{n^-}}$$

While The variance of the false positive rate for the same model is given by the following.

$$\sigma_i^{TP} = \sqrt{\frac{p_i q_i}{n^+}} = \sqrt{\frac{p_i(1-p_i)}{n^+}}$$

Each of the submitted models can be represented in ROC space with their sample variance represented as error bars. For the 31 submissions for the "male rat" challenge the ROC space is presented in figure 1. The figure illustrates that no "really good" models were submitted.

The ROCCH defines which of the models is "optimal" (with respect to the test data set). The "near-optimal" models are those that have an error bar that intersects the ROCCH. This allows the "non-optimal" models to be removed from study – as has been done for the "male rat" problem and shown in figure 2.

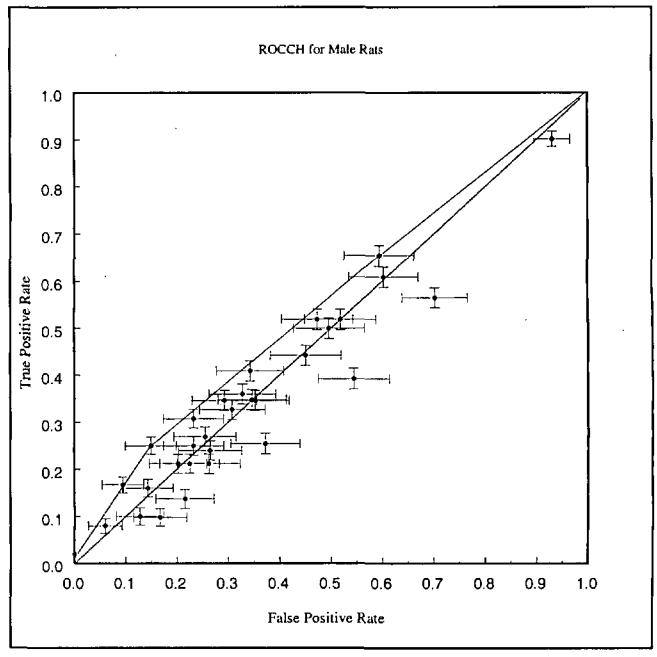


Figure 1: The ROC Convex Hull (ROCCH) for the 31 entries submitted to the PTC 2000-2001 for the "male rat" classificatory problem. The error bars represent the sample standard deviation for each entry.

4 "Collaborative" Data Mining

It is conceivable that models can be constructed that "raise" the ROCCH. Such a model could, for instance, be the result of some hybrid technique that uses techniques that have currently resulted in optimal (or near optimal) models. Such models would therefore be a result of a collaboration, which is alien to the "challenge" model.

So far this paper has considered the challenge recipe as applied in the most recent of the PTC challenges. This section focusses mainly on how the recipe may be altered to change the emphasis from *challenge* data mining to *collaborative* data mining. This is motivated by the requirements of the SolEuNet Project [7] where the objective is to pool the talents of data mining experts for use in the solution of real world problems – in a *collaborative* fashion.

The objective of collaborative data mining for real world problems has a different emphasis than that of the challenge data mining. For example, the the PTC 2000–2001 it was attempting to answer the following. *"For this type of problem and data what is the best data mining technique to use?"* In a sense any particular model has no value other than as an aid in evaluating and selecting the best techniques.

For collaborative data mining the interesting question is in the following. *"For this type of problem and data what is the best combination of data mining techniques to use that*

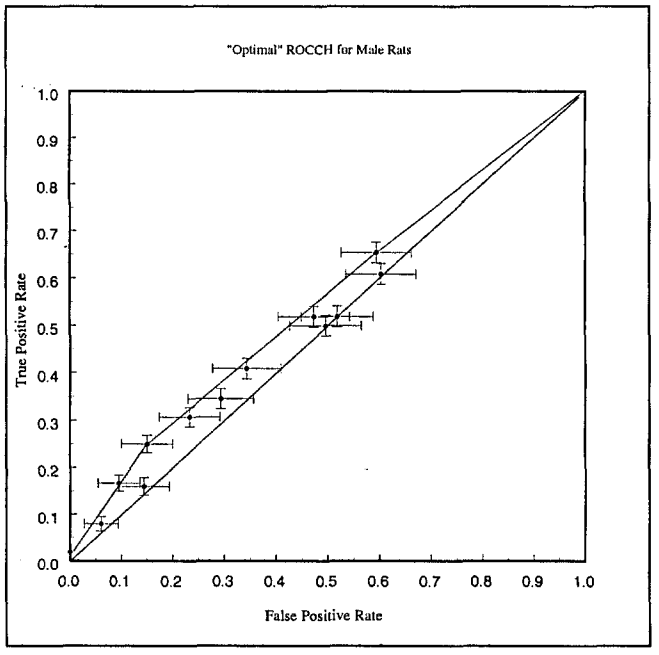


Figure 2: The ROC Convex Hull (ROCCH) for the "optimal" entries submitted to the PTC 2000-2001 for the "male rat" classificatory problem. The error bars represent the sample standard deviation for each entry.

produces the best model(s)?" The final objective being to produce the best model(s) for deployment in the business (or other) process.

Before proceeding further, a definition for *collaboration* in the context of data mining (or problem solving in general) is required. One operational definition of the occurrence of collaboration is when "*any published result – produced by one agent – is consequently used by other agents as input to produce another result*". For example results might be the suggestion of new features as a result of data pre-processing, or even complete models. In some sense models can be considered as "advanced" feature selectors.

If collaboration is defined in terms of the "sharing of results", it then pre-supposes that for collaboration to occur results must be available. With respect to the *challenge recipe* this "sharing" suggests that collaboration can only occur after models have been produced – that is after *Step 4: Model Dissemination.*

How then can the challenge recipe be modified and deployed for collaborative data mining? The following is an attempt to answer this question.

- **Step A:** ***Produce ROCCH and select Optimal and near-optimal models***
  This equates to performing challenge steps 0, 1, 2, and 3 – publishing all "results" to all participants as soon as they become available. Notice also that the ROCCH is produced by using the test data set.

- **Step B:** ***Combine models produced in step A***
  Try appropriate strategies for combining the models and the techniques which produced them. Notice no suggestion is provided here as to what "appropriate strategies" might be.

- **Step C:** ***Repeat Step A: Generate a new ROCCH***
  By repeating step A it is hoped that a convex hull closer to the ideal will be produced. Care must be taken not to over-fit the test set. Furthermore, it is possible that the new ROCCH may be worse than the first as the only accepted solutions are produced by combined methods that are better than the existing convex hull.

- **Step D:** ***Combined-Model Dissemination***
  The combined-model should be deployed into the business process

Potential difficulties with such a scheme arise with the cyclic nature and the potential for over-fitting the (test) data. One way of overcoming this problem will be to have further partitions of the data set for testing each cycle of collaboration.

5 Conclusions

This paper has detailed the recipe for a data mining *challenge* and described its application to a recent challenge. Evidence from this challenge and others demonstrates that Data Mining Challenges benefit from a sound methodology.

The use of ROC analyses is useful in ranking submitted models by selecting optimal entries. It was shown how ROC analysis can be extended to select optimal and near-optimal models (e.g. by incorporating notions of model variance). This is particularly important when the the submitted models are closely scattered along the ROC convex hull. In such a situation where there are no clear *winners* it might be useful to consider near-optimal models for further study, as well as the optimal models.

Suggestions of how to extend such a recipe to a *collaborative* setting were posed, but these have yet to be tested. Future work should consider techniques for *combining* models in a manner that improves the overall result. Care, however, should be taken to avoid over-fitting the data (both training and test).

6 Acknowledgment

The work reported here was supported in part by the EU project SolEuNet, IST-11495.

References

[1] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R.: *CRISP-DM 1.0: Step-by-step data mining guide.* CRISP-DM consortium, (2000).

[2] Helma, C., King, R.D., Kramer, S., and Srinivasan,A., editors. *The Predictive Toxicology Challenge 2000-2001.* Bioinformatics, 17:107-108, 2001. (Application Note).

[3] Murphy, P. M., and Aha, D. W.,. *UCI repository of machine learning databases.* `http://www.ics.uci.edu/~mlearn/`. Irvine, CA: University of California, department of Information and Computer Science.

[4] Page, C.D., and Hatzis, C. *KDD Cup 2001.* `http://www.cs.wisc.edu/~dpage/kddcup2001/`.

[5] Pfahringer, B. *(The Futility of) Trying to Predict Carcinogenicity of Chemical Compounds.* In Helma, C., King, R.D., Kramer, S., and Srinivasan,A., editors. *The Predictive Toxicology Challenge 2000-2001.* Proceedings of ECML/PKDD PTC Workshop 2001, Freiburg, 2001.

[6] Provost, F. and T. Fawcett, *Robust Classification for Imprecise Environments.* Machine Learning 42, 203-231, 2001.

[7] SolEUNet: Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise. `http://soleunet.ijs.si/`.

[8] Srinivasan,A., King, R.D., Muggleton, S.H., and Sternberg, M.J.E.. *The Predictive Toxicology Evaluation Challenge.* In Proceedings of the Fifteenth International Conference on Artificial Intelligence (IJCAI-97). Morgan Kaufmann, Los Angeles, CA, 1997

[9] van der Putten, P., and van Someren, M., editors. *CoIL Challenge 2000: The Insurance Company Case.* Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22, 2000.

Security authentication for on-line Internet banking

D. Hutchinson and M.J. Warren
School of Computing and Mathematics, Deakin University,
Victoria, 3217, Australia
Email contact: damienh@deakin.edu.au

Keywords: E-Commerce, Computer Security, Internet Banking, Authentication.

Received: May 13, 2001

The advent of Internet Banking has shown the importance of effective method of authenticating a users in a remote environment. There are many different countenances to contemplate when examining Internet based security. One of the most tried and trusted techniques of protecting the safety of systems and data is to control people's access. The foundation for such measures is authentication. Specifically for Internet banking there is a real need for a way to uniquely identify and authenticate users without the possibility of their authenticity being cloned. This paper proposes a framework concerning how to identify security requirements for Internet Banking.

1 Introduction

E-commerce fundamentally focuses on the electronic exchange of information using information and telecommunication infrastructures (particularly the World Wide Web and the Internet). E-commerce encompasses a wide range of commercial activities that can be categorised into business-to-consumers and business-to-business sectors. Industry sectors such banking have openly embraced E-commerce to improve their performance and gain a strategic competitive advantage. But many customers are wary to bank through their computer. Hardly a month goes by without the launch of an Internet bank. Some are new banks while others are the descendants of well-established banks. The new banks promise a lot. They are built on advanced and secure technology so customers can bank with total confidence and promise security consumers can trust and are concerned how safe is on-line banking?

However there have been numerous incidents that point to gaping security holes.

- When online bank Egg upgraded its website in May 1999, new security measures scrambled online session protocols and allowed users to see banking details of other customers [1].
- In November 1999, the UK Halifax bank suspended its online share dealing service, after an attempt to fix a bug backfired. Again, account details were made available to other users [1].
- Barclays, which claims to be the UK's largest online bank, had to take down its Web site at the end of July, when customers were served the bank statements of other clients [4][5].
- Western Union, part of Atlanta-based electronic payments giant First Data Corp., implemented a new Internet-based person-to-person payment service but the site was hacked during a routine maintenance operation that erroneously left parts of the site exposed [2]. A report by Associated Press detailed how hackers broke in and copied the credit card and debit card details of 15,700 Western Union customers who use the site to transfer funds across the Internet [3].

This paper is concerned with the service of Internet Banking and the issues surrounding authentication, which is the mechanism at the heart of E-commerce security. The content draws a correlation between the concepts depicted in figure 1, by presenting a framework that when applied to certain Internet Banking scenarios can offer the customer guidelines regarding the implementation of appropriate authentication mechanisms to ensure an adequate level of trust between the parties conducting the transaction.

2 The framework of authentication for Internet banking

The proposed model for identifying security requirements in an Internet Banking environment is intended to support the use in both business-to-business and business-to-consumer E-commerce.

Organisations, small to medium sized enterprises (SME's) and home-based customers shall be able to use this framework as a guide to identifying the security requirements for their particular banking

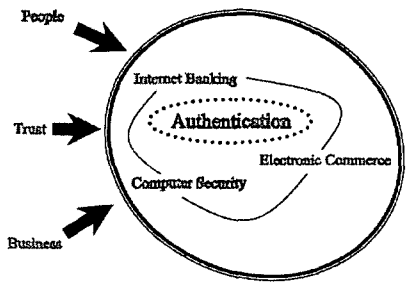


Figure 1: Security Issues of Internet Banking

environment. The objective of the scenario presented is to encourage a sense of confidence in the parties involved in undertaking Internet Banking transactions, such that there personal information is protected from prospective security breaches, alike to when Barclays, which claims to be the UK's largest online bank, had to take down its Web site at the end of July 2000, when customers were served the bank statements of other clients [4][5].

2.1 Steps in the Framework Process

Traditionally security assessment has been undertaken by applying conventional risk analysis methodologies. These have become inadequate with the advent of open, distributed networks, requiring new approaches to risk assessment. Thus, the following framework aims to identify the security requirements for an Internet Banking environment. Developed from a framework for E-commerce security [6], it consists of a defined six-step process.

Step 1 List all the security requirements for an Internet Banking environment in general;
Step 2 Identify all participants and stakeholders involved in the Internet Banking process;
Step 3 Break down transactions into different autonomous actions;
Step 4 Map these identified actions onto the participants involved, which serve as a model for the Internet Banking environment;
Step 5 Use the information obtained in step 4 to determine the security requirements for a secure Internet Banking environment and
Step 6 Use these security requirements to develop the security architecture, comprising suitable security procedures, mechanisms and policy.

Each one of these steps is further examined in the following sections.

3 Security requirements for an Internet Banking environment in general

The close relationship that exists between E-commerce and Internet Banking means that an Internet Banking session must satisfy the same security requirements as listed below:

1.Identification and authentication - The ability to uniquely identify a person or entity and to prove such identity;
2.Authorisation - The ability to control the actions of a person or entity based on its identity;
3.Confidentiality - The ability to prevent unauthorised parties from interpreting or understanding data;
4.Integrity - The ability to assure that data has not been modified accidentally or by an unauthorised parties;
5.Non-repudiation - The ability to prevent the denial of actions by a person or entity;
6.Availability - The ability to provide an uninterrupted service;
7.Privacy - The ability to prevent the unlawful or unethical use of information or data;
8.Auditability - The ability to keep an accurate record of all transactions for reconciliation purposes.

These eight security requirements have been proposed as the basis for the E-commerce security framework [6]. To extend on this, authentication mechanisms need to be incorporated to provide the corner stone of authentication for the Internet Banking framework. This would comprise the use of passwords, smart cards and possibly biometrics.

The following section describes where these security requirements fit into an Internet Banking environment.

4 The Internet Banking Environment

There are three main areas of security that are involved in Internet Banking. These are:

- The Bank;
- The Internet;
- The User's (customer) Computer.

The user's computer includes both a home customer as well as organisational customer using Internet Banking facilities.

The interaction between bank-to-bank Internet Banking has been omitted due to the discrepancy of different and advanced level of security provided at this level.

Figure 2 illustrates a simplified version of the Internet Banking transaction process through an Internet Banking environment where a customer/business wishes to pay a bill.

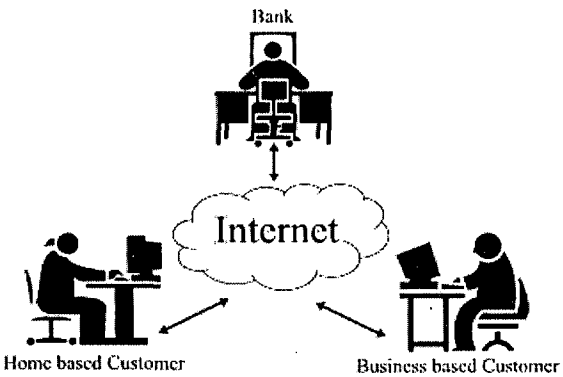


Figure 2: Internet Banking environment

In figure 2, there are two participants to any Internet Banking transaction, namely the customer that can either be home or business based and the bank. The important considerations like taxation and legislation across geographical borders have been omitted from the discussion for the sake of simplicity.

5 Description of the spheres

Four spheres can be determined from figure 3. Each has its own unique security requirements based on the Internet Banking requirements outlined later. A sphere is defined as an independent entity consisting of a person, information technology or both. Figure 3 shows a representation of these spheres.

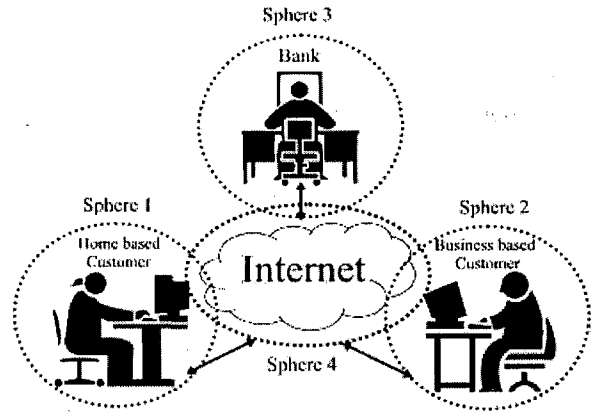


Figure 3: Spheres within Internet Banking environment

The following section defines and describes each individual sphere within the Internet Banking environment as depicted in figure 4.

5.1 Sphere 1 - Home based Customer

This customer can be any home based user on the Internet. It is therefore not viable to determine or assume that such a customer has any security (authentication) mechanisms in place. The only assumption that can be made is that most home-based customers would use browsers that support digital certificates and the Secure Socket Layer (SSL). However it cannot be presumed that these customers have the knowledge or understanding of how to use this integrated functionality.

The nature of Internet Banking is such that the majority of home-based Internet users should be seen as potential customers and hence should not be prevented or hindered in any way from participating in an Internet Banking transaction. Thus for these customers, the partaking in an Internet Banking transaction should cater for secure and user-friendly operations in a convenient environment.

5.2 Sphere 2 - Business Based Customer

This customer can be any business-based company on the Internet. The major difference between a home-based and business-based customer would be the implementation of some form of security mechanisms.

There are two distinct relationships that exist in this instance. Firstly the association of the business-based customer with the bank that represents a similar relationship as a home-based customer and secondly where the business-based customer acts as

the merchant between the home-based customer and the bank. In the second scenario, the merchant accepts the responsibility for securing the transaction with the home-based customer before forwarding it to the bank. The merchant must, therefore provide assurance that an electronic transaction can be made safely and securely and that the risk has been minimised to an acceptable level for all participants.

In terms of the nature of Internet Banking, business-based companies are entitled to the same inclusions as home-based banking.

To maintain a level of simplicity, the electronic business environment, which comprises knowledge management and workflow, does not form part of the proposed framework, although it would be possible to adopt the framework for this environment.

5.3 Sphere 3 - Bank

The framework regards the inter-network of banks as a single body as opposed to each bank being its own separate entity. The purpose of the banking sphere is twofold; firstly to validate customers through authentication mechanisms and secondly to authorise and honour transactions to ensure against non-repudiation.

5.4 Sphere 4 - Internet

The Internet is considered to be a network of networks where there is no one single entity responsible for security or held accountable for any losses suffered. It is viewed as the infrastructure that facilitates global communication, leading to E-commerce and now Internet Banking. From its outset, the Internet in no way has existed to protect any of the participants but rather to provide a channel to facilitate the connection between different entities wishing to communicate via electronic means. Despite version 6 of the Internet Protocol (IPv6) being successfully proven in various test environments, version 4 (IPv4) is still the chief Internet protocol. Adversely IPv4 is absent of the security functionality included within IPv6. Thus, the security of a message cannot be taken for granted.

6 Autonomous actions contained within the Internet banking transaction

Up to now the participants and the relationship between them have been explained. The next step is to analyse and divide the transaction into smaller, autonomous actions that combined make up a complete Internet Banking transaction. A typical Internet Banking transaction consists of the following actions as illustrated in figure 4. Note that home-based and business-based customers are interchangeable.

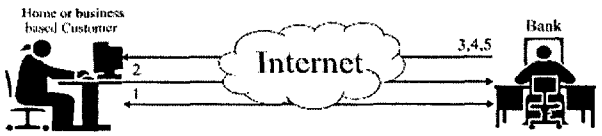


Figure 4: Autonomous actions contained within Internet Banking transaction

Action 1: - A customer uses the Internet to connect to their bank's Web site;

Action 2: - The customer browses the Web site and decides on a service. An Internet Banking transaction is initiated by the customer by providing both invoice and payment information;

Action 3: - The bank checks if the transaction is executable by verifying the customer has enough funds available and a reply is returned to the customer;

Action 4: - Upon completion of the transaction confirmation is sent to the customer;

Action 5: - The bank honours the payment and returns proof of having done so.

These transactions could be broken down further if deemed necessary. A decision table can then be used to assist in the identification of the essential security requirements for the Internet Banking environment as previously described.

7 Security decision analysis

The decision table as shown in table 1 illustrates an example based on the scenario outlined previously. A brief discussion of the steps to develop such a decision table is also provided.

		Step 3 Actions								
	Spheres	A1	A2	A3	A4	A5				
Step2	Customer	X	X		X					Step 4 Mapping
	Internet	X	X							
	Bank	X		X		X				
	Security needs									
Step 1	1	X	X							Step 5 Security framework
	2			X						
	3		X		X	X				
	4		X		X	X				
	5		X		X	X				
	6				X					
	7	X								
	8			X		X				

Table 1: Decision Table

Following is a brief description of each of the steps.

Step 1: - consists of listing all the security requirements that must be satisfied as discussed previously;

Step 2: - consists of listing the spheres that have been identified. Only the three spheres shown in figure 3 are used i.e. bank, Internet and customer, whether home or business based;

Step 3: - comprises listing the actions that make up a transaction. The five actions previously identified are used in the decision table;

Step 4: - maps the actions onto the spheres identified in step 2. Naturally not all the actions will include all the spheres;

Step 5: - associates the security requirements for an individual action.

This information is then applied to establish how it can be implemented within the relevant sphere.

In the decision table, action 1 shows that the bank must be able to identify and authenticate a customer satisfactorily to perform a transaction. At the same time the customer wants privacy regarding the personal account information being viewed. Privacy in this context refers to this information being unavailable to other parties. Actions 4 and 5 require the bank to send confirmation of the transaction and to ensure confidentiality and integrity of this message. Concurrently, the customer wants a guarantee that the bank cannot later deny that the transaction took place. This refers to the security requirement of non-repudiation. The table also indicates that the bank needs to record the transaction correctly in order to meet auditing requirements.

By looking at the security requirements for each action, it is possible to identify the security mechanisms required to secure the Internet Banking environment. For action 1, the identification and authentication security requirement could very well be facilitated by the implementation of a smart card authentication system possibly with an accompanying biometric mechanism. The required infrastructure through developed standards and technological know how has already been established for smart cards, providing certain support for this initiative. The security requirements for action 2 might suggest that SSL be used for securing the communication session (currently being used with 128-bit encryption) across the Internet. This may only be an interim approach or for the long term, depending on the implementation and widespread adoption of version 6 of the Internet Protocol (IPv6). Nevertheless it would be imperative to conduct timely checks on the protection provided by 128-bit encryption, with the high likeliness that it will be broken in the near future. The security requirements for actions 4 and 5 may be satisfied using SSL, although the acknowledgement needs to be digitally signed by the bank to conform to the security requirement of non-repudiation for all transactions. This would be catered for by the use of digital certificates.

8 Validation

For the purpose of this demonstration, the following provides an evaluation of one of the identified scenarios based on the developed framework constructed previously [7].

The first evaluation is based on the consumer-to-business E-commerce environment depicted in figure 5.

Figure 5: Consumer Scenario (Cell Phones, PDAs)

In this scenario the areas that must be secured include:

- The consumer;
- The terminal (cell phone or PDA);
- The wireless and public network (telecommunication exchange);
- The Internet (communication server) and
- The Bank.

This is represented in figure 6 below that also illustrates the autonomous actions contained within this particular environment.

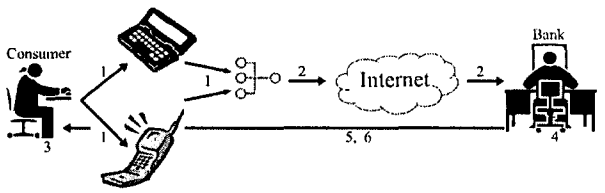


Figure 6: Autonomous actions contained within the cell phone, PDA scenario

Following is an explanation of these actions.

Action 1 - consumer uses a mobile phone or personal digital assistant (PDA) to connect to a wireless or other public network.

Action 2 - Through the telecommunication exchange and Internet, the consumer is able to connect to their bank's Web site.

Action 3 - The consumer browses the Web site shown on their cell phone or PDA screen and decides on a service; i.e. Transfer funds from account A to account B.

Action 4 - The bank checks the validity of the consumers' request.

Action 5 - The bank sends the confirmation to the consumer upon completion of the transaction.

Action 6 - The bank honours the transfer and returns verification to the consumer.

From this the decision table can be derived as shown in Table 2.

		Step 3 Actions									
	Spheres	A1	A2	A3	A4	A5	A6				
Step2	Customer	X	X	X			X			Mapping	Step 4
	PDA	X		X			X				
	Wireless network	X	X	X			X				
	Internet		X			X	X				
	Bank		X		X	X	X				
	Security needs										
Step 1	1		X							Security framework	Step 5
	2			X							
	3	X	X	X	X	X		X			
	4	X	X	X	X	X		X			
	5			X	X	X		X			
	6	X									
	7				X			X			
	8					X					

Table 2: Scenario Decision Table (Derived from Table 1)

By viewing the security requirements for each action, the security mechanisms required to secure this Internet Banking environment can be suitably determined. For example confidentiality can be assured by a smart card acting as a veritable lock between the secret code on the chip and the unsecured terminal (in this case the cell phone, PDA, and telecommunication exchange) environment. In addition authentication can be provided for via the use of a PIN as well as an integrated digital signature and digital certificate associated with a smart card system. Further data integrity can be catered for via the use of Message Authentication Codes that are in-built into the Secure Socket Layer (SSL), which can be used for securing the Web session over the Internet. To

prevent Internet based users from breaching the banking network, a firewall should be implemented to isolate the Web server from the customer information database. Finally, by complementing the identification and authentication process of Internet Banking based transactions with technologies like public-key cryptography, digital notary and digital signature, repudiation of transactions is protected.

9 Conclusion

The entities involved in the transaction including the technological components are clearly defined and arranged accordingly. Naturally the various entities will require different security requirements based on their interaction within the specified Internet Banking environment. The model caters for this determination by providing a detailed decision table that amalgamates all the information gathered in the six-step process. This valuable cross-referencing method ensures that all avenues from whence contingencies arise are covered.

The framework of authentication for Internet Banking allows customers to work their way through each step, identifying the necessary security requirements along with the counteracting authentication mechanism. The distinctive style of the framework including explicit descriptions, examples and cross-referencing capability ensures all security requirements and authentication mechanisms are sufficiently identified for correct and effective implementation.

References

[1] BBC (2000) *Safety fears for web banking*, BBC News Tuesday, 1 August, 2000 http://news6.thdo.bbc.co.uk/hi/english/business/newsid_861000/861353.stm (accessed 8 August 2000).

[2] Creed, A (2000) Western Union Site Down After Theft Of Credit Card Details, *Newsbytes*, Englewood, Colorado, USA, September 10.

[3] Gutterman, S (2000) Western Union Web Site Is Hacked, *Associated Press*, September 10.

[4] Knight, W (2000) Barclays security breach forces online service to close, *ZDNet UK*, Monday 31 July2000. http://www.zdnet.co.uk/news/2000/30/ns17002.html (accessed 1 August 2000).

[5] Knight, W (2000a) Barclays in security gaffe this Week, *ZDNet UK*, Wednesday 02 August 2000. http://www.zdnet.co.uk/news/2000/30/ns17040.html (accessed 3 August 2000).

[6] Labuschagne, L (2000) *A new approach to dynamic Internet risk analysis*, Thesis (D.Com) – Rand Afrikaans University, South Africa, 2000. http://csweb.rau.ac.za/deth/acad/thesis/ (accessed 16 August 2000).

[7] Hutchinson, D and Warren M.J (2001). A Framework of Security Authentication for Internet Banking, *Information Society 2001*, October, Ljubljana, Slovenia.

Insights offered by data-mining when analyzing media space data

Maja Skrjanc, Marko Grobelnik, and Darko Zupanic
Jozef Stefan Institut, Jamova 39, Ljubljana, Slovenia
Maja.Skrjanc@ijs.si, marko.grobelnik@ijs.si, darko.zupanic@ijs.si

Keywords: data mining, media space data, data analysis

Received: June 30, 2001

Media space consists from many different factors fighting for the attention of customer population in a certain environment. Common problem in bigger environments (or countries) is that datasets describing complete media space is hard or almost impossible to get since the detailed picture is too complex or to expensive to compose. However, this is not the case in environments, which are smaller, and is therefore easier to collect the data. We have access to the data entirely describing the media space of population of 2 million people. Because of the language and economy this media space functioning relatively independently from different factors, specially outside the country. The data was collected by Media Research Institute Mediana. The database consists of 8000 questionnaires, gathered in 1998. The sample and the questionnaires were made by comparable research standards. In this paper we will discuss different type of questions, which might become in a great assistants in unfolding the media groups, audience fluctuation and profound understanding of happenings in media space, as well as for their predictions.

1 Introduction

New emerging technologies enable more transparent communication between the media and the audience. As the response time for information feedback is becoming shorter, the general public can be more involved in the process of shaping the media. For the same reason measuring the media impact on the public is becoming much easier. Information about the media space, dynamics, interactions between the public and media are regularly monitored, collected and analyzed. Those type of information and knowledge raise different kinds og questions, which were also addressed in several other analysis [9,10]. Knowledge or new information extracted from the data is a value added information, which in this highly competitive environment represents a crucial factor.

One of the possible approaches to get additional value from the gathered information is the use of the data mining techniques, which can not only contribute to the deeper data analysis, but can also create new additional services providing new insights into the data. Although Slovenia is a small media space is also very specific because of the language, it is not an exception in comparison to some other media environments.

2 The Data

In this section we will describe the contents, structure and quality of the analyzed data set.

For the purpose of our analysis we took one of the Mediana's data sets, describing the whole Slovenian media space. The data was gathered by comparable international research standard. The data set consists of about 8000 questionnaires tracking all of the important reading, listening, and watching media in Slovenia. Each questionnaire consists of about 1100 questions collected in groups: about the person's relation to the certain media, person's activities, interests, life style, income, property, and demographic data. The relation of a person to the certain media is checked in more details with

Since 1992 Media Research Institute, Mediana (http://www.irm-mediana.si/) follows all printed, TV and radio media in Slovenia. They are trying to unfold and explain Slovenia's media image by collecting all kind of data about the media and analyze them with simple statistical methods. As a part of the Sol-Eu-Net project (http://SolEuNet.ijs.si/) we came into the position to analyze the data with more sophisticated data mining methods and present our outcomes to Mediana.

In this article we will answer to some selected questions from the large space of interesting questions, arising from the need for better understanding of the media space.
In the first section we will describe the quality, structure and contents of the data set. The second section will define the selected questions. In the third section we will describe our experiments by the methods we used to answer the questions. The answers are supported by many comprehensible examples of the rules and trees. At the end, in the fourth section we will describe the Mediana response and show some directions for the future work.

different questions testing the several aspects of using the media.

The first page of the poll contains general questions about followed by 19 pages of specific questions. The most of the questions are asked in the way that the answer consists from a grading scale with 2, 5 or 9 levels. The data set is presented in a spreadsheet table, were each questionnaire represents a row and each question represents a column. In general, Mediana's dataset if of rather high quality meaning that we didn't have much work cleaning and transforming the data.

3 Questions

Mediana didn't give us any specific questions; they just gave us a challenge to find something interesting in their data, which might be of any interest to them. Therefore, we had to think of some questions, which Mediana would find interesting in a way that the answers or techniques would represent a possibility for offering an additional commercial service. The requirements for the answers and resulting models to the selected questions were comprehensibility especially in the comparison to the classical statistical methods, which they have already used.

For our analysis, we selected the following questions:

- ⇒ Which printed media are also read by the readers of a selected newspaper/magazine?
- ⇒ What are the properties of readers/listeners/watchers of the specific media?
- ⇒ Which of these properties distinguish between the readers of different newspapers?
- ⇒ What are typical groups of people according to their personal characteristics?
- ⇒ Which media are similar?

4 Experiments

4.1 Methods

To answer the selected questions we used the following methods:

- o Correlation based clustering of the attributes
- o Association rules (Apriori) [4]
- o Decision trees (C4.5 , C5) [5]
- o Clustering (K-Means) [7]
- o Kohonen Network [6]

Our goal was to find some relations within the data, which are not obvious at the first sight and are comprehensible to Mediana. Due to that, our results do not optimize accuracy, but comprehensibility. Usually, our interpretation of the results does not rely only on the particular rule, but it is generalized over a group of similar rules. By that, we can offer better interpretation and generalization.

4.1.1 Clustering of the attributes

In this section we discuss the relation between the attributes.

To determine the dependencies between the attributes we used clustering, where the distance was the correlation coefficient between the attributes. As the result we got clusters of attributes, which are correlated with the correlation coefficient above a certain threshold (0.5). Some of the resulting groups collect different attributes describing several aspects of the same media. For the most of the groups we were able to provide very comprehensive explanation. Some other groups consist of the attributes having very obvious relationships, like the group from EXAMPLE 1. This group is composed of the attributes dealing with the same media. Questions, which correspond to this particular attributes are: (1)Did you read magazine Golf in the last year (BMERead_Golf)? (2)How many issues did you read in last 12 months (BMEIssues_Golf)? (3)How long ago did you read your last issue (*BMELastRead_Golf)*?

EXAMPLE 1

Correlations between different aspects of the same media.

Attributtes:	BMERead_Golf
	BMEIssues_Golf
	BMELastRead_Golf

Another type of clusters represents attributes with the high correlations between region and all or most of the editions of the same newspaper company. In particular, case in the EXAMPLE 2, the newspaper company Vecer is a local newspaper company. *Vecer* is the main daily newspaper and *Vecerov Cetrtek, Vecer Televizija in Radio, Vecer v Soboto* are its supplements. We can see that similar groupings as in the EXAMPLE 1 joined with some demographic attributes like region, community and local community. They emphasize the local influence of this media.

EXAMPLE 2

Attributtes:	*REGION*
	COMMUNITY
	LOCAL_COMMUNITY
	QUESTIONER
	DERead_Vecer
	DESRead_Vecerov Cetrtek
	DESRead_Vecer.Televizija,Radio
	DESRead_Vecer v Soboto
	DEIssues_Vecer
	DESIssues_Vecerov Cetrtek
	DESIssues_Vecer,Televizija,and Radio
	DESIssues_Vecer v Soboto
	DELastRead_Vecer
	DESLastRead_Vecerov Cetrtek
	DESLastRead_Vecer Televizija, Radio

DESLastRead_Vecer v Soboto

Another type of clusters stress out the correlations between the attributes describing the person's age and spare time activities, which are part of a life style group of questions. EXAMPLE 3 presents the group of attributes describing the persons age and type of spare time activities: How often are you going to the cinema (*spareTime_Cinema*), Do you study in your spare time (*spareTime_Study),* How often are you listening to CDs, LPs *(spareTime_LP/CD)?* Are you speaking English *(languagesUnderstanding_English)?* When was the last time you were in the cinema (*cinema*)? This type of clusters also point out that part of the person's life style is age dependent.

EXAMPLE 3

Attributes: *spareTime_Cinema*
spareTime_Study
cinema
basicDescriptions_BirthYear
delivered_Age
spareTime_LP/CD
languagesUnderstanding_English

An interesting but not unexpected type of clusters represents EXAMPLE 4. It presents the correlations between spare time activities and items possession. The person who is interested in the science (*mediaInterest_Science*) and the computer science (*mediaInterest_ComputerScience*) and is in a spare time is using a computer (*spareTime_Computers*) has very likely a computer at home (*householdItemsHas_PersonalComputer*), as well as modem, internet access, video, CD ROM (*householdItemsHas_Modem, householdItemsHas_InternetAtHome, householdItemsHas_Video/ComputerGames, householdItemsHas_CD ROM2).*

EXAMPLE 4

Attributtes: *spareTime_Computers*
mediaInterest_ComputerScience
householdItemsHas_PersonalComputer
mediaInterest_Science
householdItemsHas_CD ROM
householdItemsHas_Video/Computer Games
propertiesHas_PersonalComputer
householdItemsHas_Modem
householdItemsHas_Internet at Home

Similar type of clusters as the one from EXAMPLE 2 is EXAMPLE 5. It also includes correlations from EXAMPLE 1. Here we can observe that all media from a certain province are highly correlated. Dolenjski List, TV Novo Mesto, Radio Krka, Studio D, Radio Sraka are media from the province Dolenjska. People obviously like to keep track of local events, which are covered mostly from local media.

EXAMPLE 5

Attributtes: *WERead_Dolenjski list*
WEIssues_Dolenjski list
WELastRead_Dolenjski list
televisionsWatched_TV Novo mesto
televisionsLastWatch_TV Novo mesto
televisionsDays_TV Novo mesto
radiosLastListen_Radio Krka, Novo mesto
radiosLastListen_Studio D
radiosDays_Radio Krka, Novo mesto
radiosDays_Studio D
radiosListened_Radio Krka, Novo mesto
radiosListened_Radio Sraka - Novo mesto
radiosListened_Studio D

4.1.2 Association rules

Association rules [4] are part of the standard selection of data mining techniques. We used them (as implemented in [1] and [2]) to answer the two questions: (1)how readers of one daily newspaper are related to the readers of other newspapers and (2)what is the relation to all other attributes.

These questions were tested on the whole set of attributes, except the question about relations between different newspapers, which was tested only on the chosen newspapers attributes.

Surprisingly, the answers of both tests were almost identical, which gave us an idea, that reading certain newspaper is very much dependent on reading some other newspapers. The exceptions were two local daily newspapers, which are very region dependant.

Lets look at some examples of association rules. EXAMPLE 6 and EXAMPLE 7 present the comprehensible interpretation of the rules. Attributes in the EXAMPLE 6 and the EXAMPLE 7 correspond to the question about the reading certain magazines and newspapers in the last year or in the last six months. The number at the end of the left side of the rule represents number of covered examples (support) and numbers on the right side of the rule represent the number of covered examples (confidence).

As the results we got rules, which uncover the relations between different publications. These relations were very interesting – particular because of the nature of the topics these publications mainly cover. In the EXAMPLE 6 we got the rule, which associate readers of the biggest Slovenian daily newspaper Delo, with readers of magazines (Marketing magazin, Finance, Razgledi, Denar, Vip), which are covering mainly economics and marketing topics.

In the EXAMPLE 7, the rules show the connection between the readers of Slovenske Novice and the publications, which are known more as a 'yellow press' (Sara, Ljubezenske zgodbe, Omama). They cover mostly

romantic and erotic topics. This is not surprising since Slovenske Novice is known as a kind of yellow press daily newspaper. It is the most read newspaper in Slovenia.

EXAMPLE 6.
Interpretation: "The majority of the readers of any of the following publications:
Marketing magazin, Finance, Razgledi, Denar, and Vip are also readers of Delo."
Rules: DERead_Delo

1. MERead_Marketing magazin (sup.=116) ==> DERead_Delo (conf.=0.82)
2. WERead_Finance (sup.=223) ==> DERead_Delo (conf.=0.81)
3. BWERead_Razgledi (sup.=201) ==> DERead_Delo (conf.=0.78)
4. BWERead_Denar (sup.=197) ==> DERead_Delo (conf.=0.76)
5. MERead_Vip (sup.=181) ==> DERead_Delo (conf.=0.74)

EXAMPLE 7.
Interpretation: "The majority of the readers of any of the following publications:
Sara, Ljubezenske zgodbe, Dolenjski list, Omama, and Delavska enotnost
are also readers of Slovenske novice."
Rules: DERead_Slovenske novice

1. BWERead_Sara (sup.=332) ==> DERead_Slovenske novice (conf.=0.64)
2. WERead_Ljubezenske zgodbe (sup.=283) ==> DERead_Slovenske novice (conf.=0.61)
3. WERead_Dolenjski list (sup.=520) ==> DERead_Slovenske novice (conf.=0.6)
4. MERead_Omama (sup.=154) ==> DERead_Slovenske novice (conf.=0.58)
5. WERead_Delavska enotnost (sup.=177) ==> DERead_Slovenske novice (conf.=0.58)

4.1.3 Decision Trees

Decision Trees [5] are also part of the standard repertoire of the data mining and machine learning techniques. We used C4.5 (as implemented in [1]) and C5.0 (as implemented in [2]) to describe the characteristics of the readers reading certain daily newspaper. Another question that we tried to answer was how readers of one daily newspaper differ from the readers of the other daily newspaper.

With the decision trees we get the most natural and understandable interpretation for these problems. We try to identify typical description characteristics of the readers, including their life style, life statements, capability of trademarks recognition, their interests, etc. After putting together the rules and their interpretations, we got description of typical reader for every daily newspaper. Parts of descriptions correspond to the characteristics, Mediana already did with statistical methods, but some of the descriptions include very interesting personal characteristics. At first sight these rules seem unreasonable, but after some more careful analysis and discussions with the experts a perfectly reasonable explanation could be found. Those rules represent the most valuable results. We could observe that effect especially in the EXAMPLE 9, where we describe readers of the newspaper Slovenske novice.

While testing we run algorithm on several groups of attributes, which describe the person's life style, statements, activities, interests, recognition of trade marks, statements. Our priority was comprehensibility of the trees. Since some newspapers have a very few readers, we had to adopt the thresholds and parameters, to get the right level of the comprehensibility.

We presented the most promising trees in a form of rules. We took those rules, which have the accuracy above the certain threshold (0.65). On this set of rules we based our interpretations and each interpretation is based on several different rules.
Examples 8-10 will present several interpretations and example of one the rules, this particular interpretation is based on. On the right side of the rule is class (True, False) and the first number stands for the number of cases and the second number for the accuracy.

EXAMPLE 8: Description of readers of daily newspaper Delo

Typical reader of Delo reads newspapers several times per week, has higher level of education recognizes certain trademarks of newspapers/magazines, cars, bears, washing powders, he or her is tracking information about the manufactures, shopping, information about inland news, likes to watch TV and videocassettes.
The results fit well on our intuitive presumption.

Several rules, which are the basis for the description above:
Interpretation: Read newspapers several times per week.
Rule:
if (reading daily newsp. at all)>5) and (Tracking topical subjects at home)>3) and (Recognition of trade marks of daily newspapers >2) and (Recognition of trade marks of daily newspapers <=4) ➔ *T (1051, 0.84)*

Interpretation: Recognize trademarks of newspapers/magazines, cars, bears, washing powders,...
Rule:
if (Recognition of trade marks of daily newspapers >2) and (Recognition of trade marks of daily newspapers <=4) ➔ *T (2333, 0.672)*

Interpretation: Higher levels of education.
Rule:
if (reading daily newsp. at all >1) and (spend evenings at home>1) and (interest at animal world>1) and

(recognition of trade marks of washing powder<=40) and (recognition of trade marks of magazines<=270) and ((Education)>5) ➔ *T (242.9, 0.855)*

Interpretation: Watching TV and videocassettes.
Rule:
if (watching. videocasset.<=6) and (theater<= 4) and (Inf. about manufactures, shopping>4) and (recognition of trade marks of daily newspapers>0) and (recognition of trade marks of daily newspapers<=4) and ((Time of getting up<=11) ➔ *T (108.8, 0.845)*

Interpretation: Interested in inland news.
Rule:
if (newspapers weekly_read)>5) and (Topical subjects at home >3) ➔ *T (1051, 0.84)*

EXAMPLE 9: Description of readers of daily newspaper Slovenske Novice

Typical reader of Slovenske Novice is a regular reader of newspapers/magazines and likes to sit in coffeehouses, bars and sweetshops.: Recognize trademarks of newspapers/magazines, commercials for newspapers /magazines. They recognize less trademarks as readers of Delo newspaper and they also typically recognize different trademarks than readers of Delo newspaper, also reads Slovenski Delničar (magazine that covers economical topics), Jana (magazine tracking topics, which are more feminine), Kaj, Vroči Kaj (yellow press, erotic contents). If he or she is speaking Croatian then also probably reads Kaj magazine.

The most interesting statement is the one saying that readers of Slovenske Novice like to sit in coffeehouses, bars, and sweetshops. This rule looks very strange, but just at first sight. Slovenske Novice is namely the newspaper that has the highest number of readers in Slovenia, but Delo newspaper has the largest edition. Slovenske Novice has the second largest edition. How is this possible? If look closer at the bars, coffeehouses or sweetshops, we could find in the most cases Slovenske Novice newspaper on the table. So, when you are in this in kind of places, besides drinking coffee, or eating sweets, you also read Slovenske Novice.
The statements concerning trademarks could be profitably used for marketing planning.

Interpretation: Regular readers of newspapers/magazines and like to remain sitting for a while in coffeehouses, bars, sweetshops,...
Rule:
if (reading daily newsp. weekly)>4) and (Visiting coffeshops more then weekly<=6) and (interest at animal world>1) and (reading SportNovosti ==F) ➔ *T (331, 0.889)*

Interpretation: Recognize trademarks of newspapers/magazines, commercials for newspapers/magazines,...
Rule:
if (Remembering Commercials for daily newspaper >0) and (recognition of trade marks of Daily newspaper>12) and (recognition of Trade marks of Daily newspaper<=16) ➔ *T (624, 0.887)*

EXAMPLE 10: Description of readers of daily newspaper Ekipa

Typical readers of Ekipa also read Sportske novosti (newspaper which cover exactly the same topics as Ekipa, except that it is in Croatian language), they visit sport events, are interested in motoring, they like to experiment with novelties. Those characteristics are intuitive if we know, that Ekipa newspaper is dedicated to the sports, especially to team sports. Unexpected were characteristics, that they are very tidy and that they have mostly one child between 7 and 14 years old.

Interpretation: Tidy.
Like to experiment with novelties.
Read Sportske novosti.
Have at most one child between 7 and 14 years old.
Rule:
if (importance of tidyness, cleanness >3) and (trying new things >2) and (new challenges <=4) and (liking new things <=3) and (children 14 years old<=1) and (SportNovosti==T) ➔ *T (71, 0.712)*

All the examples until now are dealing with question of describing typical readers of a certain newspaper. How did we attach the other question, which try to distinguish between the readers of the two largest newspapers in Slovenia? We select only those readers who read either Delo or Slovenske Novice. The class for the learning became which newspaper they read. The decision tree divided the readers into the two distinctive groups. Tree is optimized for comprehensibility.

EXAMPLE 11: How do readers from Delo differ from readers of Slovenske Novice? See Figure 1

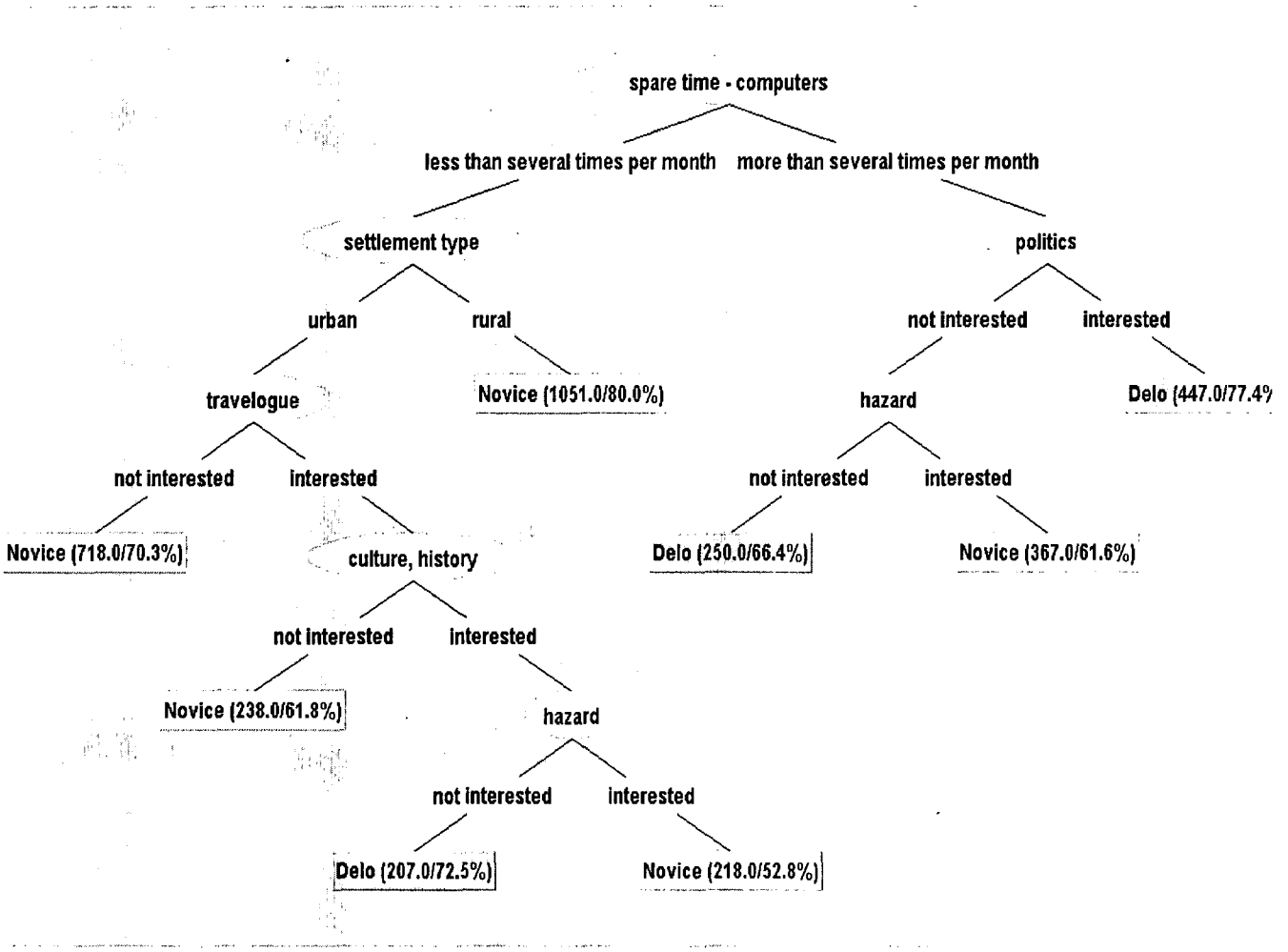


Figure 1: Decision tree for readers of Slovenske Novice and Delo

4.1.4 Clustering

One of our interests was also the identification of different groups of people and to describe their characteristics, regardless on their interest for the media.

First, to determine how many different distinctive groups could be find, we run Kohonen Network algorithm (as implemented in [2]). The result of Kohonen Networks was used as a K parameter in the K-means algorithm[7]. We run the K-means algorithm on 4 groups of attributes (life viewpoints, media topics, spare time activities, demographic properties like age, education, sex). As the result we got four clusters., consisting of 2550, 2124, 1385 and 1894 examples. For the description of cluster's characteristics we use C.5 algorithm, with the additional constrained of 200 examples per leaf. So, each cluster represents separate learning problem, where the target cluster (the cluster we try to describe) represents a positive class and the other three rep[resents a negative class value.

We got rather comprehensive trees with the following interpretation:

- 1st group consists of people younger than 30 years, they are not interested in topics like family, breeding, partnership;
  or
  younger than 20 years, they are interested in topics like living exciting, interested in novelties, films, challenges,...
  We could describe them in short as inspiring young people.
- 2nd group consists of passive people, they don't like challenges, are not interested in entertainment, science, techniques, economics, their main satisfaction is family, they like life without major changes.
  They could be described as inactive older people.
- 3th group consists of people with higher level of education, occupied by computers; mostly they are older than 30 years and listen to music; they

are interested in most of the topics, they have classic taste, they are occupied by their children, promotions, novelties, challenges are important to them, they follow media quite often.
We could describe them as ambitious people.

- 4th group form older people, occupied by handicraft, they are not interested in sport, but they are interested in most of the other topics, they also like to get know with novelties, accepting as challenge.
They can be referred as active older people.

Although most of the results impressed Mediana, the clustering results were the most exciting for them, since they performed the same experiment on the same data with the classical statistical analysis [3]. It took then several steps to define the criteria for the groups they were pleased with. It took their time, resources and expert knowledge. Our results, which were practical the same, were gained in a very short time and without any prior expert knowledge.

5 Conclusions

In the paper we presented an experience with the data mining analysis of the real world data describing the complete media space in Slovenia. Since the agency (private institute Mediana) providing us with the data didn't specify the goals and tasks we should follow, we decided for a set of tasks seemed to us as interesting. The dataset is a collection of approx. 8000 questionnaires each having approx. 1100 questions covering all kinds of topics about the personal interests and relationships of a questioned person to all important Slovenian media (newspapers, radio and TV) as well as it's personal interest, lifestyle, social status etc. For the analysis phase we decided to use several techniques with the some main goal to enable deeper understanding about the dataset.

Since the number of attributes was fairly large (above 1100) we decided first to find highly correlated groups of attributes to give some insight into the structure of the questionnaire. Next, we created with the algorithm Apriori the association rules discovering the relationships in reading habits for the people reading more than one newspaper. Using decision tree learning we enlightened personal characteristics of the people reading certain newspapers and finally with clustering (K-Means) we split the people answering the questionnaires in to several groups according to the attributes describing their personalities and lifestyle.

Most of the results were very useful for the Mediana Institute to get additional insights into their own data. Some of the results have also potential to become additional commercial services offered by Mediana.

Acknowledgement

This work was supported by the EU project Sol-Eu-Net, IST-1999-11495, and the Slovenian Ministry of Science, Education, and Sport.

References

[1] Ian H. Witten, Eibe Frank (1999), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, and the implementation of Weka system, Morgan Kaufmann

[2] Clementine system (http://www.spss.com/clementine/)

[3] Jasna Zdovc, (2000) Segmentation of audience by style (in Slovene), MSc Thesis University of Ljubljana, Slovenia

[4] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, Fast discovery of association rules. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.) Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, pp. 307-328, 1996.

[5] Ross Quinlan (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc.

[6] T. Kohonen (1984) Self-Organization and Associative Memory, Springer-Verlag

[7] Selim, S.Z. and Ismail, M.A. (1984). K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 81-87.

[8] Arabie,P.,and Hubert, L., (1995) Advances in cluster analysis relevant to marketing research. In From Data to Knowledge, W. Gaul and D. Pfeifer, Eds. Springer, Berlin, pp. 3--19.

[9] Leo Bogart (1989), Press and Public: who reads what, when, where, and why in American newspapers, Lawrence Erlbaum Associates, Publishers.

[10] Women and Elections '99" and "Elections in Croatia 2000 - 20 % is (not) Enough" , brochure of Women's Information and Documentation Center, Zagreb, Croatia (http://www.zinfo.hr/engleski/research.htm)

Data mining of baskets collected at different locations over one year

Dunja Mladenić*+, William F. Eddy+, Scott Ziolko+
*J.Stefan Institute, Ljubljana, Slovenia
+Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: Dunja.Mladenic@ijs.si, http://www-ai.ijs.si/DunjaMladenic/

Keywords: Data Mining, Meta Mining, market basket analysis, association rules, decision trees

Received: June 30, 2001

This paper describes the first steps in analysis of millions of baskets collected over the past year from a retail grocery chain containing hundreds of stores. Each record in the data set represents an individual item processed by an individual checkout laser scanner at a particular store at a particular time on a particular day. In order to get some insights in the data, we used several different approaches including some statistical analysis, some machine learning, and some data mining methods. The sheer size of the data set has forced us to go beyond usual data mining methods and utilize Meta-Mining*: the post processing of the results of basic analysis methods.*

1 Introduction

We have obtained a data set form a retail grocery chain which contains all checkout scanner records for approximately one year. The data are delivered to the corporate headquarters on a weekly basis and are processed into a large corporate data warehouse at that time. We obtained a data feed from the in-house data processing activity. The corporate programs are written in COBOL and run on a large IBM mainframe computer. Thus, the records we obtain are in EBCDIC (rather than ASCII) and contain fields which are packed decimal and other non-standard formats. The data arrive weekly on a IBM 3590 cartridge tape which we insert into the Magstar tape library. The files are copied from tape onto the RAID storage and compressed (from 6GB to less than 2GB file size) so that they can be read on the Linux systems (which have a 2GB file size limit). We run our custom conversion program to convert the data from EBCDIC to ASCII and packed decimal to ASCII, etc. This produces one file for each hour of the week. These files are then sorted (by store, time, transaction number, etc.) to put all items in a market basket together.

At this point the data are organized sufficiently for many different subsequent processing steps. Our standard processing generates a new file with one record per basket, listing the items in the basket on that record. We have other specialized projects which perform different processing on the basic sorted files.

2 Data Description

Our data set consists of about a year of data collected over several hundreds of supermarket stores having different sizes and locations. Each record in the data set represents an item that was scanned at one of the checkout stations at a given store, day, and time. For each record we have a number of fields giving details about the conditions under which it was sold, such as price, information about potential price reductions applied (coupon sale, regular sale, . . .), department inside the store, checkout scanner number, and customer number. There are a few million baskets each week and a total of several million customers that are registered as "loyal customers".

Each item is associated with a Universal Product Code (UPC) and there is additional information about the items themselves given in a 4-level hierarchical taxonomy. The top level includes nearly 100 categories of items, such as *bakery, dairy, frozen food, salads, seafood, wine,* etc. The next level, giving a finer grouping of items, includes several hundred groups, such as *bakery mixed, bakery needs, candy, deli, fresh bread & cake, juice drinks, lettuce, milk fresh, pet food,* etc. The third level includes a couple of thousand subgroups such as *fresh fish, frozen fish, other seafood, cake decor and food color, fruit snacks, carrots, peppers, tomatoes, other vegetables, pasta sauce,* etc. The leaf level contains a couple of hundred thousand items with their UPC codes, such as *cherry angel food cake* (within *cupcakes* within *cakes* within *bakery*), *Hanover whole baby carrots* (within *carrots* within *frozen vegetables* within *frozen*), *48% vegetable oil spread* (within *margarine-bowls* within *margarine and butters* within *dairy*), *"wht zinfandel"* (within *wine-misc* within *wine* within *alcohol beverage*).

Because there are a couple of hundred thousand different items it is useful to group them somehow. We found it extremely difficult to create groups by clustering the names and other methods because the text descriptions do not provide common unique identification and it is sometimes difficult to group common products together. For example, there are 1909 entries in our database which contain the text string "MILK," including "MILKY WAY," BUTTERMILK PANCAKES," etc. Of these, 291 contain the string

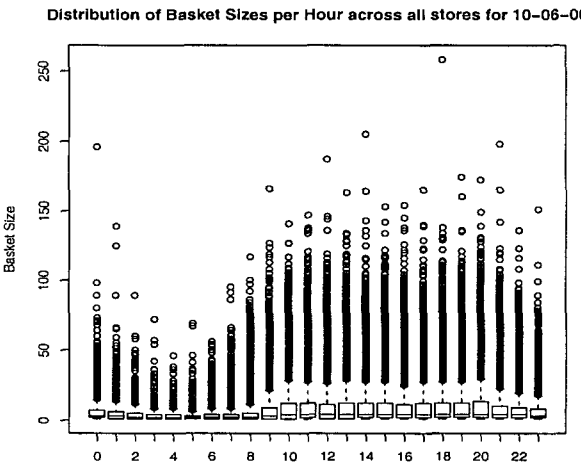


Figure 1: Side-by-side boxplots showing the distributions of basket sizes for each hour of one day across all stores. As expected the most items are purchased during the daytime (Hour 10 to 20 meaning 10 a.m. to 8 p.m.). Notice also that a considerable number of baskets have around 100 items.

"FRESH MILK." Of those, 49 contain the string "2%." Five of those contain the string "1/2%." Thus there are 44 items which correspond to 2% FRESH MILK coming from different suppliers, in different size containers, made of different materials. Because of these difficulties, in our work here we only use the third level of the taxonomy; all of our subsequent analyses are based on the couple of thousand subgroups.

The main unit we are dealing with is a basket, corresponding to the content of a physical basket the customer presented at the counter. The number of baskets varies over hours and stores and so does the number of items in an individual basket. It is interesting to see how the average basket size varies during the day. Figure 1 shows the distribution of the basket size over different hours of one day for all the stores. As expected the most items are purchased during the daytime (10 a.m. to 8 p.m.), where 25% of the baskets contain more than 10 items with a considerable number of baskets having around 100 items. There are also some outliers with over 150 items, even one basket with about 200 items that was purchased around midnight (Hour 0 in the graph). All these outliers potentially reflect noise or error in the processed data and some simple statistical processing can help in identifying such situations.

3 Decision Trees

Decision trees have often been used in Data Mining tasks such as finding cross-selling opportunities, performing promotion analysis, analyzing credit risk or bankruptcy, and detecting fraud. We use the C5.0 decision tree algorithm (an extension of C4.5 proposed by Quinlan[8]) deriving a rule set from a decision tree. We tried to predict the size of a basket based on several characteristics of the transaction, namely: the basket contents, the particular store number (this is an arbitrary corporate designation for the store), and time. We constructed rules predicting basket size within broad categories (Very Small = 1-3, Small = 4-10, Medium = 11-20, Large = 21-40, Very Large = 41 or more). For clarity we have rewritten some of the rules generated by the program. Figure 2 shows some of the rules derived (with pruning set to severity 75 and at least 10 examples per node) from the tree constructed for the hour with the smallest number of the baskets (4 A.M. to 5 A.M.) on an April Monday. The frequency counts of the five basket size categories are, respectively, (381, 84, 15, 3, 1). Since this hour is in the middle of the night most baskets are very small and the rules reflect this fact. The most significant rule notes that non-loyal customers have very small basket sizes. The remaining rules show that the three subgroups, ICE CREAM & DESSERTS, WHITE SLICED,ENTREE-SIDE DISHES, are important for distinguishing between small and very small baskets in the middle of the night.

Rules for Very Small Baskets (all rules):

```
if not loyal customer
then Very Small (275, 0.921)

if not (ICE CREAM & DESSERTS, WHITE SLICED,
ENTREE-SIDE DISHES)
then Very Small (445, 0.83)
```

Rules for Small Baskets (all rules):

```
if loyal customer
and WHITE SLICED
then Small (11, 0.692)

if loyal customer
and ICE CREAM & DESSERTS
then Small (11, 0.538)

if not ICE CREAM & DESSERTS
and ENTREE-SIDE DISHES
then Small (10, 0.5)
```

Figure 2: Some of the decision tree rules for the hour from 4 A.M. to 5 A.M. The two numbers in brackets show the number of baskets covered by the rule and the fraction of baskets for which the rule holds.

The rules for the hour with the highest number of the baskets for the same day (5 P.M. to 6 P.M.) are given in Figure 3. The frequency counts of the five basket size categories are, respectively, (20179, 12855, 6481, 3928, 1195). The number of rules for the same five categories are (5, 32, 93, 114, 33). The most obvious feature of these is that they contain a considerably larger number of clauses. One rule, which we have omitted from the figure, predicts very small basket size for loyal customers if the basket contains EGGS and does not contain any of 39 other specific subgroups.

We looked in more detail at the very small baskets and found 11,931 baskets with one item, 4,691 with two items, and 3,557 with three. The most frequent items in the one-item baskets (in descending frequency) were SNACK

BAR, HOT PREPARED FOOD, EGGS, REGULAR COLAS, BREADS, 2% MILK, BABY FORMULA-LIQUID, etc. The most frequent combination pair of subgroups in the two-item baskets were HOT PREPARED FOODS (twice), FILM COUPON and FRONT RACK ITEM, and MILK (twice) (Film is usually displayed in the front racks). The most frequent triple of subgroups was BABY FORMULA-LIQUID, BABY FOOD-CEREALS, BABY FOOD-JUICES which only occurred 12 times.

Rules for Medium (a subset of 93 rules):

```
if  LowerBound < StoreNo <= UpperBound
and loyal customer
and LUNCHMEAT
and not (MEATS DELI, PASTA, EGGS, POTATO CHIPS,
 BABY FOOD-CEREALS-COOKIES, POTATOES & ONIONS)
and CAT FOOD-WET > 3
then Medium (21, 0.826)

if  time <= 17:20:00
and  loyal customer
and  (SHREDDED CHEESE, POTATOES & ONIONS)
and  not (MEATS DELI, PASTA, YOGURT, EGGS, BACON,
 UNK, BABY FOOD-CEREALS-COOKIES, CONVENIENCE FOODS)
then Medium (21, 0.826)
```

Rules for Large (a subset of 114 rules):

```
if (REGULAR COLAS, EGGS, 2%, SOUR CREAM)
and not ( CONDENSED CANNED SOUPS, BACON)
then Large (31, 0.667)

if loyal customer
and (MEATS DELI, EGGS, POTATOES & ONIONS, BACON)
and not (CANNED CHUNK LITE TUNA, PASTA,
 POLY BAG POTATOES, PAPER TOWELS)
then Large (61, 0.635)
```

Rules for Very Large (a subset of 32 rules):

```
if (YOGURT, EGGS, POTATOES & ONIONS, UNK > 3)
then VeryLarge (48, 0.9)

if (REGULAR COLAS, EGGS, BACON, LUNCHMEAT)
then VeryLarge (77, 0.722)
```

Figure 3: Some of the decision tree rules for the hour from 5 P.M. to 6 P.M.. Notice that UNK is used for for Unknown. It is interesting to note that some of the rules use the (arbitrary) store number to predict basket size and one rule uses the time of day. Several of the rules apply only to loyal customers.

4 Association Rules

In order to find associations between the items in the data, we used association rules. As usual in the market basket analysis, each example in our experiments corresponds to a single basket of items that the customer has purchased. Each example is thus represented as a Boolean vector giving information about presence of items in the basket. Using the data we generated association rules by applying the Apriori algorithm [2] using the publicly available implementation [4], a version of which is incorporated in the commercially available data mining package "Clementine-SPSS".

In a typical data mining setting, it is assumed that there is a finite set of literals (usually referred to as items) and each example is some subset of all the literals. The Apriori algorithm performs efficient exhaustive search by using dynamic programming and pruning the search space based on the parameters given by the user for minimum support and confidence of rules. This algorithm has been widely used in data mining for mining association rules over "basket data", where literals are all the items in a supermarket and examples are transactions (specific items bought by customers).

An association rule is an implication of the form $X \rightarrow Y$, where X and Y are subsets of literals and $X \cap Y = \phi$. We say that the rule holds with *confidence* c if $c\%$ of examples that contain X also contain Y. The rule is said to have *support* s in the data if $s\%$ of examples contain $X \cup Y$. In other words, we can say that for the rule $X \rightarrow Y$, its support estimates the joint probability of the rule items $P(X, Y)$ and its confidence estimates the conditional probability of the rule's implication $P(Y|X)$.

We reduced the number of rules by imposing a ranking on the rules and keeping only the highly ranked rules. For each rule we calculated its *unexpectedness* by comparing the support of the rule with the estimate of support based on the item independence assumption. Specifically, we compared the squared difference between support and estimated support with estimated support. Large values of this statistic make large contributions to the chi-squared test proposed in [3]; see also, [5]. However, we didn't use the chi-squared test for cutting off the top rules since we had no evidence that our data would follow the chi-squared distribution. Instead, we kept the top 1000 rules out of between 40 000 and 400 000 rules, depending on the store and week.

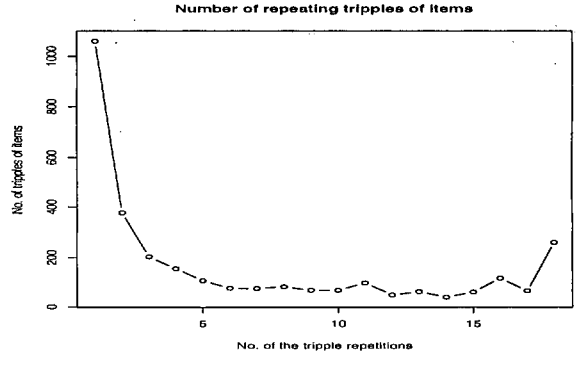


Figure 4: Number of repeating triples of items over the number of repetitions of that triplet.

We have generated association rules on 18 weeks of data collected in year 2000 from mid April through mid October. There are a few weeks that we are missing due to

some technical difficulties in the first phase of data processing needed for obtaining the data from the original format on the tape to our computers. Some of the rules repeat in different number of weeks, Figure 4 shows the number of different triples of items over the number of the triple repetitions. For each week the top 3000 highly ranked rules were selected, meaning that we have 54 000 rules selected in 18 weeks. Since many of the rules actually repeat in different weeks, there is 5856 different rules in the union of all the highly ranked rules of all 18 weeks. In these rules, 1952 different triples of items occur and 1059 of them occur exactly in one week, 377 repeat in exactly two weeks, etc (see Figure 4). Notice that the number of rules is exactly three times the number of triples, since each triple occurs in three distinct rules each having one of the triple item on the left side of the rule and the remaining two items on the right side of the rule. The reason for that is that our minimum confidence used by the rule generation agorithm was set to a very low value (0.1%). There are 780 rules with 260 different triples of items that repeat in the 3000 highly ranked rules of all the weeks. For illustration, we show some of that rules in Table 1. Some of the rules are formed from a frequent pair, such as SPAGHETTI SAUCE and PASTA or SHAMPOOS and CONDITIONER-RINSES, that is combined with different frequent single items, such as BANANAS or TOMATOES. The same rule has a different support in different weeks.

5 Discussion

Most data mining approaches concentrate on extracting interesting properties directly from the collected data. There are different proposals on how to post-process the results in order to focus only on interesting properties. An other way of post-processing is mining the results of other data mining algorithms, applying *Meta-Mining*. For instance, finding spatio-temporal information by mining the induced rules [1]. Inferring higher order rules from association rules as proposed in [7] involves applying time series analysis on support and confidence of an association rules, that is repeating over time. This assumes that we have enough data collected over time and that the same association rules repeat over time. In our data, we found 260 triples repeating over all the weeks.

The following four rules repeated in 86% of the stores in 9 consecutive weeks (15-May through 10-July):

- (RTE KIDS or RTE WHOLESOME FAMILY) and SPAGHETTI SAUCE and PASTA
- SPAGHETTI SAUCE and PASTA and TOILET TISSUE
- SPAGHETTI SAUCE and PASTA and TOMATOES
- MEXICAN FOODS and SHREDDED CHEESE and TOMATOES

If we change our restrictions slightly, such that a rule needs to appear in at least 86% of the stores in 8 of the 9 weeks (15-May through 10-July) we get 48 rules such as:

- MEATS DELI and LETTUCES and TOMATOES
- TOILET TISSUE and PAPER TOWELS and FACIAL TISSUES
- TOILET TISSUE and PAPER TOWELS and SOAPS - HAND & BATH
- VEGETABLES and CARROTS and PEPPERS

The most complete coverage are the two rules which are in 96.5% of the stores in all 9 weeks. These rules are:

- GROUND BEEF and SPAGHETTI SAUCE and PASTA
- VEGETABLES and LETTUCES and TOMATOES

By applying a simple intersection on the sets of the most "interesting" rules found for different weeks and different stores, we additionally reduced the number of rules a person may want to check when searching for "stable rules" repeating either over time or/and over different locations (stores). More sophisticated methods are needed to obtain additional information from the generated rules, such as clusters of similar stores or time series analysis of the selected rule support or confidence. These are part of our on-going work investigating items in our every-week growing database of transactions.

Acknowledgement

The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport.

References

[1] Abraham, T. and Roddick, J. F. (1998), Opportunities for knowledge discovery in spatio-temporal information systems, *Australian Journal of Information Systems* 5(2), pp. 3-12.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.) *Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press*, pp. 307–328, 1996.

[3] Brin, S., Motwani, R., and Silverstein, C. (1997), Beyond Market Baskets: Generalizing Association Rules to Correlations, In *Procedeengs of the ACM Conference on Management of Data (SIGMOD-97)*, pp. 265-276, Tucson, Arizona, USA.

[4] C. Borgelt. Apriori. http://fuzzy.cs.Uni-Magdeburg.de/~borgelt/.

[5] DuMouchel, W. and Pregibon, D. (2001), Empirical Bayes Screening for Multi-Item Associations, In *Proc. KDD, 2001*, ACM.

[6] Grobelnik, M., Mladenic, D. (1998), Learning Machine: design and impelmentation, *Technical Report IJS-DP-7824*, Department of Intelligent Systems, J.Stefan Institute, Slovenia, January, 1998.

[7] Spiliopoulou, M. and Roddick, J. F. (2000), Higher Order Mining: Modelling and Mining the Results of Knowledge Discovery, *Proc. 2nd Int. Conference on Data Mining Methods and Databases*, WIT Press. (Ebecken, N. and Brebbia, C. A., Eds.), pp. 309-320.

- BAKING CAKE-BROWNIE-COOKIE <- BAKING READY-2-SPREAD FROSTING and EGGS (0.11%, 71%)
  - EGGS <- BAKING READY-2-SPREAD FROSTING and BAKING CAKE-BROWNIE-COOKIE (0.11%, 32%)
  - BAKING READY-2-SPREAD FROSTING <- BAKING CAKE-BROWNIE-COOKIE and EGGS (0.11%, 24%)
- SPAGHETTI SAUCE <- PASTA and FROZEN POULTRY (0.27%, 55%)
  - PASTA <- SPAGHETTI SAUCE and FROZEN POULTRY (0.27%, 52%)
  - FROZEN POULTRY <- SPAGHETTI and SAUCE PASTA (0.27%, 11%)
- PASTA <- SPAGHETTI SAUCE and TOMATOES (0.67%, 47%)
  - SPAGHETTI SAUCE <- PASTA and TOMATOES (0.67%, 44%)
  - TOMATOES <- SPAGHETTI SAUCE and PASTA (0.67%, 28%)
- PAPER TOWELS <- TOILET TISSUE and PAPER NAPKINS (0.22%, 50%)
  - TOILET TISSUE <- PAPER TOWELS and PAPER NAPKINS (0.22%, 45%)
  - PAPER NAPKINS <- TOILET TISSUE and PAPER TOWELS (0.22%, 14%)
- SHAMPOOS <- CONDITIONER-RINSES and BANANAS (0.14%, 61%)
  - BANANAS <- SHAMPOOS and CONDITIONER-RINSES (0.14%, 28%)
  - CONDITIONER-RINSES <- SHAMPOOS and BANANAS (0.14%, 26%)
- VEGETABLES <- LETTUCES and CARROTS (0.81%, 68%)
  - LETTUCES<- VEGETABLES and CARROTS (0.81%, 38%)
  - CARROTS <- LETTUCES and VEGETABLES (0.81%, 26%)

Table 1: Example rules that repeat in all 18 weeks, if we generate association rules ignoring the store information. We show all three rules for the selected triples of items. For each rule we give its support and confidence from the mid May week. Support is showing the fraction of baskets containing all the rule items, thus it is the same for all three rules containing the same tripped of items. Confidence is showing the proportion of baskets containing the pair of items in the right side of the rule that also contain the item in the left side of the rule.

[8] J. Ross Quinlan (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc.

Multimedia supported study of achieving high worker's efficiency in relation to his work

Zvone Balantič and Mojca Bernik
University of Maribor, Faculty of Organizational Sciences
Kidričeva 55a, SI-4000 Kranj, Slovenia
zvone.balantic@fov.uni-mb.si, mojca.bernik@fov.uni-mb.si,

Keywords: electronic publication, multimedia, education, ergonomics

Received: November 15, 2001

Electronics, engineering and medicine expert collaboration as rule requires knowledge of both the remaining fields from each of experts to result in a useful application. Ergonomics and work process management are the goals achieved by such approach. Comprehensive analysis of work process, influenced by environmental factors and output, measured as a function of these influences, is a cornerstone of these multimedia CD compilation, presented both in a textbook and handbook format. In the electronic publication study contents are combined with practical experience of efficient Slovenian companies, which were prepare to present a piece of their experience.

1 Introduction

The interactive electronic publication MAN-WORK-EFFICIENCY is prepared in a form of CD containing pedagogical and related scientific and research elements of analysis and synthesis of working systems. By a contemporary access – that is enabled by today's information technology – we tried to additionally motivate an independent and individual study, opened in the way of discussion and research. On the whole, this CD is divided in two parts (theoretical and practical one), which are dynamically linked and interwoven. Certain corresponding points in the contents provide the link between both parts, where theoretical part is linked to the practical one and inversely.

Today, all the disciplines are increasingly interconnected, and, in this sense they are exploring new and interesting ways that usually lead towards the entirely new dimensions of thinking.

Interdisciplinary themes can't withstand not to imitate and apply the complicated system into a related one, which is mathematically and algorithmically easier to be managed. In this process we are essentially helped by an up-to-date information technology that stimulates procedures which would otherwise demand much more painful work. Among single disciplines there is an intertwined net of information and because mastering of single specialty is subject to the inner field of profession, there was, for example, in the field of interaction between a man and a machine, established a science called ergonomics. This field is very important for in any human activity we are aiming to attain the optimal results by taking into consideration all human principles of human work [1]. The study of this scientific discipline is interdisciplinary one since at each step we are reaching for in the field of ergology, medicine, biomechanics, anthropology, physiology, psychology, sociology, environmental studies, work organization, system theory, technology, techniques and industrial design [2]. We can say in regards to performance qualities, human and computers complement each other, rather than compete. Our expectations of what information technology can do, should therefore be based on human and machine complementarily, and directed towards an active support to mental processes [3]. Taking in consideration such an extensive and ramified subject it was clear that some parts of the scientific literature had to be united on an integral basis of knowledge. In this process we could be helped by a multimedia access.

2 Working methods

Electronic publication was produced in the Microsoft Windows 98 environment by means of Microsoft PowerPoint 97 tool. The content can be viewed and interactively communicated with by support of Microsoft PowerPoint Viewer 97 application.

In preparing CD we kept to didactic and pedagogical principles. Electronic publication was first of all conceived as a manual with theoretical and educational contents meant for everyone who is dealing with analysis of working processes, from work organizers, safety engineers, to medical doctors specialized in medicine of work. Besides the above mentioned the CD is also a manual providing instructions for introduction into research projects in the field of designing healthy working environments. It contains a series of practical examples from companies where working processes are already ergonomically designed and serve us as help, respectively, as a model. And indeed, this perspective gives us a superstructure of the acquired knowledge and represents applied solutions from different fields of elaborated topics.

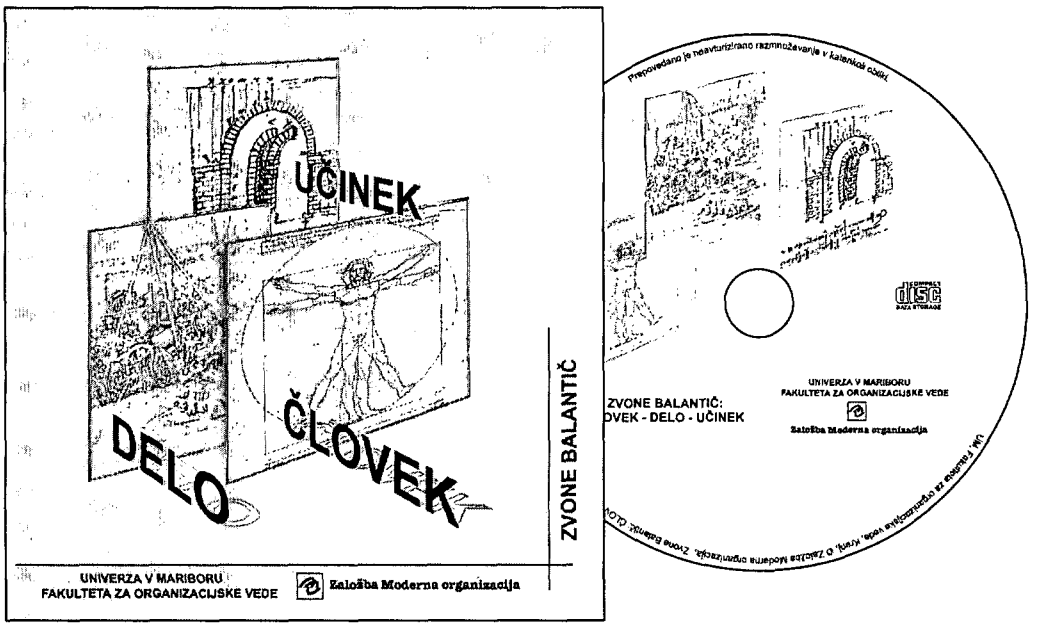


Figure 1: Electronic publication MAN-WORK-EFFICIENCY

The electronic publication MAN-WORK-EFFICIENCY could be used in the pedagogical process (in doing exercises – contents' comprehension; working in laboratory – solutions that are suitable for applications), in designing the working process at a workplace, in research projects (methods of measuring...), in encyclopedic refreshment of single topics (chapters' structure) etc. A "rounded book" is meant for students and professors in the field of organizing working processes, meanwhile in the field of direct application, to the top employees of the company and related production units, safety engineers, system theorists, consultants, design engineers, technologists...

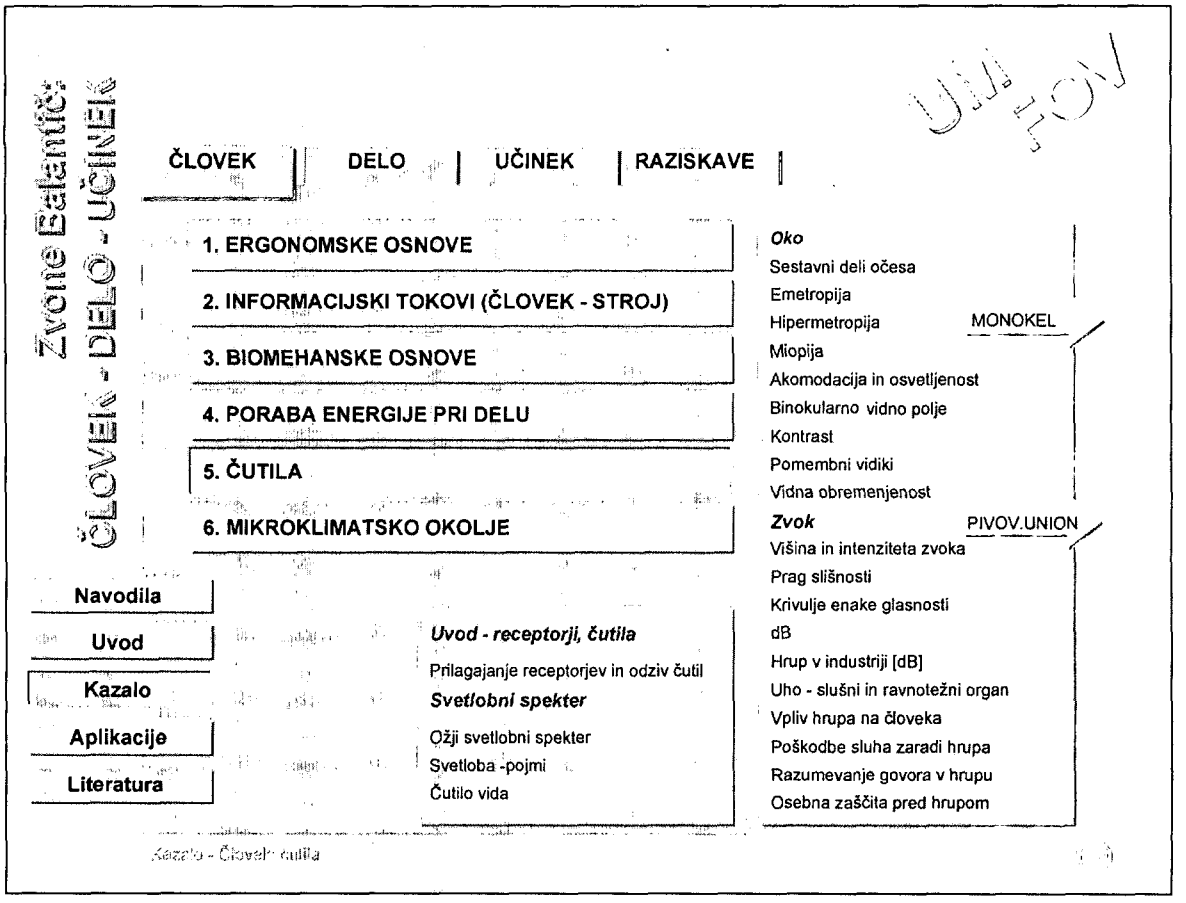


Figure 2: Communication window of the index

By using software package PowerPoint 97 we exploit all available possibilities of the multimedia presentation incorporated this package. The segments are represented in the interactive way and are dynamically interconnected. The represented topics contain scientific & research elements of the analysis and synthesis of the working systems based on theoretical comprehension and practical experiences. The guidance is carried out from introductory database and is in accordance with a logical completion of knowledge.

By inclusion of multimedia elements the presentation tries to be interesting and an up-to-date one. Already at the start we are acquainted with 17 theoretical thematic complexes that comprehend the areas of work's designing, ergonomic basis, information flow between man and machine, biomechanics in relation to a man, energy consumption at work, senses for sound and light, designing of working environment with charges and efforts, computer work, temperature of the environment, scheduling of the working time, production and technical basis of work, phases designing of work, safety at work... The communication window calls our attention also on 20 applications, which are directly linked to the theoretical topics, yet they derive from practical examples generated in Slovenian companies. In this part the following topics are presented: designing of workplace for a disabled worker, problems dealing with attaining the threshold of pain, solving the noise in the working environment, correct sitting in working with a computer and simultaneous communications and feed-backs in regulatory circles... [4].

Finishing the CD we can resume that we went through 710 pages supported by multimedia. We walked through 260 figures, 353 author's drafts and 233 graphs. Theoretical and practical part differs in colors and partly in design. Each individual can regulate the rhythm of study by himself in returning from certain theme and skipping along different generally established ways (stepping forwards/ backwards, skipping at the beginning/ at the end, from one topic to the other, from the theoretical to the practical part, using the hyperlink with the internet...).

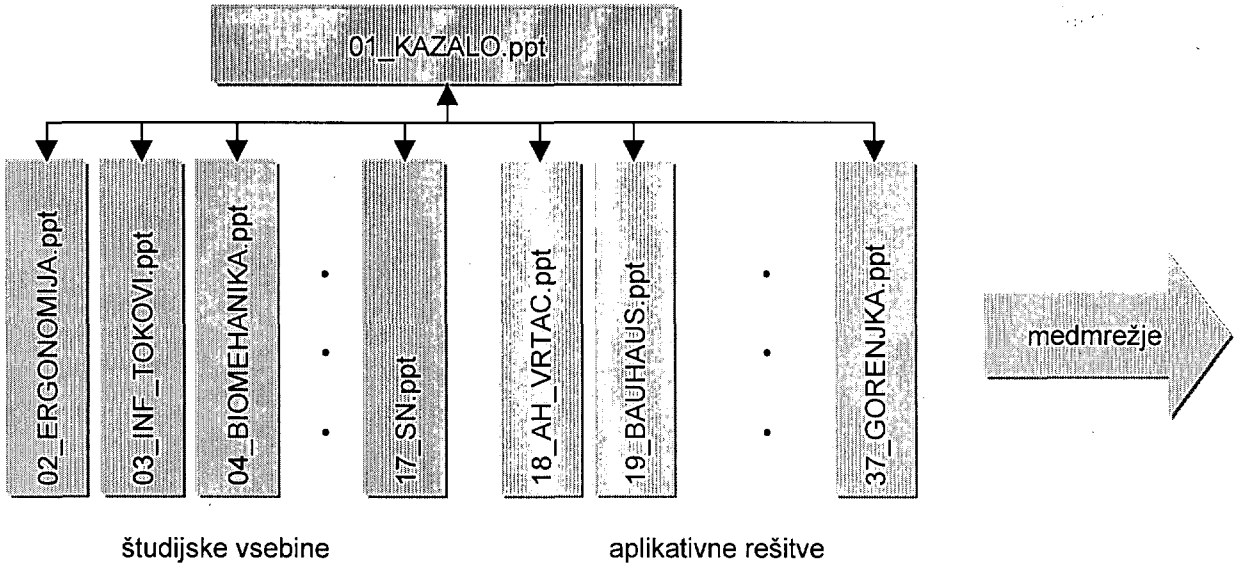


Figure 3: Thematic construction of the electronic publication MAN-WORK-EFFICIENCY

The communication with the electronic publication stimulates the quality of the study (individual study and topic related discussions), encourages the innovative activity in the field of the man in the working process and broadens the awareness about the importance of the presented knowledge. If we think about the study as of a working process, then planning, preparation, execution, control and management are taking place in it. The use of the multimedia CD takes place in the closed regulatory noose, where by means of the comparative article we can establish the deviation from the reference values and, in this way, we are able, in each moment, to correct it.

3 Discussion

In its concept, the electronic publication is devised as a system of hierarchical study. For the reasons of present ability and constant overview of the contents, it is always recommendable to set a basic point of orientation. The index is the most suitable starting point from which we are looking for the new ways and try to orientate ourselves in the system of a multi-layer study material. The system of absorbing oneself into the knowledge follows certain explanation, which means going deeper and deeper, finally to the level of details. Since in the CD MAN-WORK-EFFICIENCY we find theoretical and practical contents, we would like to emphasize, that the links between both branches are not

established at the deepest level – where by using "hyperlink" we would search for the parallels in the details – but are set at the upper level (index), where we are returning after we have been acquainted with the theoretical part. Our choice can be also the opposite one, i.e. to go through the layers to the theoretical details after being acquainted with the practical contents.

Multimedia simulation supports virtual walk through the contents, which are sometimes difficult to explain in a classical way. Studying, we go through information net, composed of graphs, tables, explanations and animations.

A set of information follows in a logical turns and, in this way, forms the so-called table figure, which is similar to a real figure from a lecture. The whole complex enables an autonomous study, which is later linked to discussions at different levels. Thus, a possibility is given to design "a study on a stock", with the students coming to attend the lectures but having already a certain knowledge that serves for a discussion on a certain topic. In this way we stimulate the process of thinking and wake up creativity, which is based on a stocked knowledge, and temporary situation points that are modeling the information mass in a new form. In creating new ideas a decisive role is played by knowledge, experiences, mental ability and capability of constructive thinking. The challenge lies in the theoretical part of the electronic publication, meanwhile in the practical–applicable part the constructional characteristics of the devices, work planning, study of documents and analysis and synthesis of the processes are exposed. By a mental activity, besides a direct action, an information process–as a part of the activity between man and environment–takes part. Within the framework of the considerations about the electronic publication we can see the meaning in encouraging perception, interpretation and processing of the information that we got through our senses. By an absorbed work with the multimedia educational tools we try to remain inside our abilities of information' reception (34 – 42 BIT/s). Namely, in everyday life, the capacity of the central brain system can't manage to process the whole incoming information' flux. The truth is, that the capacity of the human long-term memory is inconceivably large and according to some estimation it lies between 10^8 and 10^{15} BIT. But there is also presented the problem of the information channels translation, which is the highest at the perception (10^9 BIT/s) and the lowest by creating a lasting impression (0,7 BIT/s) [5]. It's obvious that always occurs a reduction of the process, and that's something that all the creators of such electronic publications should keep in mind.

4 Conclusion

The primary orientation towards health should be – besides calling attention to bad habits – encouraged as well by a professional access aiming to eliminate the risk factors that cause diseases at workplace. At this point, the interdisciplinary overgrows all the interest fields and, by team operation of different professions, it raises a hope for a success in the area of the primary prevention either. Even the physicians qualified to participate in organizing a working process (specialists for the medicine of work) can find, in such collaboration, a new motivation for their work. Besides the study contents this electronic publication contains all the elements of the manual for a real help in designing working environments that are less harmful to the human health.

5 References

[1] Pheasant S. (1999): Bodyspace (Anthropometry, Ergonomics and the Design of Work), Taylor & Francis, London.

[2] Balantič Z., Fležar M., Novak P., Šorli J., Kandare F. (1997): Computer Simulation of Lung Volumes and Ventilatory Capacities Using Anthropometric Data Obtained on the Anthropometric Measuring Chair (AMC): Respiratory Care, Vol. 42, No. 11, 1074, New Orleans, Louisiana.

[3] Bernik M. (2000): Development of HRIS in Intranet Environment, Master work, Kranj.

[4] Balantič Z. (2000): Man-Work-Efficiency [CD-book], Moderna organizacija, Kranj.

[5] Kroemer K.H.E., Grandjean E. (2000): Fitting the Task to the Human, Taylor & Francis, London

On the modelling of management decision-making processes in organized anarchy

Cene Bavec
School of Management in Koper, Slovenia
Phone: +386 5 610 2000; fax: +386 5 610 2015
E-mail: cene.bavec@guest.arnes.si

Keywords: organized anarchy, informed anarchy, strategic management. Petri net, computer simulation

Received: April 23, 2001

The paper discuses the role of the Organized Anarchy paradigm and the Garbage Can Model in strategic decision-making, and extends the idea of Organized Anarchy to the Informed Anarchy with unclear technology of allocation and dissemination of information, incomplete understanding of information, and fluid participation of information in decision processes. An example of the computer simulation is briefly presented, but the discussion in this paper is limited to the relation between the level of organization anarchy, load of problems, formal and informal information systems, and efficiency of decision-making. The models suggest that managers could enrich their decision-making by making their organization function like a net in which they catch "ingredients" needed for strategic planning and efficient decision-makings. Such nets are intelligently employed and motivated members of the organization.

1 Introduction

Information technology management was born in the stable and predictable environment of centralized information systems. The situation has dramatically changed with the explosive growth of the Internet and the appearance of the new economy. Information technology has become a technological basis of new products and services, so it has moved from the background into the core of strategic planning of contemporary organizations.

Strategic management of information technologies is becoming a crucial part of business strategy and is starting to share its uncertainty and chaotic environment. It seems to many managers that development of the new economy is so fast and unpredictable that they cannot control or plan the future of their organizations. For them success is more a coincidence than a result of planning. Others argue that nothing has dramatically changed, but many argue that, as a result of this development, strategic management in general is becoming unacceptably chaotic. Strategic management is, of course, so diverse that it could easily extend to both extremes, but we will focus mostly on its chaotic side. In the focus of the following discussion is strategic management of information technologies, but the majority of challenges and problems are the same for the strategic management in general, so will use more general phrase - strategic management.

Disorder and anarchy are more an appearance than the essence of the new economy and emerging information society [Maram, 2000]. Old paradigms or even dogmas of industrial society prevent us from detecting and understanding certain patterns in the new business landscape. That is why many managers and especially researchers are trying to re-evaluate some of the basic presumptions and to deepen their understanding of management and business sciences in the light of the new circumstances. The main question is what contemporary managers can learn from computer sciences, from different theories and models of decision-making in a chaotic environment.

In the last thirty years numerous theories and models of decision-making and system behavior were introduced, which attracted the attention of many high-level managers and researchers. Particularly, Chaos Theory [Gleick, 1987], Complexity theory [ex. Lewin 1999] and Organized Anarchy [Cohen, March, Olsen, 1972] have been in the focus of their interest.

The framework of the Chaos Theory is very familiar to managers – complex systems with chaotic behavior, with irregular and hard to predict patterns. The Chaos Theory can in many ways better explain the behavior of the organization than can the classical methods of scientific management [Phelan, 1995]. Besides its philosophical power, the application of the theory in practice is still very limited, concentrating mostly on descriptive suggestions to managers on how to understand the nature of uncertainty, and how to balance between strategic and operational management.

A somewhat different view on complexity is expressed in the Complexity Theory. From the methodological point of view it is an interdisciplinary approach to studying dynamic processes involving the interaction of many actors. Simple sub-systems can produce very complex and hard to predict systems. The Complexity theorists argue that managers should not impose their solutions on organizations but should rather introduce some basic rules and support the creativity of their employees. That would create a synergy of

individual knowledge and increase an organization's ability to produce and detect unpredictable solutions.

The third important model for decision makers is Organized Anarchy. The following discussion is based on the Organized Anarchy paradigm and the Garbage Can Model of Organizational Choice GCM [Cohen, March, Olsen, 1972]. The idea behind this model of decision-making is widely accepted and still relevant for researchers and practitioners. Its power is in its simplicity and its human acceptance as being common sense.

2 Organized and Informed Anarchy

2.1 The Garbage Can Model (GCM)

The classical theory of rational decision-making is based on the presumption that the decision is based on known options, known consequences, defined criteria, and defined decision-making technology. Studies of decision processes in real organization have shown that the theory of rational decision-making is too often far from reality. Research, and especially practice, shows that an organization can work and survive even in situations where decisions are not optimal, not intentional, and not made on time. Cohen, March, and Olsen (1972) argued that the situation in an organization is basically an organized anarchy that they characterized as a decision environment with *problematic preferences, unclear decision-making technology* and *fluid participation*. They developed a model for decision-making based on the organized anarchy paradigm (Garbage Can Model of Organizational Choice - GCM).

In the GCM we find the mix of *problems*, *choices opportunities*, *solutions*, and *participants*. The model viewed the choice opportunities as a garbage can into which the members of the organization dropped the various unresolved problems and possible solutions. The authors described the organization as a collection of choices looking for problems, problems or issues looking for decision situations in which they might be aired, situations looking for problems to which they might be the answer, and decision-makers looking for decision-making. In everyday life, solutions discovered by "accident" are as equally good as solutions found through the process of rational decision-making. From the managerial point of view, only the results are important. The model also explained why an organization could work in non-predictable or even chaotic circumstances. An organization can survive only if the "Garbage Can" decision-making process produces enough rational or useful decisions.

Later studies [Padgett, 1980; Carley, 1986; Anderson and Fisher, 1986; Masuch and Lapotin, 1989] improved some aspects of the GCM, adding features of organizational hierarchies and elements of formal decision-making, or else formalized some features of the model [Heitsch, Hinck, Martens, 2000].

2.2 Informed Anarchy

Every decision-making depends on availability of the relevant information that could come from formal information systems or other information sources, including personal contacts. Particularly strategic management depends strongly on information sources that cannot be fully formalized in the organization's information systems. Managers are overloaded with a chaotic mixture of relevant and irrelevant information on the hand, side and the lack of some crucial information on the other. From the information point of view, it definitely makes their planning and decision processes more or less organized anarchy. In the further discussion our interest will be focused on the information side of the organized anarchy and the GCM.

We could extend the idea of Organized Anarchy to include some features of formal and informal information systems, which are only indirectly present in the Cohen, March, and Olsen model. We could call Informed Anarchy a decision-making environment with:

- *Unclear technology of allocation and dissemination* of information (members of the organization do not know were to look for information and how to disseminate it on time);
- *Incomplete understanding* of information (even when members acquire information they could still misunderstand or even deliberately misuse it);
- *Fluid participation of information* in decision-making processes (the information that supports decisions are constantly changing).

An Informed Anarchy paradigm is based on the fact that the unclear technology of allocation, dissemination, and also the faulty understanding of information are prevailing features of strategic management in many organizations. Paraphrasing the GCM, in the model of Informed Anarchy:

- information is looking for decision-making situations;
- decision-making situations are looking for information;
- information could generate decision-making situations;
- participants are looking for information and decision-making situations.

The GCM has explained how Organized Anarchy can produce enough rational decisions for an organization to be able to function and survive. Similar findings should be proved for the Informed Anarchy, which should provide the organization with enough relevant information for decision-making. To build a full-scale realistic model we should implement very complicated relations between decision-making situations, which need information, different sources of information, and knowledge of people involved in decision-making. We could also simplify models to study only selected features. Such a model is presented in the paper.

3 An example of the computer simulation

To illustrate that modeling could be very realistic we shall briefly present a selection of results from the computer model based on the GCM, extended with elements of Informed Anarchy and classical hierarchical organization. The model was developed under research on the object oriented modeling of organization [Bavec 1995]. The organization structure is represented with a colored Petri net and fuzzy logic that reflects informal and ambiguous features of organization hierarchies. The colored Petri net's superior semantic power makes possible a very rich interpretation of organization structure and computer implementation of the discrete simulations. The model describes the complex interplay of different levels of organizational rigidity or anarchy, the workload with incoming problems, the time for decision-making, and the ratio of solved problems.

The discussion in this paper is limited to two examples. The first one (Figure 1) is the relation between *level of organizational anarchy* (probability that the individual will act in accordance with his/her position in the organizational hierarchy), *load of problems* (flow of problems detected by an individual) and *efficiency of decision-making* (ratio of solved problems that were detected). It shows the following:

- Very rigid organization enables decisions in a very broad band of loads (effectiveness of decision-making is just slightly effected by the load of problems), on the other hand, the average efficiency is lower;
- An organization with low rigidity or high anarchy is very sensitive to loads and is less effective under high loads;
- Under extremely high loads (curve 6) even slight anarchy makes the effectiveness fall sharply (such organizations could function only as a strong hierarchy);
- The model anticipates that under medium loads (curves 4 and 5) strongly hierarchal organization is less efficient, and it is a good managerial policy to allow a certain degree of anarchy (in the GCM sense). But, if anarchy increases over a certain level, the efficiency starts to decrease. Similar results are reported by Masuch and Munter (1989) using the double AISS model. In their empirical research, Collins and Munter (1990) also concluded, that under high loads the range of informal communications is increased.
- The model also anticipates that under low problem loads the efficiency rises constantly (curves 1 and 2). It is difficult to assess how realistic this assumption is in practice, but it indicates that organizations that face a small flow of problems (though they could of course be very serious for the particular organization) should rely on the self-initiative of their members.

It is a well-known fact that a certain degree of intentional or unintentional anarchy in an organization increases its effectiveness. This is the foundation of all contemporary business organizations. Our model however shows also the third dimension – the load of problems, which could dramatically change some intuitive assumption about efficiency.

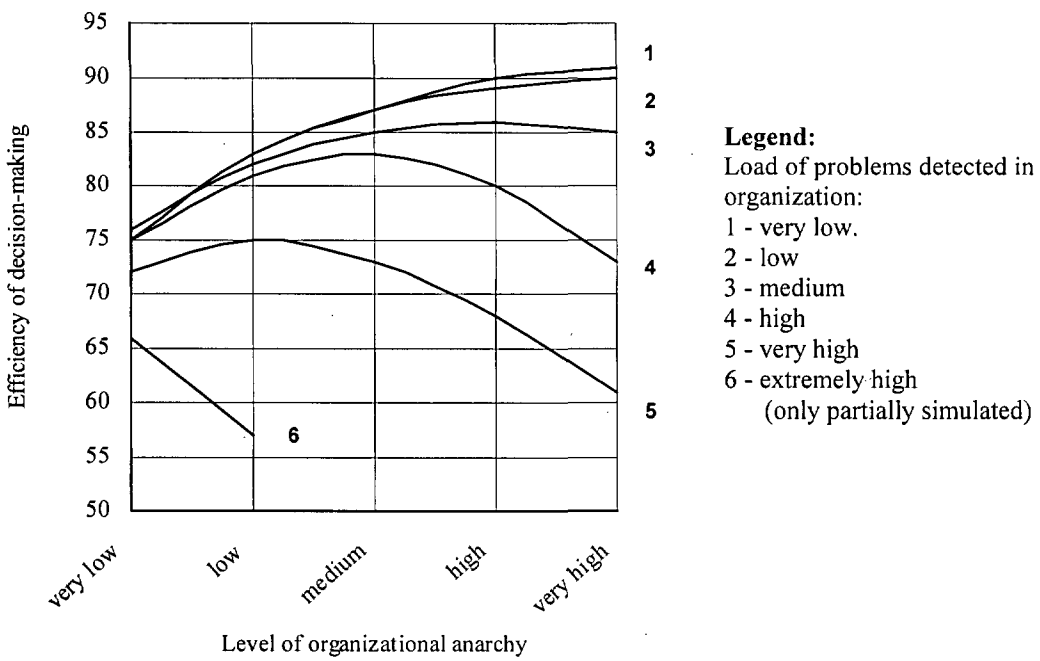


Figure 1: Efficiency of management decision-making of as a function of the level of organizational anarchy and load of problems

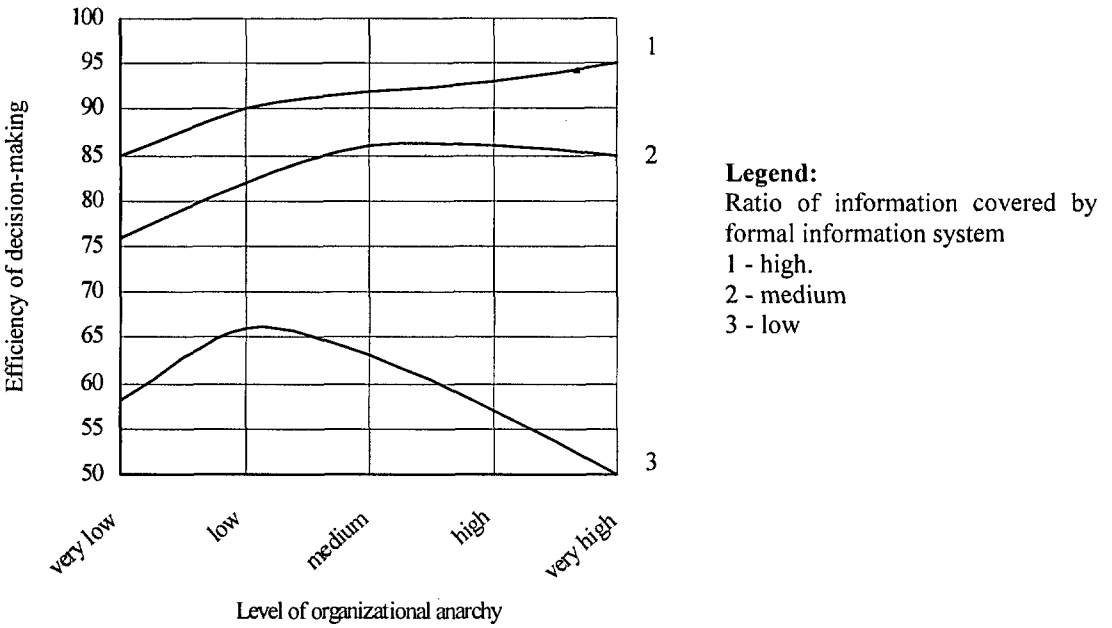


Figure 2: Efficiency of management decision-making of as a function of the level of organizational anarchy and the ratio of information covered by formal information system

Another example (Figure 2) from the same model shows the relation between *level of organizational anarchy*, *ratio of information covered by formal information system* and *efficiency of decision-making*. The new parameter introduced - ratio of information covered by formal information system - presents two different information sources for decision making: formal information systems and informal information sources.

In the model we experimented with statistical probability to get part of information for decision-making through fast and relatively accurate communication channels (from the computerized information systems), and part of information through less accurate and significantly slower channels (from informal information sources like personal contacts). We tried to simulate as realistic situations as possible, including different assumptions for strategic, tactical, and operational decision levels. Results of the simulation show the following:

- The efficiency of decision-making is very sensitive to load of problems in the decision-making environment with high level of informal information resources – at the beginning it increases with the rise of organizational anarchy but, it soon starts to decrees even in moderate organizational anarchy (curve 3). The model predicts that rigid organizations are more efficient in decision-making situations with predominantly non-computerized information systems. From the historic perspective it makes a sense. We are moving into digital economy with total computerization, virtual organizations [Strausak 1998, Mowshowitz 1999, Franke 2001], and ambiguous business environment;
- On the other side, in the situations with higher utilization of the formal information systems the efficiency of decision making is less sensitive to the load of problems (curves 1 and 2). This result could also be, at least intuitively confirmed in real organizations – one of the primary goals of modern information systems is to increase efficiency of decision-making under high pressure of very diverse problems.

The described model also simulate some other parameters in decision-making situations like *time for making decisions*, but the purpose of this brief presentation is just to demonstrate that the computer simulation based on the GCM and the Informed Anarchy, as described in the previous section, can present realistic results that could be confirmed in experience.

4 Discussion

There will be always room for rational decision-making, but contemporary managers should also master other side of the coin – how to manage uncertain and ambiguous situations with a high level of information anarchy. Different theories and models could provide them with an insight into decision-making technology and draw their attention also to new approaches that could be very far from traditional scientific management. This is particularly important for strategic decision-making, which is usually faced with ambiguous and even chaotic situations. .

Understanding the nature of decision-making in an ambiguous environment, also gives managers an opportunity to develop new criteria for measuring their effectiveness and, even more importantly, the effectiveness of their organization. The new management paradigms or even doctrines must incorporate new definitions of risk and responsibilities of managers in decision-making. As an example, even now there is a very noticeable difference between managers' attitude in Europe and America toward risk in decision-making.

It cannot be done in a totally chaotic manner, so we need methodologies that are simple enough and useful to managers in everyday life. Many authors [Drucker 1999; Morabito, Sack, Bhate 1999] point to basic differences in traditional and new management approaches and techniques. Nevertheless, strategic management has not yet developed efficient methodologies and recommendations to cope with the extremely fast changing environment of the new economy. Modeling and computer simulations, as ones based on the Chaos Theory, Complexity theory, and particularly the Organized Anarchy, could be one of the useful tools for researchers and practitioners to describe and study new management paradigms, and also to develop new efficiency and benchmarking criteria.

5 Conclusion

The models and theories described imply that information, problems, solutions, and opportunities are around us, and they come and go. The main strategic task of management is to detect and to utilize them. The models suggest that managers could enrich their decision-making processes and raise the quality of organizational decisions to simulate something like fishing with nets. A fisherman is not aiming at a particular fish but rather at the shoal. Similar, management should design the organization to function like a net in which they would catch all "ingredients" needed for strategic planning and efficient decision-making. The elements of such nets are wisely employed and motivated members of the organization that are sharing, on highly organized manner, their experience and knowledge.

References

[1] Anderson P.A., Fisher G.W. (1986) A Monte Carlo Model of a Garbage Can Decision Process, *Ambiguity and Command,* Ed. J.G. March, R. Weissinger-Baylon, Pitman Publishing Inc.

[2] Bavec C. (1995) Object Oriented Modeling of Organization, *Dissertation* (in Slovene), University of Ljubljana, Faculty of Economics

[3] Cohen M. D., March J. G., Olsen J. P. (1972) A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly*, 17 1-25

[4] Collins, F.; Munter, P. (1990) Exploring the Garbage Can: Study of Information Flows, *OMEGA International Journal of Management Sci*, Vol.18, No. 3, pp. 269-281,

[5] Drucker P.F. (1999) Management Challenges for the 21st Century, *Butterworth-Heineman*

[6] Franke U.J. (2001) The Concept of Virtual Web Organizations and its Implications on Changing Market Conditions, *Virtual Organization Net, Electronic Journal of Organizational Virtualness*, Vol 3, No. 4

[7] Gleick, J. (1987) Chaos: Making a New Science, *Penguin Books*, New York and London

[8] Heitsch S., Hinck D., Martens M. (2000) A New Look into Garbage Cans - Petri Nets and Organizational Choice. *CPM Report* No.: 00-63

[9] Kilduff M., Angelmar R., Mehra A. (2000) Top Management - Team Diversity and Firm Performance: Examining the Role of Cognitions, *Organization Science*, 11: (1) 21-34

[10] Knoke D. (2001) Changing Organization, Business Networks in the New Political Economy, *Westview Press*

[11] Lewin, R. (1999) Complexity: Life on the Edge of Chaos, *University Chicago Press*

[12] Masuch M. (1992). Artificial Intelligence in Organization and Management Theory, *Artificial intelligence in Organization and Management Theory: Models of Distributed Activity* (pp. 21-40). Amsterdam: North-Holland.

[13] Masuch M., LaPotin P. (1989) Beyond Garbage Cans: An AI Model of Organizational Choice. *Administrative Science Quarterly*, No. 34, 38-67.

[14] Morabito J., Sack I., Bhate A. (1999) Organization Modeling – Innovative Architectures for 21st Century, *Prentice Hall*

[15] Mowshowitz, A. (1999) The Switching Principle of Virtual Organization, *Organizational Virtualness and Electronic Commerce, Proceedings of the 2nd International VoNet – Workshop*, September 23-24

[16] Padgett, J. F. (1980). Managing Garbage Can Hierarchies. *Administrative Science Quarterly 25*: 538-604.

[17] Papows, J. (1999) Enterpise.com, Market Leadership in the Information Age, *Nicolas Brealey Publishing*, London

[18] Phelan, S.E. (1995): From Chaos to Complexity in Strategic Planning *(working paper), La Trobe University*

[19] Smith R.D (1998) Social Structures and Chaos Theory, *Sociological Research Online,* Vol. 3, No. 1

Electronic formation of a contract

Urška Mikl
University of Maribor
Faculty of Law
Mladinska ulica 9
2000 Maribor
urska.mikl@uni-mb.si

Keywords: e-commerce, e-contract, offer, acceptance, formation of contract

Received: July 15, 2001

This paper analyses main legal problems that are related to formation of e-contracts. Because of the different legal consequences it is highly important for parties to be aware and understand the basic phases of contract formation process. For a contract to exist, usually one party must have made an offer, and the other must have accepted it. Once acceptance takes effect, a contract will usually be binding on both parties. A variety of procedures are available for forming an electronic contract, such as e-mail, EDI etc. Therefore, we have established that an electronic communication could create a legally binding contract. Business and other contracting parties should define and articulate most of the rules that will govern electronic commerce. Internationally, UNCITRAL Model law establishes rules and norms that validate and recognize contracts formed through electronic means, sets default rules for contract formation and governance of electronic contract performance.

1 Introduction

Electronic commerce enables fast interactivity and business transactions between parties all over the world. Business transactions need a set of legal rules to apply to them. In international commerce the obvious choice is contract, a legal institution that determines the content of obligations and rights by mutual consent of both parties. A contract determines a flexible, yet legally binding mechanism that follows general principles of contract law – freedom of contract.

Electronic commerce is generally defined as an individual transaction by means of electronic messaging. Though it is mainly used in doing business it is quickly finding its way in other fields of our lives. It is crucial that the e-business is conducted by electronic means and with high level of automation. Unfortunately particular phases of any transaction cannot be conducted in this way (e.g. delivery of goods). Of course there are also transactions that can be entirely conducted by electronic means (e.g. acquisition of a software). The question is whether a transaction concluded and conducted entirely by electronic means can be treated as a legally binding contract.

UNCITRAL Model Law on Electronic Commerce regulated legal framework concerning formation of e-contracts on international level back in 1996. The European Communities Directive on Electronic Commerce followed it in 2000.[1] This directive determines, among other dispositions, that Member States of the European Union are bound to change their legal systems in a manner that they permit formations of contracts by electronic means.

Slovenia soon followed international initiative and in 2000 enacted Law of Electronic Commerce and Electronic Signature (Official Gazette RS, No. 57/2000, CECES).

This paper analyses main legal problems that are related to formation of contracts such as offer and acceptance, time and place where e-contract are concluded. This paper will also consider the question whether the existing legal rules are suitable or is there a necessity to establish a new set of rules. The paper does not discuss the question of electronic signature, although it is closely related to formation of contracts by electronic means.

2 The conclusion of e-contracts

In a general manner formation of contracts is widely covered by both municipal and international law. In Slovenia the main source for law of contract is Law on obligations (Official Gazette SFRJ, No. 29/78 - LOR). However, Slovenia has adopted the new Code of

[1] OJ EC L 178, p.1-16, 17.7.2000

obligations that shall enter in force on 1. January 2002, which recodifies the existing legal system of contract law. However, in international commerce the main source worldwide governing sales law is the UN Convention on International Sales of Goods (Official Gazette SFRJ – International treaties, No. 10/1984 – CISG). When dealing with the CISG it is essential to pay attention to criteria of its applicability (Article 1). If the CISG is to be applied in an international sales contract, the parties must not only have their place of business in different countries, but these countries must also be contracting States to the Convention at a given time (article 100) or, where this criterion of applicability set forth in article 1(1)(a) is not met, the rules of private international law of the forum must lead to the law of a Contracting State, as indicated in article 1(1)(b). As far as the first criterion is concerned, it makes no difference whether the contract is concluded electronically or by any other means, since the only required feature is that the countries in which the parties have their place of business are Contracting States. As far as the second criterion is concerned, the use of electronic means (as opposed to more traditional means of communication) when concluding international sales contracts becomes relevant where the rules of private international law of the forum refer, as a connecting factor, to the place of conclusion of the contract. In this case, the determination of place where the contract has been concluded may cause difficulties due to lack of specific rules on this issue. Where, however, the rules of private international law of the forum do refer to connecting factors different from the place of conclusion of the contract, as do for instance the 1980 Rome Convention on the Law Applicable to Contractual Obligations, the use of electronic means should not lead to other problems than that encountered when concluding contracts by traditionally means. Therefore, at least for that area it does not appear that electronically concluded contracts should be treated differently from contracts concluded by any other means.[2]

2.1 Methods of contract formation

A contract is formed when all parties agree on its essential terms (article 26 LOR). Contractual obligations are established, changed or finished by contracts. Here an obligation is a relation between two persons, which is a legal basis for the transaction. If there is no agreement between the parties as to the terms of the contract, no contract is concluded. The agreement is reached only when both parties assent to intention to create legally binding relation as to the terms of contract. Intention of the party must be expressed as stipulated in different legal provisions. It can be expressed orally, with conventional practice or by conduct that can without a doubt express its existence. Expression of the intention must be without mistakes and serious (article 28 LOR).

Both parties agree on formation of contract, its type and its contents. Article 23 of the CISG states that a contract is concluded at the moment when an acceptance of an offer becomes effective in accordance with the provisions of the CISG (article 23).

Contracts can be concluded by oral or written agreement. Agreement can be implied by conduct of the parties. And with regard to e-commerce, they can be formed electronically. A variety of procedures are available for forming electronic contracts[3]:

1. Electronic mail ("e-mail"): By exchanging e-mail communications, the parties can create a valid contract. Offers and acceptances may be exchanged entirely by e-mail, or can be combined with paper documents, faxes, and oral negotiations.[4]
2. Web Site Forms: In many cases a web site operator will offer goods or services for sale, which the customer orders by completing and transmitting an order form displayed on screen. When the vendor receipts the order (that is acceptance of an offer), a contract is formed. The goods and services may be physically delivered off-line.
3. Electronic Data Interchange ("EDI"): EDI involves the direct electronic exchange of information between computers. The data is formatted using standard protocols so that it can be implemented directly by the receiving computer. EDI is often used to transmit standard purchase orders, acceptances, invoices, and other records, thus reducing paperwork and the potential for human error. These exchanges (which are sometimes made pursuant to separate EDI trading partners agreements) can create enforceable contracts.

3 Online Offers and Acceptances

Before parties enter into a contract, each of them usually gathers data about their perspective partners. Only then the party decides whether she'll attempt to carry out the transaction or not. The contracting parties are now engaged in negotiations (this is not compulsory phase in the contract formation process). The parties can negotiate every details of a contract and simply end negotiations by making an offer or by termination of the negotiations.

[2] Legal aspects of electronic commerce, UN Commission on International Trade Law, Working Group on Electronic Commerce, 38. Session, New York, March 2001, p. 6

[3] See T. J. Smedinghoff, Electronic contracts and digital signatures, p. 2

[4] E-mail could be compared with regular mail. More about the issue see at acceptance.

3.1 Offer

In order to constitute an offer to sell or buy goods in international commerce, a proposal must meet certain minimum requirements. The offer is parties' proposal for concluding a contract addressed to the specific person (paragraph 1 article 32 LOR). Such a proposal must contain all essential terms of a contract. If not, it is considered merely as an invitation to make an offer.

According to the CISG a proposal for concluding a contract addressed to one or more specific persons constitutes an offer if it is sufficiently definite and indicates the intention of the offeror to be bound in case of acceptance. A proposal is sufficiently definite if it indicates the goods and expressly or implicitly fixes or makes provision for determining the quantity and the price (article 14). Therefore the key components of a CISG offer are specificity, definiteness and an indication to be bound.[5] As far as the element of specificity is concerned, it appears to make no difference what form of communication one uses. In respect of this substantive feature of the offer, there are, in other words, no more problems intrinsic to electronic forms of communication than to other forms of communication.

This is basically also true in respect of the required intention to be bound, which distinguishes an offer from an invitation to make an offer. Some kinds of transactions involve a preliminary stage in which one party invites the other to make an offer. This stage is called an invitation to treat. Like Slovenian LOR, the CISG distinguishes between an offer, which binds the offeror, and an "invitation (that others) make offers" (invitatio ad offerendum), which have no such binding effect.[6]

In most cases, an offer will be made to a specified person. However, offers can be addressed to a group of people, or even to the general public. Article 33 LOR defines so-called "general offer". General offer is a proposal made to indefinite number of persons containing all essential term of a contract (offerta ad incertas personas). Such a proposal is considered as an offer unless circumstances of the case or legally recognized custom dictate something else. But under paragraph 2 article 14 CISG a proposal not addressed to one or more specific persons is interpreted merely as an invitation to treat. However, one who clearly indicates an intention to be bound by such a proposal will be treated as having made an offer. Yet, the new Slovenian Code of obligations is in conformity with the CISG.

Having the system described above in mind it is possible to analyse individual legal consequences of formation of contract by e-mail or EDI. The offer is always sent to a specific person if sent by e-mail. EDI on the other hand usually means closed communication between known parties. Such a proposal is considered (in both cases) as an offer and not as an invitation to treat.

Catalogues, advertisements, price lists etc. can also be sent per e-mail. Generally, advertisements in newspapers, radio and television, catalogues, brochures, price lists are considered as invitations to treat (paragraph 1 article 35 LOR). However, the sender of such invitation is liable for prejudice cause to offerors, if he doesn't accept the offer without substantiated reason. The same interpretation might be extended to web sites through which a prospective buyer can buy goods: advertisement on a web site should be considered as invitation to treat.[7]

Yet the doctrine has also elaborated an opposite opinion. Catalogues, advertisements, price lists, etc. could be considered as an offer if they contain all essential terms of a future contract. They should contain precise description of goods, price, time, place and method of performance, etc.[8] To avoid problems whether your message is an offer or invitation to treat it is recommended that in a case of doubt there should be a clarification stipulating that a message sent is not considered as an offer. There is also a possibility to derogate the system of an irrevocable offer (article 25 New Code of Obligations) and replace it by the mechanism mentioned above.

Form of an offer

Form of an offer is usually not defined. When the law determines a special form as an element for the contract's validity, an offer is valid only if it is in the same form (paragraph 1, article 38 LOR). The CISG also adopted the widely accepted principle of informality. The conventions itself does not demand any special formality when issuing an offer. Its scope of application (international sale of goods) helps to explain why the authors of the convention had chosen that regulation. As a result, exchange of e-mail messages should suffice to form a legally binding contract under the CISG.[9]

Furthermore besides oral and written there are also so called "real offers". These are made when a party sends another party the subject of the offering.[10] There is no reason why an electronic offer would have less validity as the offers mentioned above. Of course there is always a question of reliability of electronic communication and possibilities of introducing evidence in court.

[5] H. Bernstein, Understanding the CISG in Europe, p. 33
[6] H. Bernstein, Understanding..., p. 36
[7] Legal... p. 13
[8] V. Kranjc, Prodaja po katalogih in oglasih, Pravnik, 9-10/1992, p. 408-415
[9] Legal ..., p. 10
[10] Stojan Cigoj, Obligacijska razmerja, ČZ Uradni list SR Slovenije, Ljubljana, 1978, commentary to article 38, p. 31

3.2 Acceptance

Acceptance of an offer means unconditional agreement to all the terms of that offer. Typical offline acceptances include written and oral communications, as well as acceptance by conduct. Their online counterparts include acceptance by e-mail or any other form of electronic message, and by conduct such as clicking on a button or downloading contents.

We can expect that case law will allow acceptance by e-mail as a reasonable practice if all parties have the means to communicate with each other. If acceptance was made by another communication method, like a letter or fax, it should be treated as an acceptance by classic means. Acceptance does not necessarily have to be sent in the same way as the offer.[11]

The acceptance becomes effective when the offeree's indication of assent reaches the offeror (article 39 LOR). Therefore LOR imposes receipt theory. The acceptance becomes effective also if the offeree sends goods or pays the price or makes anything else that could be reasonably treated as an acceptance of an offer. The intention to conclude a contract can be expressed explicitly, in silence or by conduct of offeror. The CISG does not deviate from LOR when dealing with acceptance. In either case the key to a successful acceptance is offeree's indication of assent[12].

3.3 Revocation of an offer and revocation of acceptance of an offer

LOR allows offerors to revoke their offers in case that revocation reaches an offeree before or at least the same time as the offer (paragraph 2, article 36).

According to article 15 CISG an offer, even if it is irrevocable, may be withdrawn if the withdrawal reaches the offeree before or at the same time as the offer. Article 15 concerns the offeror's right to withdraw an offer. Until a contract is concluded an offer may be revoked if the revocation reaches the offeree before he has dispatched an acceptance. However, an offer cannot be revoked: (1) if it indicates, whether by stating a fixed time for acceptance or otherwise, that it is irrevocable; or (2) if it was reasonable for the offeree to rely on the offer as being irrevocable and the offeree has acted in reliance on the offer. Unlike the right to revoke in article 16, the right to withdraw deals with offers which have never taken effect.[13]

Note the difference between LOR and CISG view on irrevocability of an offer. Under the CISG an offer without explicit denotation of bounding by the offer is considered revocable. However LOR enforces the opposite assumption. The CISG makes a distinction whether acceptance date is stated in an offer.

So what happens if an offeror revokes his offer while acceptance of his offer is already waiting in his e-mail box?

It is actually a question of the revocation on time. If an offeree received revocation of an offer before or at the same time as an offer, revocation was valid (paragraph 2, article 36 LOR). If revocation was not sent on time an offer binds the offeror. Also, as denoted in the CISG, if withdrawal was valid an offer does not bind the offeror. In the presented case acceptance of an offer was already sent. According to paragraph 1 article 16 CISG an offer could not be revoked if the revocation did not reach the offeree before he has dispatched his acceptance. If the offer was irrevocable, as in the example stated above, it could not be revoked as stated in paragraph 2 article 16 CISG.

Acceptance of an offer can also be revoked if the offeror received notification of revocation before or at the same time with the statement of acceptance (paragraph 3, article 39 LOR). When withdrawing acceptance of an offer receipt theory should be used as stated also in CISG, article 22.[14] When an offer is made to a number of persons whose identity is unknown to the offeror as, for example an web site advertisement, it is clear that it is impossible to revoke in the usual manner that is to say by sending a letter of revocation. In such a case giving equal publicity to the revocation may terminate the power of acceptance. Normally this is accomplished by using the same medium for the revocation as was used for the offer.

Regarding revocation CECES does not interfere with law of obligations. It only explicates when a party is deemed to have received a message. Unless otherwise stipulated between the originator and the addressee, the dispatch of a data message occurs when it enters an information system outside the control of the originator or of the person who sent the data message on behalf of the originator (article 9 CECES). Unless otherwise agreed, the time of receipt of a data message is considered time when it enters designated information system (article 10 CECES).

Unless otherwise agreed, regardless the paragraph 1 article 10 CECES, the receipt of a data message is considered to occur when the data message enters the designated information system if the recipient has designated an information system for the purpose of receiving data messages or when the recipient retrieved the data message if the data message is sent to an

[11] S. Cigoj, Obligacijska razmerja, p. 31
[12] Bernstein, p. 40
[13] Bernstein, Understanding ..., p. 37

[14] An acceptance may be withdrawn if the withdrawal reaches the offeror before or at the same time as the acceptance would have become effective.

information system of the recipient that is not the designated information system.

There appears to be, however, one instance where problems may arise if electronic messages are compared to more traditional ones, such as telegrams, letters, telex, as the Convention contains one provision that makes a distinction between these forms of communications. Namely, according to article 20(1) "a period of time for acceptance fixed by the offeror in a telegram or a letter begins to run from the moment the telegram is handed in for dispatch or from the date shown on the letter or, if no such date is shown, from the date shown on the envelope. A period of time for acceptance fixed by the offeror by telephone, telex or other means of instantaneous communication, begins to run from the moment that the offer reaches the offeree." Thus, for the purpose of deciding when the time for acceptance begins to run, a decision should be made as to whether the electronic message should be compared to a means of instantaneous communication rather than to a letter or telegram.

Paragraph 2, article 40 of LOR states that parties are present when an offer is sent by phone, teleprinter or direct radio link. In this cases parties are considered present because of their ability to express their terms without a delay. This definition is clearly not problematic in every day two-way communications (e.g. talking over a phone). E-mail is another story. It is more like a one-way communication, because offeror sends his offer to offeree that is usually not going to read it at once and consequentially responds with a delay. Considering this fact it is evident that we should deal with e-mail messages more like with a regular mail than with a phone call. EDI in contrast is without a doubt a two-way communication allowing involved information systems to instantly respond to each other's messages.

Effect of a Late Acceptance

What if the offeree sends his acceptance on time but this acceptance is delayed because of errors in transfer and reaches offeror too late (late acceptance). Acceptance was received too late and is therefore considered a new offer from offeree (article 43 LOR). Still, the contract can be concluded under the terms of the expired offer if the offeror received statement after the expiration of time for acceptance of an offer, which was sent in time and he knew or could have known that it was sent on time. In that case the contract could still be concluded despite obstruction in communication. The contract would not be concluded if the offeror notified immediately or on next workday after receipt of a statement or also before receipt of a statement but after expiration of time for acceptance of an offer, to offeree that because of this delay he is not bound by his offer (paragraph 3, article 43 LOR).

Paragraph 1 article 21 CISG also indicates an exception against paragraph 2 article 18 CISG. An acceptance is not effective if the indication of assent does not reach the offeror within the time he has fixed or, if no time is fixed, within a reasonable time, due account being taken of the circumstances of the transaction, including the rapidity of the means of communication employed by the offeror. A late acceptance is nevertheless effective as an acceptance if without a delay the offeror orally so informs the offeree or dispatches a notice to that effect (par. 1 article 21).

Acceptance of the offer cannot be revoked after it is in effect (from the moment offeror received statement of acceptance).

3.4 Offers and acceptances by computers using EDI

EDI is usually closed communication between known parties. A computer program that runs on a computer of one party can generate an offer and send it using EDI to a computer program running on another party's computer. An inventory computer program could, for example, identify low supplies and automatically generate an electronic purchase order to the vendor. Computer program automatically, in the name of its owner, makes an offer and sends it to other contracting party. The computer of other contracting party can also automatically generate the answer. This imposes the question whether this is an acceptance of the offer or merely an acknowledgement of receipt.

Partial solution to that problem is offered in the Slovenian Law of Electronic Commerce and Electronic Signature. Article 5 states whether a message originates from the sender - originator. When a message is generated by information system without human intervention it is treated as originating from the legal entity on behalf of which the information system is operating. Regarding addressee of the message this is not the case. In EDI computers usually automatically acknowledge transactions, e.g. acknowledgement of purchase order. Although usually that indicates a computer's ability to read and understand the message and cannot be always treated as a legal acceptance of an offer. Of course EDI messages could contain all obligatory information to be treated as such.

Slovenian Law of Electronic Commerce and Electronic Signature (CECES) allows contracting parties to decide whether to acknowledge receipt of the message automatically or by hand. Where originator has stated that the data message is conditional on receipt of the acknowledgement, the data message is treated as though it had never been sent, until the moment when the acknowledgement is received (article 7 CECES). Where the originator has not agreed with the addressee that the acknowledgement be given in a particular form or by a particular method, an acknowledgement may be given by any communication by the addressee, automated or otherwise or any conduct of the addressee, sufficient to indicate to the originator that the data message has been

received. Therefore acknowledgement is not treated as acceptance of an offer but only acknowledgement. of receipt, as would be a notice of receipt in usual written communication.

Article 8 CECES explicitly states that acknowledgement of receipt is not also acknowledgement of received content (a letter of confirmation).

Regardless of CECES the acceptance of an offer is still enforced by LOR and should be interpreted accordingly.

The person (whether a natural or legal one) on whose behalf the information system operates is liable for any offer (message) its information system generates automatically.[15]

4 Time of formation of a contract

LOR adopted so called receipt theory for time of formation of the contract. A contract is concluded at the moment when the offeror receives offeree's accept (paragraph 1 article 31 LOR).

Article 18, paragraph 2 CISG states that an acceptance of an offer becomes effective at the moment the indication of assent reaches the offeror. Regulation under CISG is mostly identical because article 24 defines when the declaration of acceptance reaches the offeror. Declaration of acceptance or any other indication of intention "reaches" the addressee when it is made orally to him or delivered by any other means to him personally, to his place of business or mailing address or, if he does not have a place of business or mailing address, to his habitual residence.

5 Place of formation of a contract

A contract is formed at the offeror's place of business or his permanent residence in time when he made his offer (paragraph 2, article 31 LOR). CECES also adopted provisions from LOR. Unless otherwise agreed between the originator and the addressee, a data message is deemed to be dispatched at the place where the originator has its place of business, and is deemed to be received at the place where the addressee has its place of business.

Electronic transmission of messages allows recipient to obtain his messages on a large distance from his information system. That is why place of information system is of no relevance. As in law of obligation permanent residence of legal entity is crucial. Cigoj argues that place of business of a contracting party that must perform a contract should be crucial when determining a place of formation of a contract.

[15] UNCITRAL Model Law on Electronic Commerce, Guide to enactment, paragraph 35.

6 Conclusion

E-commerce changed the way business is done and businesses quickly adopted them. When conducting business on-line a simple message can result in a contract formation. That is why businesses should be aware of relevant contract regulations. Slovenia has no special regulations that govern e-contracts. In conformity with that businesses should use other available sources of regulations, e.g. LOR or new Code of obligations. Any business conducted over the global Internet can quickly evolve in an international contract. That is when international regulations come in play. We have illustrated this in various examples in conformity with CISG. It appears that the CISG is, in general terms, suitable not only to contracts concluded via traditional means, but also to contracts concluded electronically. The rules set forth in the Convention do appear to offer workable solutions in an electronic context as well. Some of the rules, such as those relating to the effectiveness of communications, may need to be adapted to an electronic context.

In the near future new Slovenian Code of Obligations will take effect and we can concur that within the extent of this paper it is in harmony with CISG.

References

1. Drago Mežnar, Nekateri obligacijskopravni pogledi na sklepanje pogodb pri elektronskem poslovanju, RIP – računalniško izmenjavanje podatkov, Poročilo o raziskovanju v letu 1994
2. Janez Toplišek, Elektronsko poslovanje, Atlantis, Ljubljana 1998, str. 248
3. Janez Toplišek, Pravna zmogljivost elektronske pošte, Pravna praksa - Priloga, Gospodarski vestnik d.d., Ljubljana 1997, št. 11
4. Stojan Cigoj, Teorija obligacij, Uradni list RS, Ljubljana, 1998
5. Stojan Cigoj, Obligacijska razmerja, ČZ Uradni list SR Slovenije, Ljubljana, 1978,
6. Stojan Cigoj, Komentar obligacijskih razmerij, I. Knjiga, ČZ Uradni list, Ljubljana 1984
7. Vesna Kranjc, Poslovni običaji in gospodarske pogodbe, Gospodarski Vestnik, Ljubljana 1998
8. Poročevalec DZ, št. 15/2000
9. UNCITRAL, Legal aspects of electronic commerce, Possible future work in the field of electronic contracting, New York, 2001
10. UNCITRAL Model Law on electronic commerce with guide to enactment, 1996, http://www.uncitral.org/en-index.htm
11. Directive on electronic commerce, OJ EC L 178, p.1-16, 17.7.2000
12. Zakon o obligacijskih razmerjih, Ur. l. SFRJ, št. 29/78
13. Zakon o elektronskem poslovanju in elektronskem podpisu, Ur. l. RS, št. 57/2000

Intelligent systems applications

Matjaž Gams and Marko Bohanec
Jožef Stefan Institute, Department of Intelligent Systems, Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773644; fax: +386 1 4251038
matjaz.gams@ijs.si, http://ai.ijs.si/mezi/matjaz.html

Keywords: intelligent systems, intelligent agents, applications, information society

Received: September 12, 2001

In this paper we present and discuss the state of intelligent system applications, with special emphasis on Slovenia. We take the viewpoint that intelligent systems facilitate a new qualitative step towards the information society. Thus, they provide an important opportunity for any country to intensify its information, economic and social development. We present some intelligent system applications developed or being developed in our group: (1) EMA, an employment agent, (2) GIVE, a semantic web speaking system, and (3) DEXi, a decision modeling system. Based on past experience, we highlight the principal problems that occur in the development and exploitation of intelligent systems, and suggest improvements for the future.

1 Introduction

Information society is transforming the way we live. One of the key opportunities in this context is provided by intelligent systems (Goonatilake, Treleaven 1996; Gams 1998; Hopgood 2001). Intelligent systems simulate intelligence so that a typical user seemingly perceives it as a truly intelligent system. In reality, these systems have more or less preprogrammed patterns of human behavior. Due to limited application areas, most of cliché replies can be preprogrammed in advance. From the viewpoint of computer systems, they therefore somehow manage to fake simple intelligence, and from the practical viewpoint it does not really matter whether the user deals with true or simulated intelligence.

Intelligent systems are aimed at extending the applicability of computers and providing a technological basis for new and improved information services. Intelligent systems have found a variety of valuable applications in the areas such as:

- Manufacturing and design
- Business operations
- Finance
- Diagnostics and troubleshooting
- Claims processing and auditing
- Telephony
- Software industry
- Military and Space industry

Generally speaking, intelligent systems are useful because they provide more useful functions (Buchanan et al. 1999; Hedberg 1998). Computers are much cheaper and faster than humans, yet much dumber. In fact, computers can still be regarded only as very fast computing machines without any trace of true intelligence (Gams 2001b).

Classical computer systems can function 24 hours per day, all days per year, with overall small expenses compared to humans. The speed of communication and calculation enable a single computer to communicate with hundreds of human users at the same time. Advances in engineering intelligence combined with advanced hardware enable intelligent computer systems to compete with humans in more and more tasks.

Another important factor is the growth of the information society. A growing number of functions are supported by computers (Hamilton 1999). Humans are getting overloaded due to information overload. Huge amounts of data are processed by computers several orders of magnitude faster than by humans (Lewis 1999; Schwartz, Treece 1992).

Another very important property of information society is the growing space of all possibilities. Namely, technical possibilities in recent years grow much faster than they are – or can be – exploited. In reality this means that we are dealing with huge space of generally available intelligent information technology capabilities. This is quite different compared to other technologies, such as space flights. The production of a new space shuttle is so expensive that it is hardly economic. It is cheaper to exploit the current form for years before introducing a new one. Unlike this, intelligent systems in the information age enable more new applications than we humans actually manage to implement (Figure 1). In other words – while space flights are restricted only to the richest and strongest countries in the world,

practically every qualified information technology (IT) group can in principle develop world-class computer programs. This is slightly similar to the times when everybody could grab new land. All that was needed was a good horse, a good idea and determination. In analogy, IT horses are cheap and freely available.

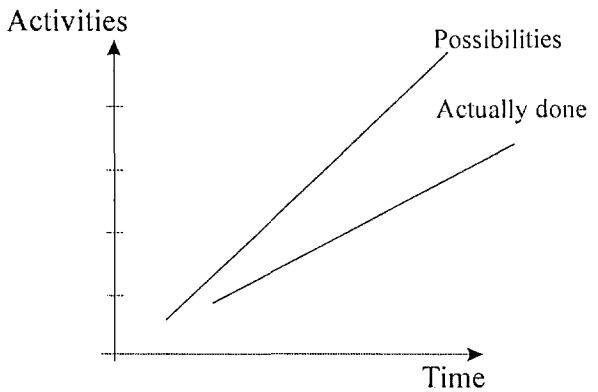


Figure 1: The information society enables more applications than we humans manage to implement. The gap is widening.

The information society also introduces major changes in the scientific community. Knowledge is not only stored in books or remote encyclopedia – it has become easily available throughout the Internet. IT has significantly changed the way we create and use knowledge.

Generally, we accept the assumption that scientific disciplines produce new ideas spurring further progress in the whole society. Intelligent systems are no different, with additional observation that these systems have a huge area of potential applications. Any system that has improved properties in the sense of intelligence is more capable and potentially more acceptable to the users.

In this light we analyze the relation between science, development and applications. Unfortunately, scientific and development communities still seem two worlds apart. When analyzing these differences and problems regarding applications we try to identify generic problems that are probably present in many countries. On the basis of these analyses, improved relations between intelligent systems and artificial intelligence (AI) research versus applications can be proposed.

In reality, the gap especially between AI research and applications is wide indeed. Many if not majority of AI researchers and professors have not cooperated in any real-life application and many if not majority of AI development engineers have not done any formal research at all. Therefore, the question at hand is why and how these strained relations occurred and which are the reasons prohibiting better cooperation?

In our department, we have several 100 man-year experience in intelligent systems. We have cooperated in several tens of applications, in several thousands of scientific publications and personal communications with several hundred researchers, developers and engineers in most of European countries and USA.

By analyzing problems in Slovenia and Europe as countries we have some experience with, we shed some light not only on the problems but also on a couple of successful applications that were developed or are being developed in our Department of Intelligent Systems. The developed systems have been already used by several ten thousand users monthly, they controlled the production of several hundred thousands tons of products and overall had a considerable impact on our society.

2 Problems with cooperation

The first difference between the scientific and development community is in different motivation and evaluation. The number and quality of publications determine the quality of researchers, while developers are evaluated mainly by their commercial success.

On average, in both communities a researcher or a developer is highly above average with respect to knowledge, intellectual and technical abilities. Many of them would be successful either in scientific or in practical development tasks.

It is generally accepted that a successful application demands the fulfillment of several successful attributes, and that many of them are not well known. For example, the key factor is the interest or motivation of the management of a particular company. Many researchers would probably regard as the major factor the novelty of the idea/product, but this is just one of important factors, not the major one.

In our experience, the single major factor was indeed the desire and knowledge of the top manager. If that person was in favor of research, then sooner or later his/hers company found some cooperation with researchers. And sooner or later one of those cooperations resulted in important advanced systems.

The following reservations are often encountered in the development/engineering community:

- Academic research and practical applications are worlds apart. Academic institutions are interested in their academic puzzles often not related to real life or real-life applications. This academic knowledge is often nice for teaching simple cases in formal domains but often irrelevant for commercial applications.
- The government supports academic institutions regardless of their influence, success and relation to human society, while commercial companies must

take care of their own, are evaluated by strict real-life commercial criteria, and have no guarantee for survival. Instead of funding academic research, governments should support research departments in commercial companies.
- By supporting R&D departments at national academic institutions, governments actually subsidize dumping competition. These research groups can offer dumping prices, thus eliminating true marker capitalism and economy.
- Commercial companies should be motivated to order research projects at national institutions by tax reductions and other systematic national support.
- Commercial companies fear cooperation with academic institutions due to bad experience with famous researchers who did not deliver products and projects in promised time, and faced practically no consequences.

Academic community, on the other hand, faces other problems. There are individuals or groups that are not interested whatsoever in any cooperation with industry due to several reasons such as their purely theoretical type of research or vague perceptions that applications only distract true researchers from high-quality work. However, those researchers that are willing to cooperate and see reasonable motives in doing so for both sides, often mention the following problems:

- Cooperation with industry is not stimulated enough compared to pure research – those interested in cooperating with industry produce less scientific publications, get lower personal evaluation marks and thus lower income and slower career progress. Since the incomes are more or less fixed in governmental institutions, additional projects and applications hardly provide additional incomes.
- Commercial companies and governmental institutions often waste huge amounts of money for non-functional matters, such as employing disproportional number of employees, luxuries, or even half-legal matters while often disregarding the possibilities of cooperation with research institutions which by definition can not be all commercially successful.
- Business institutions are not systematically motivated for cooperation with research – either by tax reductions or by projects supported by the government
- Many top-level managers are not inclined to take any risk with new research; they are fully overloaded with actual production / cash-related activities.
- Governmental institutions are often stiff and inflexible. Each risk is punished while successful actions are hardly rewarded.

It is a bit surprising that a reasonable proportion of the problems with too little cooperation between research and industry is related to small interest of top-level managers, but that is clearly the case at least in Slovenia, and probably also in Europe. For example, our Intelligent Systems group from time to time offers cost-less applications to specific companies. The idea is simple: by introducing successful applications – prototype or complete – we hope to increase cooperation. But important business companies often tend to decline such free cooperation due to several reasons that can hardly be accepted as reasonable. For example, one of the arguments is often that the company's data are confidential and sensitive, so they cannot be released to outside research organizations for analyses. But such data can easily be transformed into unrecognizable strings, and there are means to strictly impose law and regulations so as to guarantee its secrecy. The Internet and email applications are often declared unsafe, thus enabling stealing and misconduct. But compared to real life, e.g., problems with business fraud, and taking into account the availability of successful counter-measures in electronic business, these problems first of all demonstrate unreasonable fears and lack of knowledge at the top.

Some of these problems were encountered also at the post-graduate courses where students proposed advanced applications of intelligent systems and were often not allowed to proceed due to the fears of local managers. Actually, there was another fear behind – that the students will supercede their bosses, who will consequently degrade on the hierarchy ladder.

3 Some real-life applications

Here we briefly describe a couple of intelligent-systems applications. They were designed in the Department of Intelligent Systems at the Jožef Stefan Institute in Slovenia. The group has from 20 to 30 members and several hundreds of man-year experience in intelligent systems.

3.1 EMA: Employment agent

The first system we describe is the EMA employment agent (Gams 2001a; Gams et al. 1998). The basic task of the system was to provide employment information (Figure 2). About seven years ago the system offered over 90% of all nationally available vacant jobs. At that time, no other country provided similar percentage of all jobs on the Internet. Of course, due to 2 million inhabitants, absolute numbers were still small compared to large countries. On the other hand, it provides further evidence that Figure 1 showing the growing area of possible applications is indeed correct – how else could we provide a better penetration of the Internet in a specific area knowing that the average income of the country is well bellow the European Union average?

One big advantage is obviously size – in a small country it is enough to provide just one global database while large countries have to take into account federal and local specifics.

When presenting EMA, we would like to emphasize two additional factors – EMA consisted of several tens of modules, including job-description ontologies, natural speech modules in English and Slovenian, and global automatic data-wrapper as a kind of universal agent employment communications. However, all these advanced functions were not of major relevance for the business. Most of the users just wanted information about free jobs.

The major problem we faced with EMA was organizational - due to inappropriate legislation, all job applicants have to receive a hard-mail reply. Since interesting jobs offered through the Internet got hundreds and even thousands of job applications, the institutions had to send as much written replies. In return, many of job applicants complained about being returned down, thus producing additional delay, and finally, institutions were unsatisfied because of the additional work. While EMA enabled substantially improved national job services, institutions were dissatisfied because of the archaic law. Therefore, since the law regulations did not follow technological IT advances, the system was not as helpful as it could and should have been.

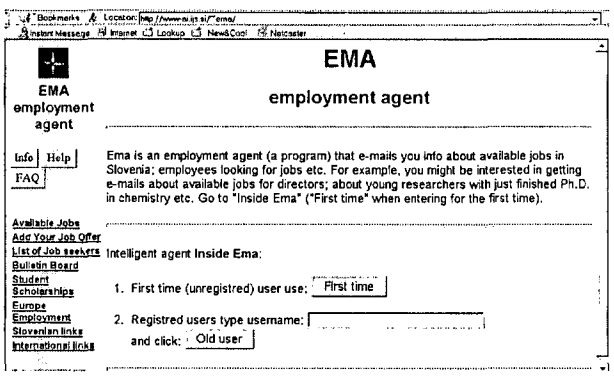


Figure 2: Intelligent agent EMA offered over 90% of all nationally available jobs seven years ago. This was a world-class result showing that Internet applications indeed enable even small countries to produce relevant systems.

There were many single stories indicating the advantages of the system. For example, a brilliant student wanted to return to Slovenia from USA and with EMA's help he found an interesting vacant job of a faculty assistant. When he applied, no competing candidate met his characteristics. But the professor already had an agreement with some local friend to hire another, reasonably decent candidate. So they decided to declare all job applications invalid and retried until the desired candidate succeeded.

It seems strange, but the result of these problems was that EMA was allowed to offer only those jobs that the employers wanted to. Consequently, any job offer was allowed to remain hidden.

In a sharp contradiction, some developed countries like USA a bit later introduced all governmental national job offers (EMA included both governmental and nongovernmental) through the Internet, but there is no way a governmental job can be hidden from the eyes of the Internet public.

The EMA experience is therefore mixed – we managed to introduce several new advanced functions, implemented a system that was used by around 10% of all our population in best months, but had to degrade the system substantially because of the lack of appropriate legislation. Better to say – the system was developed due to the encouragement of the minister of science and the director of the National Employment Service, who strongly supported the idea. But they could not modify the national laws, which later hampered the system.

Overall, the system was the most often used intelligent system in Slovenia so far, and the percentage of Internet-available jobs temporarily put it at top world level.

3.2 GIVE: Semantic-web speaking system

Another Internet-based system is GIVE – a national project that enables multimedia access and use of the Internet through telephones (Figure 3). The idea is that due to several circumstances the users need Internet functions through mobile or stationary phones – e.g., one drives a car and can only use a phone.

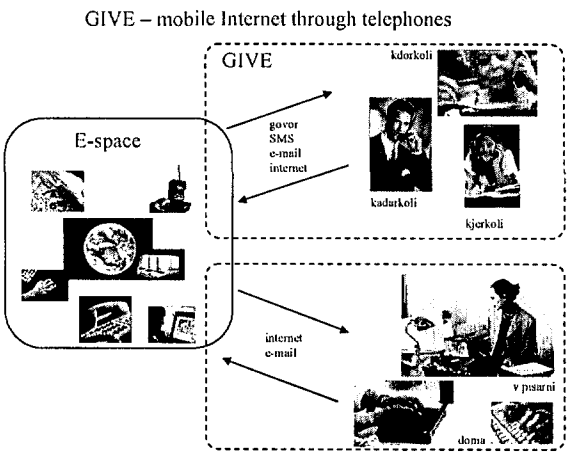


Figure 3: Project GIVE provides Internet through phones in Slovenian language.

At the first stage, there is a prerecorded human speech and dynamically generated program speech. The speech system enables voice output from any URL address by parsing HTML pages and speaking it in Slovenian. More relevant, the system enables any user to put text information through the GIVE system into GIVE's local databases. The most common access is through a specific input telephone number and the telephone

number of a particular institution or a person. In this way, for instance, anybody can get information about some governmental institution or service even though no human is present there late at night.

The GIVE project is still in progress, but several modules and functions already function fully.

3.3 DEXi: Decision modeling system

DEXi is a computer program developed in collaboration with the University of Maribor, Faculty of Organisational Sciences. The program facilitates the development of qualitative multi-attribute models (Figure 4). Such models are useful for supporting complex decision-making tasks, for instance, where there is a need to select a particular option from a set of possible ones so as to best satisfy the goals of the decision maker. A multi-attribute model is a hierarchical structure that represents the decomposition of the decision problem into subproblems, which are smaller, less complex and easier to solve than the complete problem.

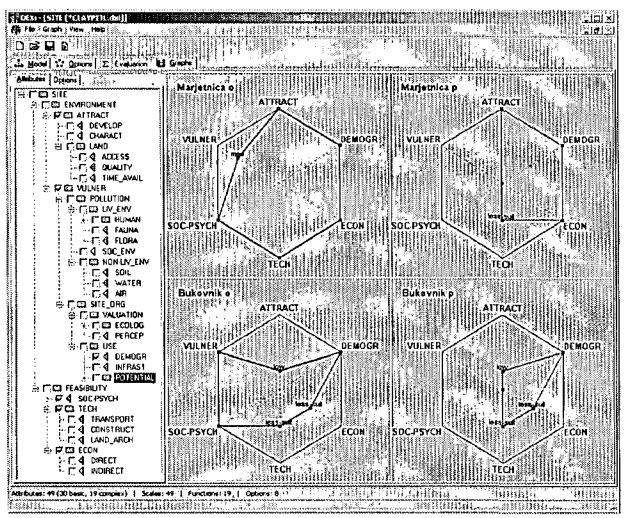


Figure 4: Decision modeling system DEXi facilitates the development of qualitative multi-attribute models (left) and their application for the evaluation and analysis of options (right). The case shown is from the area of land-use planning.

DEXi supports two basic functions: the development of models and their application for option evaluation and analysis. The models are developed by defining:

- attributes: qualitative variables that represent decision subproblems,
- tree of attributes: a hierarchical structure representing the decomposition of the decision problem (Figure 4, left),
- utility functions: rules that define the process of aggregation that takes place from the bottom to the top of the hierarchy.

In the evaluation and analysis stage, DEXi facilitates:

- description of options: defining the values of basic attributes, i.e., the leaves of the tree,
- evaluation of options: a bottom-up aggregation procedure according to utility functions,
- analysis: what-if analysis and selective explanation of options,
- graphical presentation of the evaluation results (Figure 4, right).

DEXi is not an intelligent system by itself, but contributes to the development of intelligent systems. Namely, the models developed by DEXi can provide a firm basis for intelligent decision support; the models can be either embedded into other intelligent systems, such as agents, or used in a stand-alone way to support the analysis and justification of decisions.

So far, DEXi and its predecessor DEX (Bohanec, Rajkovič 1990) have been applied in more than fifty real-life decision problems (Urbančič et al. 1998), which involved thousands of users. Recently, there were a number of applications in health-care (Bohanec et al. 2000) and industry (Bohanec, Rajkovič 1999) for decision problems related to land-use planning (Figure 4), ecology, and the evaluation of enterprises, products, projects and investments.

4 How to improve cooperation

Successful cooperation between research and application community is seen as one of the keys for successful progress in the information society. The following arguments seem particularly important:

- Researchers at faculties and institutions have great amounts of relevant and most advanced knowledge. That knowledge enables introduction of new-generation real-life systems with potentially wide influence and major market success. Researchers and professors represent the intellectual elite. Many if not even most of them are quite interested in cooperation with industry.
- Regular contacts between industry and academia are beneficial for all sides. Researchers get state-of-the-art knowledge about current level of commercial applications, and developers get some ideas about major research directions probably influencing future commercial progress.
- Regular contacts can be established on a good-will cooperation basis without major costs or funding. Conferences, public lectures and other forms of knowledge representations offer the easiest and very efficient contacts.
- Research community is diverse and heterogeneous. Therefore, directors are not good representatives of what is done at an institution. Researchers and developers should meet at department level searching for specific areas of interest.

5 Conclusion

Intelligent systems are one of the most attractive fields of IT. Due to the progress of computer raw power and advanced engineering-intelligent applications, these systems are becoming more cost-beneficial compared to humans in more and more tasks. Computer systems are much faster and cheaper than humans and with a bit of simulated intelligence they outperform humans in several bureaucratic or every-day tasks.

Information society enables great opportunities not only for the most developed countries. Small countries can quickly accept knowledge from the most developed countries and quickly reapply it in small countries for world-novel problems.

The major problem in developing intelligent systems applications is the cooperation between academic and engineering community. Improved cooperation could serve well both, but there are unreasonable obstacles indicating the need for systematic improvements.

References

M. Bohanec, V. Rajkovič, DEX: An expert system shell for decision support. Sistemica 1(1), pp. 145–157, 1990.

M. Bohanec, V. Rajkovič, Multi-attribute decision modeling: Industrial applications of DEX, Informatica 23, pp. 487-491, 1999.

M. Bohanec, B. Zupan, V. Rajkovič, Applications of qualitative multi-attribute decision models in health care, Int. Jou. of Medical Informatics 58-59, pp. 191-205, 2000.

B. Buchanan, S. Uthurusamy (eds.), Innovative applications of artificial intelligence, AI Magazine, Spring 1999.

M. Gams, Information society laws, in Information Society 98 (ed. M. Gams), Državna založba Slovenije, pp. 1-4, 1998 (in Slovene).

M. Gams, A. Karalič, M. Drobnič, V. Križman: EMA - An intelligent employment agent, Proc. of the Forth World Congress on Expert Systems, Mexico, pp. 57-64, 1998.

M. Gams: A Uniform Internet-communicative agent, Electronic Commerce Research 1, Kluwer Academic Publishers, pp. 69-84, 2001a.

M. Gams: Weak intelligence: Through the principle and paradox of multiple knowledge, Advances in computation: Theory and practice, Volume 6, Nova science publishers, inc., NY, ISBN 1-56072-898-1, pp. 245, 2001b.

S. Goonatilake, P. Treleaven, Intelligent systems for Finance and Business, John Wiley & Sons Ltd, 1996.

S. Hamilton, Taking Moore's Law into the next century, IEEE Computer, January, pp. 43-48, 1999.

S. R. Hedberg, Is AI going mainstream at least? A look inside Microsoft research, IEEE Intelligent Systems, March/April, pp. 21-25, 1998.

A. A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001.

T. Lewis, Microsoft rising, IEEE Computer Society, 1999.

E. I. Schwartz, J. B. Treece, Smart programs go to work, Business Week, pp. 97-105, 1992.

T. Urbančič, V. Križman, I. Kononenko, Review of AI Applications, Jožef Stefan Insti-tute, Ljubljana, Report DP-7806, 1998.

Stratified frameworks

Colin Atkinson and Thomas Kühne
AG Component Engineering
University of Kaiserslautern, Germany
{atkinson,kuehne}@informatik.uni-kl.de

Keywords: components, refinement, architecture, stratification

Received: August 16, 2001

Finding the optimal level of abstraction at which to document the architecture of a system has long been a problem in software engineering, particularly for large and complex systems. In this paper we argue that providing just a single abstraction level is inappropriate, and that instead, multiple architectural descriptions should be developed and documented, each capturing a specific aspect of a system's realization at a particular level of abstraction. Further, we argue that such a stratified architecture is especially valuable when used to organize a framework. After explaining the basic motivation for the work, and defining the basic principle of stratification, the paper illustrates the approach in conjunction with a small case study. The paper then discusses the methodological issues associated with the creation, application and maintenance of stratified frameworks.

1 Introduction

Although it is generally accepted that an optimal representation of software architecture includes multiple views, the structural view still invariably plays the central role. However, today's enterprise systems have reached such a level of complexity that a single "components + connectors" view of a system's structure as popularised by Garlan and Shaw [1] is no longer adequate. Not only has the scale and functionality of software increased, but with the advent of component technologies the boundary between the application and system level services (so called middleware) has significantly blurred [2]. As a result, it is no longer clear what level of abstraction provides the best overall structural description of complex software systems.

In the following we argue that for complex, industrial-scale software systems it is no longer appropriate to think in terms of just one structural view of a software system, but that instead it is better to provide multiple structural views (or strata), each elaborating upon a different aspect of the system's overall functionality. Since the strata in such a multi-level architecture each provide a complete description of the system's structure, they are not layers in the traditional layered architectural style. On the contrary, they each describe the entire structure of the system but at different levels of abstraction and with respect to different aspects of the system's overall functionality. Different stakeholders can therefore understand the system's structure at the level of abstraction which best matches their individual needs or tasks.

The advantages of such a multi-level view of system structure are twofold. First, since the relationships that connect individual strata reflect those that would result from a process of top-down, step-wise refinement, starting from the highest-level structural view, the approach can serve as the basis for an architecture development methodology. Instead of describing the whole structural architecture of a system in one fell swoop, system developers can focus on separate aspects of the system individually, and gradually work towards the detailed description [3]. A multi-level view of system structure therefore provides a powerful basis for separation of concerns and step-wise progress in the software development process. Moreover, with a reasonable degree of rigour in the structural description [4], the approach can provide a foundation for a generative approach to software development.

Second, by viewing the higher level strata as an end in themselves rather than a means to an end the concept of multiple views forms the basis for a powerful model of component-based enterprise frameworks. Regarding the upper-level structural views (or strata) as merely stepping stones in the creation of the real (most detailed) view makes them second class citizens, and makes them vulnerable to the neglect that befalls most intermediate "documentation" artefacts in software engineering. However, by viewing all strata as first class citizens of a framework, they become more stable software assets that are related more in space than in time. Multiple structural viewpoints within a component-based framework enhance the range of possible parameterisation points, and thus facilitate more flexible and straightforward instantiation.

In this paper we introduce and explain the concept of architecture stratification, and show how it can be used to increase the flexibility of component-based enterprise frameworks. We give a small example of what a stratified framework looks like, and then elaborate upon the processes by which such frameworks can be created, instantiated and evolved. Finally we discuss related principles and research areas in software engineering.

2 What Is Stratification?

In this section we explain the basic motivation for architecture stratification, and describe fundamental principles for its realization. We then consider the ramifications of stratification from the perspective of individual system components.

2.1 Which Architecture is *the* Architecture?

Consider a very simple client-server system in which a client component requests some form of service from a remote server component. At the highest level of abstraction the structure of this system could be captured using a class diagram of the kind in Figure 1.

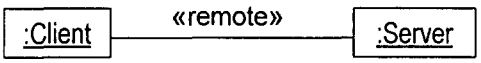


Figure 1: High-Level Client-Server Architecture

The stereotype, «remote», is used here to capture the high-level properties (or semantics) of the interaction between the client and server. This view of the system's structure conforms to the fundamental "components + connectors" concept of software architectures popularized by Garlan and Shaw [1]. However, it is not the only structural view that has this property. It is perfectly possible to provide alternative representations of the system's structure that conform just as well to "components + connectors" concept of architecture. Figure 2, for example, provides an alternative view of the structure of the system that explains how ORBs (Object Request Brokers) serve to mediate the interaction between the client and the server.

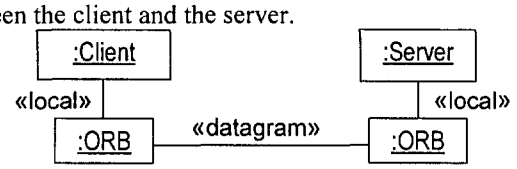


Figure 2: Low - level Client - Server Architecture

Figure 2 represents an equally valid structural description of the system. The only difference between the two views is the level of abstraction at which the structure is described. Figure 1 provides a high-level view of the system, involving a semantically rich interaction between the client and server, while Figure 2 gives a more detailed view that elaborates upon how the interaction between the client and server is realized.

Since both provide acceptable structural representation of the system consistent with the "components + connectors" model, the immediate question that arises is which is the correct, or the best, one? In other words which diagram represents the *real* architecture? The implicit answer in most contemporary methods is that the lowest, non-executable description of the system structure represents *the* architecture. Descriptions that are executable are typically regarded as an "implementation" or a "program" rather than an architecture, while higher-level descriptions are generally viewed as being incomplete or merely models.

Were one forced to select just *one* architecture to represent the structure of the system, the more detailed version of the two architectures would probably be the best choice. However, a better solution is obtained if more than one architectural description can be chosen. This is because higher-level views of the structure provide descriptions of the system that are of more value to certain stakeholders than the lowest-level view. For example, the user or customer is probably going to find Figure 1 a much more useful representation of the system than Figure 2. Therefore, higher-level views are valuable assets in their own right, and do not have to merely represent stepping stones along the road to the "real" (most detailed) architecture. In other words, rather than being just a means to and end, higher level structural views represent a valuable "end" in themselves.

The basic premise underlying the concept of stratification is that a complete representation of a system's architecture should contain all relevant abstraction levels. Moreover, the relationships between the different levels should be carefully and explicitly documented so that they represent a single coherent, whole, rather than a set of disjoint structural viewpoints.

2.2 Interaction Refinement

Most component and architecture description languages include the concept of "connectors" to capture the interaction protocols through which components, or architectural units, can be connected together. Ideally these connectors should be first class citizens of a description language, amenable to the same set of manipulations as components themselves. However, to date, no entirely satisfactory model for connectors has been found. Introducing the notion of abstraction levels, referred to as strata in the following, enables connector semantics to be understood in terms of lower level components.

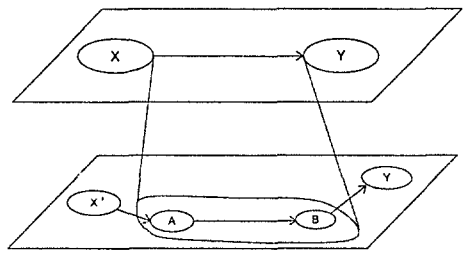


Figure 3: Interaction refinement

Figure 3 shows how a high level interaction between components X and Y is realised in terms of additional interactions and components at a lower level of abstraction. In other words, the interaction between X and Y is reified into components (A and B) and other interactions which reside one stratum lower in the hierarchy. Viewing connectors as collections of components at lower architectural strata, provides a clean model for their access and manipulation.

Note that the original component X in Figure 3 has to be adapted to X' in order to communicate with A rather than

with Y. This is an important aspect of stratification and is discussed further in section 2.3.

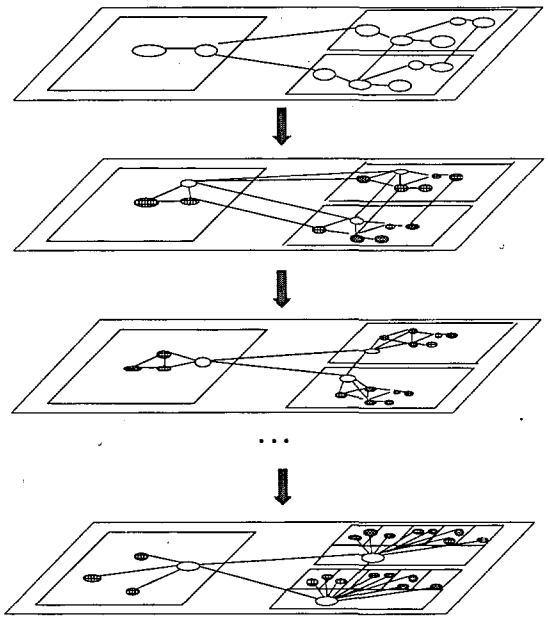


Figure 4: Structure of Stratified Architecture

In general, a complete stratified architecture is comprised of multiple strata as illustrated in Figure 4. Each stratum represents a refinement of the stratum above, and thus typically contains additional components and interactions that explain how interactions in the higher strata are realized. In Figure 4 the new components in a stratum are represented by white ovals, while those projected down from the stratum above are solid. The abstraction levels in a stratified architecture range from a high level, analysis-like description of the system down to the most detailed implementation-oriented view. The highest level will be most useful for understanding the overall solution strategy. This stratum is populated with components and connector types that are key to understanding how the system is organized. In the lowest stratum one will find the usual complex web of objects where it is hard to "see the forest for the tree". However, this is often the only appropriate level of abstraction to resolve realization details.

The key to effective stratification is the selection of appropriate sets of interactions within one stratum whose collective refinement elaborates upon a well-defined aspect of the system's realization. For instance, if in a refinement step starting from a business logic stratum one elaborates upon how remote interactions are realized, this will give rise to a "communication stratum". The next level down could be devoted to dealing with persistence issues and so on. Depending on the application domain the same sequence of types of strata will be useful in structuring a system's architecture according to its emergent (often non-functional) properties.

It is important to realize that each stratum of a stratified architecture is complete in its own right with respect to the level of abstraction it addresses. In contrast to layers, which only encapsulate a certain subset of a system, each stratum describes the entire system, albeit with varying degrees of abstractness. Clearly, all levels of abstraction are useful in a certain context. Someone trying to understand the overall structure of the system is best served with a high level view. Another person, whose task it may be to change the way data is marshalled over a network, gains more from looking at the communication stratum. Each stratum, therefore, provides the appropriate level of abstraction for a specialist working on a particular system property.

The multi-level separation of concerns provided by a stratification approach is even more valuable when it is used to organize a framework, since it offers enhanced opportunities for parameterisation. A fundamental concept in framework technology is the idea of so called "hot spots", i.e., points of adaptability. With a conventional, "single abstraction level" approach, all hot spots reside at the same level, and it is not clear what their individual role is within the complexity of the detailed architecture. In contrast, a stratified framework distributes hot spots over the various strata according to their place in the abstraction hierarchy. Hence, it is much easier to understand the role that a point of variability plays in the overall framework organization and what are the implications of plugging in a certain new behaviour.

Not only are the hot spots easier to see and understand in a stratified framework, but they can also have a greater range of effects on the eventual system's functionality. In conventional object-oriented frameworks, the hot spots are typically hooks to replace one *object* with another. This means that although the application developer is free to adjust the objects in the architecture, the interaction mechanisms, in contrast, remain relatively fixed. However, interactions, which also have a critical bearing on the system's functionality, are equally as likely to change as objects if not more so[1]. This is especially true when a framework has to be evolved to fit an extended or slightly different application context. By expressing interactions as objects in a lower stratum, a stratified framework can offer more flexible parameterisation than traditional object-oriented frameworks. Interactions within the framework can be made parameterisable by providing points of variation in their realization stratum. This allows application engineers to influence interaction semantics, as well as object semantics, by providing the required adaptation objects tailored to the precise needs of the customer.

2.3 Component Metamorphosis

As well as introducing new kinds of components and interactions, a given architectural stratum (except the top level) also contains projections of the components in the stratum above. The precise nature of the projection depends on the nature of the interaction refinement

[1] In fact, the success of object-oriented architectures partly lies in the fact that objects are more stable architectural abstractions than functions (i.e. interactions)

defining the relationship between the strata. Sometimes a component at one level will appear in the lower level completely unchanged. Often, however, the implementation or even interface of a component will be changed, either due to a change of its interaction partners, or a change in the nature of the interactions. For example, Figure 3 shows how component X is changed to X' because of the refinement of its interaction with Y. In the lower stratum the new version X', only communicates directly with A rather than with Y.

Because of the obvious analogy with the biological development of insects, we refer to the set of changes applied to a given component as it is projected across the different strata as *metamorphosis*. When viewed from the perspective of an individual system component, stratification can be understood as the successive metamorphosis of a component to its most detailed form in the most detailed architecture.

Consider, for example, the elaboration of an aspect of a system, such as authorization, within a given stratum. Authorization cannot be handled by a single module alone, but is a cross-cutting concern, i.e., it cuts through component boundaries and its introduction causes changes that are scattered in a non-local manner throughout the system. In a stratified architecture, however, the strata above the aspect-elaborating stratum do not deal with the aspect in an explicit manner, but defer matters of authorization to the stratum that explicitly elaborates upon the realization of this aspect (i.e. the authorization stratum). When the authorization interactions from this stratum are refined in lower level strata, the spreading of the aspect-related details starts until the bottom level is reached (see section 3.1 for a sample application).

The need for changes to components as they are projected into lower strata is simply a testimony to the fact that components typically cannot be used "out of the box" and that a pure black-box "plug & play" composition strategy rarely works in practice. To make component composition really work, components have to offer some open implementation facilities [5] and connectors have to be grey-box connectors [6].

3 Example stratified Framework

To illustrate how the principles of stratification explained in the previous section would be applied in practice, in this section we walk through a small example. The subject of the example is a simple banking system which stores and accesses accounts on behalf of customers. Key requirements for this facility are that the bank and the accounts are potentially remote, access to accounts must only be granted to authorized users, and interactions between the bank and accounts must be secure (i.e. non-interceptable).

In the discussion below we focus on only a small part of the systems potential functionality, but the ideas explained can be scaled up easily to the other parts of the systems.

3.1 Structure of a Stratified Framework

Naturally the top-level stratum is the simplest, since it describes the structure of the system using the semantically richest connectors. The class diagram in Figure 5 shows that the class `Bank` interacts with the class `Account` by a means of an interaction labelled with the stereotype `«remoteSafe»`. This conveys the fact that the interaction is a semantically rich connector and "wraps up" the requirements for remoteness, authorization and security identified above.

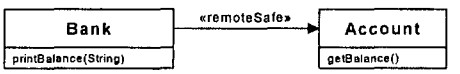


Figure 5: Application Stratum

The next stratum elaborates upon the realization of the authorization aspects of the system's functionality, as illustrated in Figure 6.

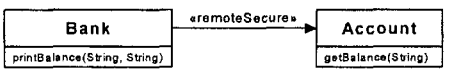


Figure 6: Authorization Stratum

In principle, this is done by defining the additional components and interactions that are involved in the authorization process, and by defining how the components and interactions from the Application stratum are changed. In this case, no new components are required but the interaction between the `Bank` and the `Account` components needs to carry the information needed to perform the authorization. In particular, methods `getBalance()` and `printBalance()` now are elaborated to feature one more parameter. A so called PIN (personal identification number) has to be provided when requesting a service from the bank and is checked by the account before access is granted. Note, how the annotation of the association has changed from `remoteSafe` to `remoteSecure`, as the safety aspect of the interaction is addressed. It remains to specify that the interaction in addition is remote and secure.

The next stratum elaborates upon the distribution aspect of the system. As illustrated in Figure 7, this uses the broker pattern as defined by Buschmann et al. [7] to realize distribution.

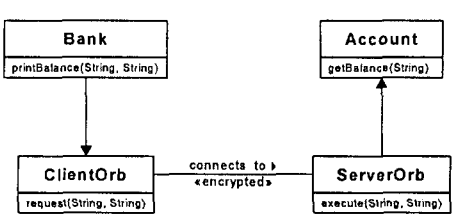


Figure 7: Distribution Stratum

In this particular refinement, two new component types and one new interaction type are introduced. Note that this implies a change to the required interface of the `Bank` since it must now interact with its `ClientOrb` rather than with the `Account` directly. The `Account` component type, on the other hand, is totally unaffected by this refinement.

The final stratum in this example elaborates upon the realization of encryption. As illustrated in Figure 8, this is achieved by the introduction of an additional `Encryption` component type that is used by the `ClientOrb` and `ServerOrb` components to respectively encode and decode message before they cross the network.

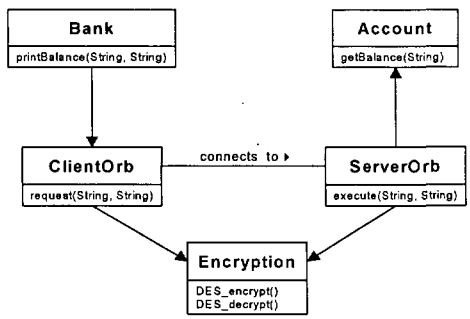


Figure 8: Encryption Stratum

Further strata would be defined to elaborate upon successively lower level details of the system's realization until a level is reached that can be translated directly into an executable form. In general, strata are defined to focus on the realization of each of the key aspects of the system realization. Since the aspects affecting a system tend to be domain specific so are the strata.

3.2 Refinement transformations

Documenting architecture at multiple levels of abstractions as shown in the previous section is a useful activity in its own right. However, it is unrealistic to expect that all the strata can be made mutually consistent manually.

Today it is rare to find examples of software development approaches in which just the code and a single, very high level architectural description are kept mutually consistent, so it is not feasible to expect this to be achievable when there are even more system representations. Therefore, a stratification tool is needed that can organize and help apply -

1. an annotation language and
2. a set of refinement transformations,

both of which are typically domain dependent. In the example used above we exploited the fact that one could use annotations such as `encrypted` or `remoteSecure` and that there are relations such as `remoteSafe = authorized + remoteSecure`. The task of the annotation language is thus to provide labels that are used to refer to certain interaction semantics and relations between them.

Each atomic label of the annotation language refers to an individual refinement transformation. Such a transformation takes one component scenario involving a labelled interaction and creates a new scenario in the stratum below. The resulting scenario has less rich interactions and possibly additional components implementing the interaction semantics of the initial interaction (see Figure 3). A tool that fully supports the stratification approach therefore needs to –

- manage the annotation language.
- support the definition of refinement translations.
- apply the refinement translations, thus keeping the strata in sync with each other.
- facilitate the reengineering of detailed scenarios to high-level interactions by applying refinement translations in the reverse direction.

In particular, for the purpose of discovering and documenting the architecture within already existing software, the last point is useful in creating high-level views from representations with too much detail. Naturally, in creating a stratified architecture one will *use* and *define* both the annotation language and refinement translations in an interleaved fashion. Both sthese assets, although typically domain dependent, are reusable for other systems in a similar domain.

4 Developing, Using and Evolving Stratified Frameworks

The previous sections have respectively defined and exemplified the structure of a stratified architecture. In doing so they have essentially focused on the *product* of stratification, but have said little about how stratified architectures are created, used and evolved, that is, the *processes* related to stratification. In fact, just as a modeling language such as the UML can be understood and used without an associated process, the concept of architecture stratification is useful as an organizational principle on its own. It is, therefore, useable with a variety of methodological approaches. In this section we explain the most important methodological issues associated with stratified frameworks, and suggest some solutions.

A framework oriented approach to software development raises three main questions –

- how should the framework initially be developed ?
- what are the necessary actions to instantiate the framework so that applications meet the needs of a specific customer?
- how should the framework be evolved over time in response to change requests?

4.1 Developing Stratified Frameworks

In addition to the usual framework development activities the following five main activities have to be

incorporated into a process for creating and using stratified frameworks-

1. Identify candidate strata
2. Order candidate strata
3. Identify hot-spots and allocate them to strata
4. Elaborate strata
5. Partial Implementation

4.1.1 Identify Strata

The first challenge in the development of a stratified framework is the identification of the appropriate strata. This question must be tackled from both a top down and bottom up perspective. From the top down perspective, the key aspects of the system as identified in the requirements specification are extracted and analysed. Each system aspect that is likely to involve numerous objects is an indicator of a possible stratum. But it is also possible that an aspect does not introduce objects at all but only changes method signatures and implementations (e.g. the "authorization" aspect in the example in section 3.1). From the bottom up perspective, the key reusable components within standard or predefined architectural solutions (e.g. CORBA) are evaluated and considered for the problem in hand. By carefully balancing the needs of the application against the potential solutions an optimal set of strata is gradually distilled. In general, a strata-inducing aspect could be anything that is not essential to a very high level analysis model of the system. A good indication for proper aspects is the one would expect them to appear in similar systems within the system's domain.

4.1.2 Order the Strata

The next step is to order the strata according to their dependencies and levels of abstraction. A key tenet of the stratification approach is that strata can be arranged in a strict order, such that a stratum on one level depends on (is refined to) only the stratum immediately below. Often, however, analysis of the initial candidate list of strata reveals mutual dependencies, or strata that are dependent on two or more other strata. Such situations must be rectified either by removing strata from the list, or by splitting strata into more specialized strata. At the end of this process, a list of strata must be identified in which each stratum only depends on (is refined to) the stratum immediately below.

4.1.3 Identify and Allocate Hotspots

The next step is to analyse the intended future uses of the system with a view to the identification of the key points of variation, or "hot spots". This activity is akin to the scoping and variability analysis activities found in product line approaches. Once candidate hot spots have been identified, a first pass is made at allocating hot spots to the identified strata. This is driven by a consideration of the aspects of the system that are affected by particular hot spots. One straightforward technique for accomplishing this is to consider one hot spot at a time and proceed down the strata from top to bottom until the hot spot's purpose is given meaning by the corresponding stratum. Strata above this stratum suppress the level of detail that the hot spot is addressing and strata below are likely to contain abstractions which support the realization of hot spot components. A concrete definition of the hot spot cannot be completed until the architecture of the stratum has been elaborated, but an initial allocation of hot-spots to strata can be made based on the purpose of each hot spot.

4.1.4 Strata Elaboration

The most challenging activity is the elaboration of the individual strata. Apart from the top-stratum, the process is basically the same for each. The process of creating the top-level stratum is a little different, since it has much in common with the typical process of deriving an initial high-level architectural in a traditional object-oriented development method. The main difference is that care must be taken to keep the top stratum as abstract as possible, since the architectural features derived from specific aspects of the system should be elaborated in lower-level strata. This differs from traditional development methods where all aspects of the system are usually considered in the (single) architectural description.

The main difference between normal architecture development, and the development of the top-level stratum in a stratified framework is that the latter is allowed to make assumptions about services elaborated by lower-level strata. For example, in a stratified framework containing a stratum that realizes remote communication (i.e. in the style of an ORB), a remote interaction mechanism can be assumed as a basic architectural primitive (i.e. connector) by any stratum above it (including of course the top level). This is not so in regular architecture development, wherever functional requirement must be considered within the single overall architecture.

Once a first version of the top-level stratum has been defined, the next step is identifying which of the interactions at the top level are affected by the aspect handled by the stratum below. These interactions are then analysed in turn to identify appropriate refinements based on the facilities handled by the underlying strata. This can be done using the transformations identified above. The set of candidate refinement steps are then analysed and a common solution is distilled. A complete architecture for the underlying stratum is then determined, taking into account relevant architectural patterns or lower level design patterns depending on the current level of abstraction. Note that the realization of a particular stratum as well as the definition of the refinement steps mapping into the stratum lends itself to enactment by someone who is an expert in the area the stratum addresses. For instance, experts in distribution, fault-tolerance, persistence, encryption, etc. will work on their respective strata, potentially in parallel to some extent.

The previous activity is repeated until each stratum in the framework is elaborated and candidate architectures have

been selected. Finally, once the detailed architectures are available, the hot spots can be revisited and accommodated in the framework using one of the standard object-oriented techniques, as explained below.
One of the things that must be considered at each stratum is the concrete representation of the various hotspots allocated to that level. Various techniques can be used to capture hot-spots (for instance, see [8]), but for our purposes here it is sufficient to distinguish between so called "white-box" and "black-box" specialization [9]. A mature framework, which has been instantiated a number of times and has already undergone all major refactorings, will offer mostly if not exclusively black-box specialization. Here the hot spot has a well defined interface and is parameterised by plugging in a prefabricated component (e.g. by using pluggable adaptors or the Function Object pattern [10]). If a prefabricated component (packaged as part of the framework) is used one of the known framework behaviours is selected. If a new component has to be created than the framework will still behave within the range of possibilities opened up by the black-box hot spot. Analysing were that hot spot is located, i.e., which stratum it populates and which higher-level interactions depend on it gives a good picture about the scope of the variation introduced. Here the term traceability is given meaning, since it is possible to travel the refinement steps up and down to trace where the variation is being propagated. This advantage of the stratification approach becomes even more important when the other form of parameterisation, i.e., white-box specialization is used. In this case, a framework class is subclassed to override one or more methods. This form of framework adaptation is common when the framework is still evolving to its final form and it is not yet clear how the black-box hot spots should be modelled and where they should be located. Not only does this specialization variant require more intimate knowledge of the framework but it is also potentially much more dangerous, as erroneous overriding of behaviour could have disastrous effects on the integrity of the framework behaviour. Therefore, if white-box specialization has to be used it is doubly important and rewarding to travel up and down the strata, following the refinement transformations, to check the scope of the changes made.

4.1.5 Partial Implementation

The lowest-level stratum of a stratified framework represents the realization of the system containing all of the details introduced in the strata above. This is the stratum which is used to derive the implementation of the system. Depending on the level of detail present, i.e., on the amount of low level design aspects introduced by one or more strata above, the derivation of an implementation ranges from a standard "design-to-implementation" activity to a simple "one-to-one" mapping into a programming language.

4.2 Using the framework

Using a stratified framework to create a concrete application involves three main activities:

1. Variant Resolution
2. Hot-spot Instantiation
3. Implementation Completion

4.2.1 Variant Resolution

Variant resolution is essentially the same for all framework-based approaches to software development. The basic goal is to determine precisely what set of features the desired variant of the system requires in terms of the hot spots made available by the framework. Those features that correspond to default options, if these are available, can be immediately dealt with and should be documented with interaction annotations so that they can be validated in case the default option changes in the future.

4.2.2 Hot-Spot Instantiation

The next step is to actually resolve the non-default choices by providing concrete representations of the chosen features for each of the affected hot-spots. The advantage of stratification is that the hot-spots are clearly separated and defined with respect to the aspect of the system that they affect. The instantiation of hot-spots in a stratified framework is therefore much easier than in other non-stratified frameworks. First, it is clearer where to look for an appropriate hot spot because of their distribution to strata. Second, as detailed above, the refinement transformations represent paths to the places within the framework affected by the hot-spot allocation.
Since hot-spots are defined at different strata, the effects of the allocation of a hot spot must be carried down through the strata to the lowest stratum. This involves the reapplication of the refinement transformations where appropriate. As indicated beforehand, this is where stratification can benefit most from appropriate tool support.

4.2.3 Implementation Completion

Once the realizations of all non-default hot-spot resolutions have been mapped down to the bottom stratum, any modified or completed architectural elements must be translated to an executable form using the same techniques specified previously.

4.3 Evolving the framework

The final major activity involving in building and applying stratified frameworks is their evolution. No software system, or family of systems, is constant, so some systematic procedure is required for handling the inevitable requests for change. In the context of a stratified framework this involves three main activities:

1. Change analysis
2. Change Localization
3. Realization

4.3.1 Change analysis

Most requests for changes are received from users and customers of specific version of the system. Thus, before changing the framework, it is first necessary to determine whether the request can actually be handled by a reinstantiation of the framework with a modified set of parameters. If this is so, the existing framework is capable of handling the request and no framework level modification is necessary.

4.3.2 Change Localization

For those changes that are deemed appropriate for the framework, it is then necessary to establish two things. First, is the change "almost" covered by an existing hot-spot? If so, the hot-spot realization must be generalized to handle the change. If not, then it is necessary to repeat the "Identify hot-spots and allocate them to strata" activity in framework development, and adjust the framework accordingly to handle the requested change.

4.3.3 Realization

In both cases it is necessary to bring the whole framework, including the associated implementation, into a consistent state by reapplying all relevant refinement transformations. This ensures that all changes are reflected in lower strata and the executable code. Again, tracing out the affect of changes within higher strata and following them along the refinement transformations to all critical locations within the framework gives an extra means of validating the new framework version in addition to regression tests of the framework itself and its known instantiations.

5 Related Work

The concept of stratified frameworks is related to several other ideas which are currently under investigation in the software engineering community. In this section we briefly describe the most important of these.

5.1 Aspect-oriented programming

Aspect-oriented programming has received a great deal of attention recently as a way of separating out, and partially automating, the treatment of orthogonal aspects of a systems overall functionality [11]. The stratification approach shares the same basic philosophy of aspect oriented programming since it is built on the basic idea of separating concerns for orthogonal issues [12]. As explained above each stratum basically focuses on the realization of a specific aspect of as system.

Most aspect oriented approaches only deal with the code level of a system whereas stratification is an architectural approach as well. The main difference between stratification and the primary "aspect-oriented programming" approaches, however, is the strategy adopted for the description and manipulation of these aspects. In aspect oriented programming, the emphasis is on the semi-automatic integration of distinct aspects using special tools typically known as weavers. This is fine for a number of aspects that can be handled well by weavers, but in principle this approach suffers from a hard superimposition problem when independent aspects affect the same software abstraction. It is, for instance, a non trivial problem to decide the order of the modifications to be made in such a case. Many of the "concerns" which need to be addressed in large-scale industrial software development are too complex to be handled in this way. Sometimes *all* the details of a realization need to be available in order to find a bug or achieve the desired behaviour. In this case a pure version of the system plus a number of aspects and an implicit weaving strategy is insufficient. The stratification approach essential offers an alternative way of separating aspects using a more manual software engineering based approach (based on standard software development activities). The order of aspect introduction is fixed. Each aspect is defined on a complete system description (with regard to the required level of abstraction) and not on a pure, i.e., incomplete, system and possibly more aspects. While stratification successfully resolves superimposition conflicts with a linear and dependent aspect introduction, the downside of this is that it is not possible to view a system's architecture with an aspect B but without an aspect A if A is introduced earlier (higher up the hierarchy) than B. With an optimal ordering of the strata, however, it is unlikely that such a need will occur.

5.2 Reflective Architectures

Another community of researchers, which has pursued the idea of capturing distinct aspects of a system's behaviour within separate "architectures" is the reflective programming community. The basic premise of this community is that the description of a system should comprise two "architectures", one representing the standard application software, and the other "meta" architecture describing certain aspects of the support software that are amenable to change, sometimes even during execution of the software. Since the meta architecture allows aspects of the system to be modified (at run-time), which were previously considered fixed primitives of software construction in traditional approaches, the whole approach is known as a reflective architecture. Since is separates out the description of different aspects of the systems behaviour into separate "architectures" this approach has much in common with stratification. In fact, we view stratification as a generalization of the reflective architecture approach - a generalization which supports an arbitrary number of "architectures" rather than just two. Interestingly, in the case of a reflective programming language, such as CLOS, the reflective "stratum" must be viewed as being below the programming language stratum, as it elaborates and defines the meaning of programming language mechanisms (called interactions in systems). By the same token a meta stratum will be found *below* strata which depend on it, although the general interpretation of "meta" being higher and above something.

6 Conclusion

As the complexity of software systems continues to grow, and the boundary between applications and systems level software continues to blur, it will become increasingly difficult to visualize the structure of software systems using only one structural model. If all the "components and connectors" for all aspects of the system's functionality are crammed into a single structural model, it will become increasingly impossible to "see the forest for the tree." In this paper we have described a solution to this problem referred to as architecture stratification.

The basic premise of the approach is that several distinct models of a system's structure should be developed, each yielding a different level of abstraction, and focussing on the elaboration of different aspect of the system's functionality. As such, the approach can be viewed as extending the tenets of aspect-oriented programming to higher-level development concerns, but in a more software engineering oriented style.

The idea of creating a series of models of a system's structure, related by rigorous interaction refinement transformations, has value as a systematic technique for realizing the full, all encompassing structural model ready for implementation. When used in this way, the higher-level strata essentially represent stepping-stones, or milestones, along the route to towards the development of the "real" or ultimate all-encompassing architecture. The higher-level strata thus play a secondary role, and are likely to be neglected over time.

Greater leverage can be gained from the architectural stratification approach, however, when it is used to describe the structure of a component-based framework. When used for the purpose, the higher-level strata have the same weight as the lowest level stratum, and play a valuable role in describing the framework structure from the perspective of a particular stakeholder. The main value of the distinct strata is to provide a clean way of capturing functional (i.e. interaction-based) variation points in terms of components (albeit in a lower stratum), and to provide a clearer model of the effects of parameterised "hot-spots".

Another major benefit of the separation of concerns afforded by stratification is that is clarifies the role of proven architectural patterns in the structuring of a software system. The majority of architectural and design patterns that have been published to date are intended to be used together, but when a single, all-encompassing architectural model is used to describe the structure of a system, the application and interrelationship of specific patterns is all but lost. Stratification helps by providing distinct abstraction levels that focus on the deployment of only one or a few patterns, and thus the role and location of specific patterns, as well as their relationship to other patterns at different strata, can easily be discerned.

As well as explaining the basic principles of stratification, and illustrating their application in the context of a small example, this paper provided an outline of the primary development activities associated with the approach. The paper also discussed the relationship of the stratification concept with other leading architectural research initiatives. As a generalization of the aspect-oriented programming and reflective architecture approaches to software development, the principle of architecture stratification represents the next step along the road towards greater separation of concerns in the engineering of quality of software systems.

7 References

[1] M. Shaw, and D. Garlan (1996) Software Architecture: Perspectives on an Emerging Discipline, Prentice-Hall

[2] C. Szyperski (1998) Component Software, Addison-Wesley

[3] Nicholas Wirth (1971) "Program Development by Stepwise Refinement." Communications of the ACM, vol. 14, no. 4, pp. 221—227.

[4] M. Broy (1997) Towards a Mathematical Concept of a Component and Its Use, TUM Report I9746.

[5] G. Kiczales (1996) Beyond the Black Box: Open Implementations, IEEE Software, vol. 13. no. 1, pages 8—11.

[6] U. Assmann and A. Ludwig (1999) Introducing Connections into Classes with Static Metaprogramming, 3rd Int. Conf. on Coordination, LNCS 1594.

[7] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal (1996) Pattern-Oriented Software Architecture – A System of Patterns, John Wiley & Sons.

[8] W. Pree (1994) Meta Patterns – A Means for Capturing the Essentials of Reusable Object-Oriented Design, ECOOP '94, pages 139—149.

[9] R. E. Johnson and B. Foote (1988) Designing Reusable Classes, Journal of Object-Oriented Programming, vol. 1, no. 2, pages 22—35.

[10] T. Kühne (1997) The Function Object Pattern, C++ Report, vol. 9, no. 9, pages 32—42.

[11] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, J.-M. Loingtier and J. Irwin (1997) Aspect-Oriented Programming, *In proceedings of the European Conference on Object-Oriented Programming*, Finland. Springer-Verlag LNCS 1241.

[12] C. Atkinson and T. Kühne (2000) Separation of Concerns through Stratified Architectures, International Workshop on Aspects & Dimensions of Concerns, ECOOP 2000, Cannes, France.

Learning and understanding human skill

Tanja Urbančič [1,2]
[1]Jožef Stefan Institute, Department of Intelligent Systems, and Center for Knowledge Transfer in Information Technologies, Jamova 39, SI-1000 Ljubljana, Slovenia
e-mail: tanja.urbancic@ijs.si
[2]Nova Gorica Polytechnic, School of Engineering and Management, Vipavska 13, p.p.301, SI-5001 Nova Gorica, Slovenia

Keywords: human learning, modelling of human skill, machine learning, transfer of human skill, dynamic systems control

Received: November 15, 2001

Research results in machine learning have confirmed the possibility of reconstruction of human skill, resulting in similar or even improved performance of the same skill by a computer program. The methods that build symbolic models of skills are preferred if generated models are to be used to help humans in improving their own skill. The paper discusses these possibilities with the objective of accelerating and enhancing the process of human skill development. Experiments in learning control of dynamic systems are presented and wider potential of applications in assisting learning of skill is discussed.

1. Introduction

In many areas, professionals solve problems and perform tasks using highly specialized skills, acquired and accumulated through the years of experience. Examples can be found in almost every profession, such as piloting aircrafts, deciding in business dilemmas, diagnosing patients, operating cranes, etc. Development of computer science and informatics, especially in the field of intelligent systems, offers support to expert problem solving in many aspects, providing help in unusual cases, in achieving required reliability, in optimizing resources and meeting other specified success criteria. Still, learning human skill, including inevitable learning by doing (and making mistakes), remains a time-consuming and in some cases a very expensive process where appropriate computer applications could contribute to great improvements.

It has been shown experimentally that people have problems when trying to describe their performance of a task that requires skills. Their descriptions are not only incomplete and inaccurate, they can even differ from the facts that can be revealed from the recordings of their performance (Urbančič and Bratko, 1994a). This can be explained by the fact that skills become more or less subconscious when they develop to a satisfactory level of performance. Consequently, also teaching skills is difficult. It is not always clear what are the essential elements of skill that make somebody an outstanding performer, and trying to pass this knowledge to a learner and expecting him or her to apply it in their own performance is even harder. The phenomenon relates to the experience of brothers Wright, observing birds when looking for a solution for a technical detail in constructing their airplane a hundred years ago. In a book by Ferguson (1993), their experience is described as reported by Orvile Write: "Learning the secret of flight from a bird was a good deal like learning the secret of magic from a magician. After you once know the trick and know what to look for, you see things that you did not see when you did not know exactly what to look for."

To help learners and teachers involved in learning of skills, modeling of human skills could play an important role in the future. It is important that such a method captures current strategy at the conceptual level as well as important details critical for successful task performance. It must also allow for comparing models of teacher's and learner's skill, as well as the models of the learner's skill in different learning stages. It is easier to use such a model in the learning process if it is written in a formalism close to the human understanding of the problem. Therefore, in this context, we are not interested in subsymbolic learning methods such as neural networks, although it is known that in many cases, they can successfully model human skill.

Several studies have shown that symbolic machine learning methods can reconstruct human skill, generating a transparent model of human skill from traces of human performance (Michie et al. 1990, Sammut et al. 1992, Urbančič and Bratko 1994a). In all mentioned cases, the modeled skill was manual control of dynamic systems. Resulting decision trees can mimic the performance, recorded in traces that served as learning examples for the computer program. A detailed presentation of these studies with the emphasis on their limitations was published in

(Bratko and Urbančič 1997). An essential improvement of the approach was introduced by Šuc and Bratko (2000). In their work, Šuc and Bratko greatly enhance the relevance of the learned models in terms of human conceptualization by introducing goals and subgoals as typical elements of human problem-solving.

In Section 2, different types of knowledge are discussed and distinguishing characteristics of skill as a special type of knowledge are described. Section 3 presents experimental results in human learning of manual skill in dynamic system control and gives characteristics of this process in two series of experiments: one with a computer simulator and one with a physical model of a container crane. Findings of these experiments are used in Section 4 that discusses wider potential of modeling of human skill in helping humans to learn skills better and faster.

2. Types of knowledge

In everyday speaking, we use the word "knowledge" in very different contexts, without being aware of its multiple meanings. Different layers of knowledge become visible when it is applied in a repetitive mode in a well-known environment, in a changed environment, or in completely new circumstances, connected to other pieces of knowledge and leading to a new understanding of a whole or even to discovering new knowledge.

Classical science and philosophy have been discovering and studying propositional knowledge, which can be expressed and described in a propositional form, i.e. in sentences. Classical definition of propositional knowledge, introduced already by Platon, consists of three necessary and, when being together, sufficient conditions: it should be grounded, truthful and convincing. Since science functions at an intersubjective level, it should be grounded on basic facts, recognized by a scientific community as foundations of a certain field.

Although in science and education we do not cover only systematic and methodic explanations of different phenomena, this remains a basic goal that enables also achieving other goals, like solving problems in natural, social and technical areas. Answers to the "What" and "How" questions are important as well, although they are not necessarily systematically connected to the global tresury of scientifically accepted knowledge. In this sense, Ryle (1978) distinguishes between "knowing *that*" and "knowing *how*", being convinced that not all knowledge can be represented as propositional knowledge.

A typical example of "knowing *how*" is any example of a skill that somebody has developed through a long lasting period of performance, including learning that typically involves at least some trial and error process, and results in a more or less subconcious performance of the task. Such skills can be found in everyday life, like riding a bycicle or driving a car, or as parts of highly specialized professional tasks in various fields, ranging from operating cranes to piloting aircrafts or diagnosing in medicine. When asking a well-skilled performer for explanation, how does he or she actually perform a specific task which involves skills, they have difficulties in finding words for a description. They are typically not aware of all of their reactions, and even if they are, they can hardly give satisfactory details and explanations for them. It is much easier for them to manifest their skill by performance itself. As described in (Michie 1986), "know *how*" could be seen as a recognizing motto of skills, while "show *how*" could be used for their external manifestation.

Through a repetitive performance of a certain task, people develop a way of performing that has proven to be successful to a satisfactory extent, and they change it only if they are forced to do so due to changed circumstances or requirements. If we can describe such a way of performing in a reasonable way, we can talk about the *strategy*, which enables also better insight into the skill. This contributes to at least some exchangeability, giving us the possibility of talking about the strategy, discussing it with other people, and finally, using it in an educational process to help teachers and learners to make learning of skills more efficient.

3. Learning of skills

At the Jožef Stefan Institute, several series of experiments in learning skills were carried out in the domain of dynamic systems control. They cover

- experiments in which a computer program learned to control a dynamic system by a trial-and-error (Urbančič and Bratko, 1994b; Filipič et al., 1999), without a priori knowledge or with partial knowledge about the domain,
- experiments in which volunteers developed their human skill which was then used to produce learning examples for machine learning programs (Urbančič, 1994; Urbančič et al., 1998).

Results of exhaustive experimentation with machine learning, described in the publications mentioned in the previous paragraph, have confirmed that

- a computer program can learn to control a dynamic system without prior knowledge,
- a table of state/action rules as an encoded control rule can be compressed into a comprehensible control rule,
- a compressed comprehensible rule can be optimized, keeping all the qualitative characteristics of the strategy, but improving performance by setting quantitative details differently,

- qualitative knowledge about the domain can be used to shorten the process of learning, replacing the phase of learning without prior knowledge.
- traces of human performance can be used to shorten the process of learning, replacing the phase of learning from scratch.

Figure 1 shows how the mentioned modes of learning can complement and upgrade each other.

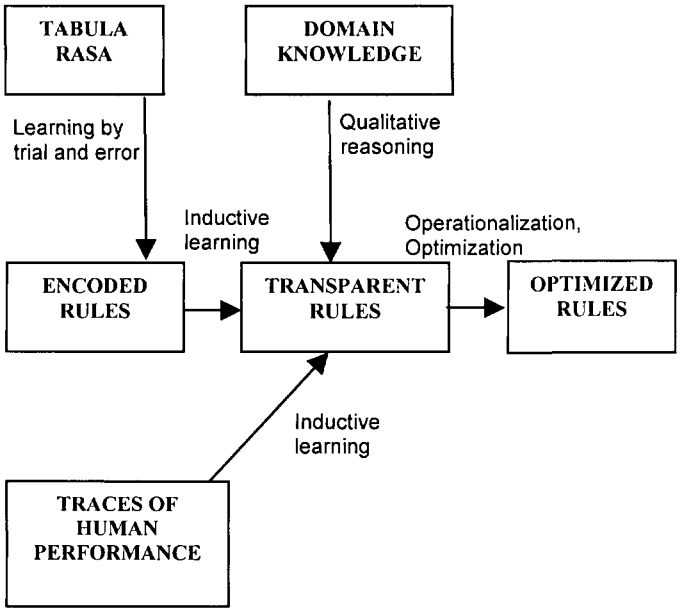


Figure 1: Different types of knowledge and learning can be used in learning skills and building their models.

We are further interested if the same repository of methods and techniques can help humans in learning their own skill. With this question in mind, in this paper we explore on the experiments where the emphasis is on observing human unsupervised learning of skill by trial and error and on the influence of transparent rules in form of verbal advice on this process. Knowing that a model of human task performance can be learned automatically from traces of performance, the idea is to use them together with other techniques (e.g. visualization, discussion) to help learners to develop their skill better and faster. In this paper we concentrate on the problem of transferability by studying to which extent and under what conditions a model of a skillful performer can help other learners that are still in the process of developing the same skill.

3.1. Learning to control a computer simulator of a dynamic system

Six volunteers participated in an experiment where they were asked to learn to control the crane simulator. They were given just the "instrument" version without any information about the system they controlled. They even didn't know it was a crane. The precise definition of the control task was as follows. By striking the keys $\leftarrow$, $\rightarrow$, $\uparrow$ and $\downarrow$, bring the height of the six columns, representing state variables, within a narrow target region marked on each column. By striking the control keys, control variables presented in two additional columns change their value. The system is supposed to be stabilized at the target position, when all the six state variables stay within the target region for at least 8 seconds. The task is to stabilize the system as fast as possible. A trial is unsuccessful, if any of the six variables falls out of the specified boundaries (also marked) as well as if after three minutes the system is not stabilized within the target region.

Each subject was asked to perform 200 trials with the simulator. This amounted to roughly 8 to 10 hours of training for each subject. In order to develop their own control strategies, they were unsupervised and were not allow to observe other subjects' learning during this phase of the experiment. The whole learning process of the subjects was recorded, including the very first, unsuccessful trials.

After 200 performed trials, all the six subjects succeeded to accomplish the task they were given. While one of them still felt very uncomfortable and unsure, the others believed that they knew adequate control strategies. However, remarkable individual differences were observed regarding the speed of controlling and the frequency of successful experiments and the characteristics of the strategy used. Although observing when successful trials appear gives some basic information about the learning process, this is not sufficient to help subjects or their teacher. For example, they can warn that somebody is currently not making a progress, but they don't tell us why. Is the learner just not sufficiently attentive or concentrated or is it due to the fact that the learner builds an incorrect strategy? If yes, which elements of his or her current understanding should be corrected? Maybe the learner is building a promising strategy (maybe even a novel one, not yet known to the teacher), but needs more time to develop it? The problem is even harder in the domains where tasks can be performed in different ways. For example, in controlling cranes, some subjects tend towards fast and less reliable operation, while others were slower, more conservative, and more reliable. Some are avoiding large angular accelerations at the expense of time. Such strategies produce reliable, but slow performance. This is in contrast with some subjects' strategies that tend to achieve better times, but require higher accelerations whitch causes large angles and requires very delicate balancing at the end of the trail. To which extent individual characteristics could be respected and when one should try to change them?

To tackle this kind of questions, we need a better insight into a strategy that underlies a subject's current performance and represents his or her current understanding of the problem. To get this insight, we

invited the subjects to a discussion where they were expected to compare their strategies and get some information that could help them in further improvements of their performance. After the discussion, subjects were given additional 50 trials each, in order to see how the discussion and other subjects' instructions affected their control performance.

It turned out that, although being incomplete and vague, the instructions formulated during the discussion provided useful guidance for some subjects. During the additional 50 trials, 2 volunteers significantly improved their performance. E.g., while in the first series of trials subject A had the best result of 75.64 seconds, after the discussion he achieved 56.84 seconds. Also his average improved, from 136.77 seconds to 120.20 seconds. It is interesting that his strategy on qualitative level remained the same, and he explicitly reported that his improvement was due to the one single numerical detail he didn't know before. On the other hand, subject B who didn't develop a consistant strategy during the first 200 trials, didn't improve at all. Even more, the subject reported about a confusion when trying to mix her way of performing with somebody else's instructions. This highlights the limited value of general verbal advice and confirms the importance of knowing which piece of advice is to be given to a particular subject at his/her specific stage of performance level. If it is given too early, the learner can not attach it to his or her current partial understanding. On the other hand, if it is given too late, the learner can already be very fixed in his or her way of performing the task which is therefore very difficult to be changed.

3.2. Learning to control a physical model of a crane

To study learning of control skill in a more realistic environment, in further experiments we used a physical model of a container crane. The functional part of the model consists of six sensors and two step motors. The motors are used to control the horizontal position of the trolley and the vertical position of the load. The inclination of the rope that carries the load is measured by an angle sensor mounted on the trolley. There are five other sensors, mounted on the construstion and used to detect the end positions of the track and the top and bottom positions of the load. The sensors and power electronics of the motors are interfaced with a computer program which is responsible for normal motor functioning, sensor data interpretation and control. The crane can be operated manually from the keyboard, as well as automatically by a computer program. Due to the importance of swing control, the study was focused on this part of the task. More precisely, the control task consisted of two subtasks: (1) increasing swinging from zero to a specified amplitude of 10 degrees, (2) damping swinging under a specified amplitude of 1 degree. Similarly to the objective in the experiments described in the previous section, the goal was to minimize the cumulative execution time.

Like in the experiments described in the previous section, also in this case a group of six volunteers learned to control the system by a series of unsupervised trials, with no communication among the subjects allowed. Each subject had 60 trials for the learning phase and 10 trials to exhibit the best possible performance. Basic characteristics of their learning process were seen from the graphs as the two given as an example in Figure 2. The dots represent time needed for each trial of a particular subject. Dots on the upper end of the diagrams represent unsuccessful trials while dots lying lower represent successful trials – the lower the better.

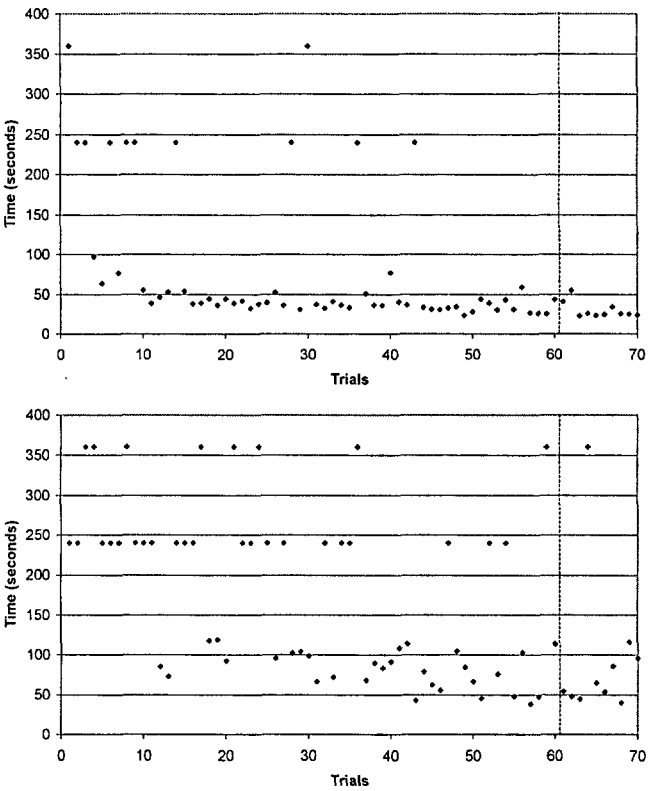


Figure 2: Basic characteristics of the learning process of two subjects, showing differences in learning speed, in number of unsuccessful trials and in time needed to accomplish the task in successful trials.

After the subjects had accomplished their task, the discussion with exchange of discovered features and advice took place, followed by additional 20 trials for each subject. Again, the goal was to observe the influence the discussion had on performans in the subsequent trials.

Based on the experience from the previous experiments where difficulties with wording of skill description revealed, in this case we used graphical presentation of qualitative characteristics of the skill (Figure 3) which was only in some details completed by quantitative information. Consequently, the process of describing and

comparing the most distinguishing features of subjects' skill was much faster and easier.

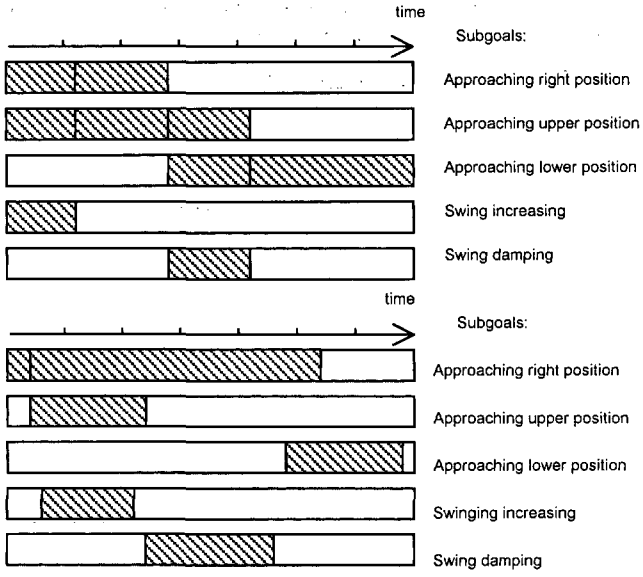


Figure 3: Basic qualitative features of control strategies developed by two subjects show differences in subgoal coordination.

Subjects reported on a point in the series of trials from which on "you know you will as a rule perform successfully". This is not necessarily the point when the first successful trial occurs. Differences in the speed of coming to this point were significant. Also differences in the subsequent improvement were big and not necessarily connected to the speed of progress in the initial phase of learning. In particular, a person that mastered the task relatively slowly, greatly improved in the continuation of the learning process and also ended up with a very reliable style of performing.

Experimental results reveal the same phenomenon as the ones with the numeric simulator, described in the previous section. While some of the subjects performed significantly better after the discussion, one improving the performance time by 13.5% and one by 20%, there was again a subject that could not improve at all. His comment was that he was obviously already too adjusted to the way of performance that he had developed.

4. Discussion

Presented experimental results could be briefly summarized by the fact that it is extremely important when to give a piece of advice to a certain learning subject, and which piece of advice should be given at a certain level of learner's understanding of the task. By intervening too early, learners can become confused since they have still not developed a model of the domain to which they could attach the told facts in a meaningful way. On the other hand, giving advice too late can result in necessarily big efforts needed first to "forget" already established control strategy that should be corrected, and then to develop a new one.

In any case, it is not sufficient to present the correct model of the skill to a learner, expecting him or her to adopt it and use it as his or her own. It is important to get some insight into the learner's current understanding and current stage of development of his or her own skill. Is it basically correct, but incomplete? Is it incorrect in some important features? What really counts is the comparison between the teacher's and learner's understanding. Different approaches that facilitate this understanding can be used, including discussion and simple visualization techniques. While the majority of the machine learning studies applied to the reconstruction of human skill concentrate of the problem of replicating the performance by a computer program, we believe that they have great potential as a modeling approach that could be used for enhancing the process of human learning in the domains where practice by trial and error is a necessary part of gaining needed experience.

Acknowledgements

The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport.

References

[1] Bratko I., Urbančič T. (1997): Transfer of control skill by machine learning. *Engineering Applications of Artificial Intelligence*, Vol.10, No.1, 63–71.

[2] Ferguson E.S. (1993): *Engineering and the Mind's Eye*. The MIT Press.

[3] Filipič B., Urbančič T., Križman V. (1999): A combined machine learning and genetic algorithm approach to controller design. *Engineering Applications of Artificial Intelligence*, Vol.12, No.4, 401-409.

[4] Michie D. (1986): Machine learning and knowledge acquisition. *Experts systems: Automating Knowledge Acquisition* (eds. Michie D., Bratko I.). Addison Wesley.

[5] Michie D. (1995): Consciousness as an Engineering Issue, Part II. *Journal of Consciousness Studies*, Vol.2, No.1, 52–66.

[6] Michie D., Bain M., Hayes-Michie J. (1990): Cognitive models from subcognitive skills. *Knowledge-Based Systems in Industrial Control* (eds. Grimble M., McGhee J., Mowford P.), Stevenage: Peter Peregrinus, 71–99.

[7] Ryle, G. (1978): *The Concept of Mind.* Penguin Books Ltd., Harmondsworth, Middlesex, England.

[8] Sammut C., Hurst S., Kedzier D., Michie D. (1992): Learning to fly. *Proceedings of the Ninth International Workshop on Machine Learning* (eds. Sleeman D., Edwards P.), Morgan Kaufmann, 385–393.

[9] Šuc D., Bratko I. (2000): Skill Modeling through Symbolic Reconstruction of Operator's Trajectories. *IEEE Transaction on System, Man and Cybernetics, Part A: Systems and Humans*, Vol.30, No.6, 617-624.

[10] Urbančič T. (1994): Automated knowledge synthesis for dynamic systems control, Ph.D. Thesis, Faculty of Electrical Engineering and Computer Science, University of Ljubljana (in Slovenian).

[11] Urbančič T., Bratko I. (1994a): Reconstructing human skill with machine learning. Proceedings of the 11th European Conference on Artificial Intelligence ECAI 94 (ed. Cohn A.), John Wiley & Sons, 498–502.

[12] Urbančič T., Bratko I. (1994b): Learning to control dynamic systems. In *Machine Learning, Neural and Statistical Classification* (eds. Michie D., Spiegelhalter D., and Taylor C., editors. Ellis Horwood, 247-269.

[13] Urbančič T., Križman V., Filipič B. (1998): Learning and refining human skill of dynamic systems control. *Proceedings of the Seventh Electrotechnical and Computer Science Conference ERK '98,* Ljubljana, B: 133–136 (in Slovenian).

A model for compressing probabilities in belief networks

Shichao Zhang[1,2] and Chengqi Zhang[1]
[1] School of Computing and Mathematics
Deakin University, Geelong, Vic 3217, Australia
{scz,chengqi}@deakin.edu.au
[2] Institute of Logic and Cognition, Zhongshan University
Guangzhou, 515063, P.R. China

Keywords: Probabilistic Reasoning; Belief Network; Fuzzy Reasoning; Compressibility of Information; Encode Technology.

Received: July 18, 2001

Probabilistic reasoning with belief (Bayesian) networks is based on conditional probability matrices. Thus it suffers from NP-hard implementations. In particular, the amount of probabilistic information necessary for the computations is often overwhelming. So, compressing the conditional probability table is one of the most important issues faced by the probabilistic reasoning community. Santos suggested an approach (called linear potential functions) for compressing the information from a combinatorial amount to roughly linear in the number of random variable assignments. However, much of the information in Bayesian networks, in which there are no linear potential functions, would be fitted by polynomial approximating functions rather than by reluctantly linear functions. For this reason, we construct a polynomial method to compress the conditional probability table in this paper. We evaluated the proposed technique, and our experimental results demonstrate that the approach is efficient and promising.

1 Introduction

Bayesian networks are directed acyclic graphs, where each node represents a random variable, and is attached with conditional probability tables to the node given its parents. The methods have been widely accepted as a suitable, general and natural knowledge representation framework for reasoning and decision making under uncertainty. However, computing with these networks has been proven to be *NP-hard* as we might have expected [3, 15]. This has generally prevented problem formulations from utilizing the full representational capabilities of Bayesian networks. On the other hand, there are two key factors which prevent the general use of Bayesian networks. The first is network topology, and the second is the conditional probability table size [13]. This means that there can be no generally efficient procedure for practical applications in Bayesian networks.

Recently, various researches have attempted to improve the probabilistic reasoning model. For example, by using probabilistic information compressed as an approximation function [13], "Noisy-OR" models [11, 10, 16], a search algorithm for estimating posterior probabilities in Bayesian networks [12], and by propagating imprecise probabilities in Bayesian networks [8]. Some general researches are in [12, 14, 15, 17, 18]. The former suggests a new approach for manipulating the probabilistic information given. It can avoid being overwhelmed by essentially compressing the information using approximation functions called *linear potential functions*. However, much of the information in Bayesian networks, in which there are no linear potential functions, would be fitted by polynomial approximating functions rather than by reluctantly linear functions. For this reason, we construct a *polynomial method* to compress the conditional probability table *encoding*. We evaluated the proposed technique, and our experimental results demonstrate that the approach is efficient and promising.

The rest of this paper is organized as follows. We begin with briefly presenting preliminaries in Section 2. In Section 3, we first give a pre-process to actual data that are fitted to a polynomial function using an encoding technique. And then we construct a method for fitting this kind of probabilistic information in order to compress more information than in Santos' model. Section 4 presents the algorithm of polynomial function constructed, and the efficiency of the polynomial approximation function. Finally, a summary of this paper is shown in the last section.

2 Preliminaries

In this section, we firstly describe our motivation. Then simply recall some related work. Finally, some needed concepts are presented.

2.1 Motivation

Santos' method (Fro details, please see [13]), called a linear potential function (LPF), use an approximation function to capture all the values in the conditional probability table in a given Bayesian network. It can potentially

reduce the information from a combinatorial amount to roughly linear in the number of random variable assignments. The key problem of constructing approximation functions for a given table is how to choose encoders for the random variables. LPF is an approximation function $S(E_{C_0}(c_0), E_{C_1}(c_1), \cdots, E_{C_n}(c_n))$ of the form

$$S(E_{C_0}(c_0), E_{C_1}(c_1), \cdots, E_{C_n}(c_n)) =$$
$$exp(k_0 E_{C_0}(c_0) + k_1 E_{C_1}(c_1) + \cdots + k_n E_{C_n}(c_n) + k),$$

where, $E_{C_i}(c_i)$ is the encoder of C_i, which $E_{C_i}(c_i)$ is a one-to-one mapping from $R(C_i)$ to the set of positive rational numbers, $k, k_0, k_1, \cdots, k_n$ are constants to be determined. The ideal case is

$$S(E_{C_0}(c_0), E_{C_1}(c_1), \cdots, E_{C_n}(c_n)) =$$
$$P(C_0 = c_0 | C_1 = c_1, \cdots, C_n = c_n).$$

In order to determine $k, k_0, k_1, \cdots, k_n$ and $E_{C_i}(c_i)$, it can accomplish this by minimizing the following sum over the entries in a given conditional probability table T:

$$\sum_{(c_0,\cdots,c_n)\in T} [\sum_{j=0}^{n} k_j E_{C_j}(c_j) + k -$$
$$lnP(C_0 = c_0 | \cdots, C_l = c_l, \cdots)]^2.$$

It can solve $E_{C_i}(c_i)$ by using the "absorb" technique and the partial derivatives over the above sum as follows,

$$E_{C_i}(c_i) =$$
$$\frac{1}{\Pi_{\substack{j=0 \\ j\neq i}}^{n} m_j} \{ \sum_{\substack{(c'_0,\cdots,c'_n)\in T \\ c'_i = c_i}} lnP(C_0 = c'_0 | \cdots, C_l = c'_l, \cdots)\}$$
$$- \frac{n\xi(T)}{(n+1)\Pi_{j=0}^{n} m_j}.$$

where $m_j = |R(C_j)|$ for $j = 0, \cdots, n$ and

$$\xi(T) = \sum_{(d_0,\cdots,d_n)\in T} lnP(C_0 = d_0 | \cdots, C_l = d_l, \cdots).$$

Now we demonstrate the uses of LPF by the example in [13]. For example, consider the simple table consisting of only one random variable, say A. Suppose $R(A) = \{$ red, green, yellow, blue, purple $\}$ and the table is

$$P(A = red) = 0.02$$
$$P(A = green) = 0.50$$
$$P(A = yellow) = 0.00$$
$$P(A = blue) = 0.37$$
$$P(A = purple) = 0.11$$

First, let's use the simple encoding of the colours to $E_A(red) = 1, E_A(green) = 2, E_A(yellow) = 3, E_A(blue) = 4, E_A(purple) = 5$ based on the order they were shown above. In this way, the points: (1, 0.02), (2, 0.50), (3, 0.00), (4, 0.37), (5, 0.11) are not able to fit in a line. Now, let's choose another encoding of the colours to $E_A(yellow) = 0, E_A(red) = 0.2, E_A(purple) = 3, E_A(blue) = 3.7, E_A(green) = 5$. Then, the points: (0, 0.00), (0.2, 0.02), (1.1, 0.11), (3.7, 0.37), (5, 0.50) are able to approximately fit in a line. Hence, we can replace the information in the table with a simple continuous real function by using Santos' method.

However, information in tables, for which there is no linear potential function, would be fitted by polynomial approximating functions. For example, Suppose the information of the above table is

$$P(A = red) = 0.06$$
$$P(A = green) = 0.48$$
$$P(A = yellow) = 0.00$$
$$P(A = blue) = 0.16$$
$$P(A = purple) = 0.30$$

Certainly, the encoder of the colours is very difficult to be constructed by using the absorption methods (scale, translation) in [13] such that replace the information in the table with a linear approximating function and guarantee that the encoder of the colors is invertible simultaneously. But we can replace the information in the above table with an approximation polynomial function by using the following method in this paper. For example, let the encoders of the data of A and $P(A)$ are as

$$E_A(yellow) = 1$$
$$E_A(red) = 2$$
$$E_A(blue) = 3$$
$$E_A(purple) = 4$$
$$E_A(green) = 5$$

and

$$E_{P(A=yellow)} = 0$$
$$E_{P(A=(red)} = 6$$
$$E_{P(A=blue)} = 16$$
$$E_{P(A=purpl)} = 30$$
$$E_{P(A=green)} = 48$$

We can obtain an approximation polynomial function fitting these data as follows.

$$E_{P(A)}(x) = 2E_A^2(x) - 2$$

and

$$P(A = a) = E^{-1}_{P(A)}(a) = E_{P(A)}(a)/100.$$

To compress this kind of tables, we would use approximation polynomial functions rather than linear functions.

On the other hand, $k_0E_{C_0}(c_0) + k_1E_{C_1}(c_1) + \cdots + k_nE_{C_n}(c_n) + k$ must be a non-positive value. This can be easily resolved by selecting the translation constant k such that all encoders are positive. However, in our opinions, to insure $k_0E_{C_0}(c_0) + k_1E_{C_1}(c_1) + \cdots + k_nE_{C_n}(c_n) + k$ and all encoders are positive, we can choose an encoder for $P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n)$, it will be written as $E_{P(.)}$. For example,

$$E_{P(C_0=c_0|C_1=c_1,\cdots,C_n=c_n)} =$$

$$10^d P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n),$$

where, d is the bit numbers after decimal point of $P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n)$. For the sake of simplicity, in the following sections, we use the above method to choose the encoders for $P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n)$, and the same method as that of [13] to choose the encoders for the random variables.

2.2 Related Work

Bayesian networks (or belief networks), one of the most popular models for probabilistic reasoning, are directed acyclic graphs, where each node represents a random variable and is attached with a conditional probability of the node given its parents. They have been widely accepted as a suitable, general and natural knowledge representation framework for reasoning and decision making under uncertainty. They have been successfully applied to such diverse areas as medical diagnosis [6], diagnosis of bottlenecks in computer systems [2], circuit fault detection [10], planning systems [7], fraud detection [5], and advisory and control system for colon endoscopy [9].

In probabilistic reasoning, the method of propagating probabilities is with multiplying a vector by a matrix in Bayesian networks. However, though it has a good (or solid) theoretical basis and some successful applications, the computing complexity with general belief networks has been proven to be NP-hard [3, 15], which has generally prevented problem formulations from utilizing the full representational capabilities of Bayesian networks. On the other hand, there are two key factors which prevent the general use of Bayesian networks. The first is network topology and the second factor is conditional probability table size [13]. In the construction of a Bayesian network, it is always assumed that the variables starting from the same parent are conditionally independent. In practice, this assumption may not hold. Consequently, the reasoning with this assumption may give rise to incorrect solutions [9]. Usually, the probabilities in Bayesian networks are treated as though they were known precisely. In the present paper we analyze Bayesian networks in which the probabilities are not known precisely [8]. All of them mean that there can be no generally efficient procedure in Bayesian networks for practical applications.

Recently, many researchers attempt to improve the probabilistic reasoning model. Some of them concentrate on the conditional probability table size, such as probabilistic information can be compressed as an approximation function [13], the table size problems have been developed such as independence-based assignments [14], "Noisy-OR" models [10, 11, 16]. Some of them focus on perfecting the model, such as an optimal approximation algorithm for Bayesian inference [4], the creation of a hidden node [9], a search algorithm for estimating posterior probabilities in Bayesian networks [12], propagating imprecise probabilities in Bayesian networks [8], and optimization of Pearl's method of conditioning and greedy- like approximation algorithms for the vertex feedback set problem [1]. And other more general improvements are [15, 17, 18]. Santos [13] suggested a new approach for manipulating the probabilistic information given. It can avoid that the problem that the amount of probabilistic information is overwhelming, by essentially compressing the information using approximation functions called linear potential functions(LPF). In the this paper, we presented alternative polynomial functions for Bayesian network by encoding.

2.3 Needed Concepts

Before we head off into constructing polynomial functions for compressing Bayesian computations, it is necessary to recall some needed concepts used throughout this paper.

In this paper, upper case letters such as $A, B, \cdots$ will represent random variables and lower case letters such as $a, b, \cdots$ will represent the possible assignments to the associated upper case letter random variable. $\Theta = (V, P)$ will represent a Bayesian network, where V is the set of random variables in the Bayesian network and P is a set of conditional probabilities associated with the network. $P(A = a|C_1 = c_1, ..., C_n = c_n) \in P$ iff $C_1, ..., C_n$ are all the immediate parents of A and there is an edge from C_i to A, $i = 1, 2, \cdots, n$, in the network.

Definition 1 *Given a random variable A, the set of possible values for A, known as the range of A, will be denoted by $R(A)$. For $x \in R(A)$, x is the point value of A. And $|R(A)|$ denotes the number of possible values of A.*

For example, let the possible values of random variable A are *red, green, yellow, blue, purple*, then $R(A) =$ { red, green, yellow, blue, purple }, and $|R(A)| = 5$.

Definition 2 *Given a random variable A, all point values of A can construct a vector such as $(x_1, x_2, ..., x_n)$. Each state of A can be described by its point values associated with probabilities in the vector, that is, $(P(x_1) = a_1, P(x_2) = a_2, ..., P(x_n) = a_n)$ is a state (or an observation) of A. This state will be written as $(a_1, a_2, ..., a_n)$. All states of A construct the state space of A, which will be denoted by $S(A)$.*

For example, let the possible values of random variable A are *red, green, yellow, blue, purple*, then $(P(red) =$

$0.3, P(green) = 0.25, P(yellow) = 0.05, P(blue) = 0.11, P(purple) = 0.29) \in S(A)$, or (0.3, 0.25, 0.05, 0.11, 0.29) $\in S(A)$ is a state of A.

There are infinite elements in $S(A)$. We concentrate on a random sample space $\Omega(A)$ of $S(A)$ in this paper, because it is impossible to give a consideration to all elements in $S(A)$. Let the sample space have ℓ elements in $\Omega(A)$, where ℓ as the capacity of $\Omega(A)$, denoted by $\Im(\Omega(A))$.

Definition 3 *Given a random variable $A \in V$, a one-to-one mapping $E_A : S(A) \longrightarrow \Re^+$, is called an encoder for A, where $\Re^+$ is the set of positive rational numbers.*

The requirement that an encoder is one-to-one mapping is used to assure that the encoder is in reversible because we need to be able to recover the original probability values after being handled.

Definition 4 *Given a rule $X \longrightarrow Y$(or X is a directly parent of Y in a Θ), let $R(X) = \{x_1, x_2, ..., x_m\}$, $R(Y) = \{y_1, y_2, ..., y_n\}$, then $M_{Y|X}$ is defined as*

$$M_{Y|X} \triangleq P(y|x) \triangleq P(Y = y|X = x) =$$

$$\begin{bmatrix} p(y_1|x_1) & P(y_2|x_1) & \cdots & p(y_n|x_1) \\ p(y_1|x_2) & p(y_2|x_2) & \cdots & p(y_n|x_2) \\ \cdots & & & \\ p(y_1|x_m) & p(y_2|x_m) & \cdots & p(y_n|x_m) \end{bmatrix}$$

where, $p(y_j|x_i) = p(Y = y_j|X = x_i)$ are conditional probabilities of $Y = y_j$ given $X = x_i$, $i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$. Let $x = (p(x_1), p(x_2), ..., p(x_m))$ be an observation, we can get y of the form $(p(y_1), p(y_2), ..., p(y_n))$ as

$$y = xM_{Y|X}.$$

Now we illustrate the use of this method with an example.

Example 1 *In many cracked criminal cases, some discovered suspicious footprints at the scene of a crime are generally useful for estimating the height and weight of a suspect. Let X denotes the length of footprint of a suspect, Y the height of the suspect. For simplicity, supposed the domain of X be $\{long, middle, short\}$ and the domain of Y be $\{tall, middle, small\}$. The experience shown, Y is relatively dependent on X. For a data set D of the length of footprint and the height of people in some town, the conditional probability matrix of Y given X is as follows.*

$$M_{Y|X} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Now, supposed an evidence be "the probabilities that the length X of a discovered suspicious feetprint is long, middle and short are 0.7, 0.2 and 0.1, respectively", then we have,

$$[0.7\,0.2\,0.1] \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.59 & 0.24 & 0.17 \end{bmatrix}.$$

So, $y = (0.59, 0.24, 0.17)$ is the result what we want. This means, 0.59, 0.24, and 0.17 are the probabilities that the height of the suspect is tall, middle, small.

Our work in this paper is focused on compressing the probability information such as in matrix $M_{Y|X}$.

3 Compressing Model

In his article, Santos [13] established a new method in which probabilistic information can be compressed as an approximation function. However, for some of the information, there is not any linear potential function to represent it. So, if we want to compress this information, it would be fitted by polynomial approximating functions. Hence, we now show a method for fitting this kind of probabilistic information.

3.1 Encoder

We can choose an encoder for $P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n)$, it will be written as $E_{P(.)}$. For example,

$$E_{P(C_0=c_0|C_1=c_1,\cdots,C_n=c_n)} =$$
$$10^d P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n),$$

where, d is the bit numbers after decimal point of $P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n)$. For the sake of simplicity, in the following sections, we use the above method to choose the encoder for $P(C_0 = c_0|C_1 = c_1, \cdots, C_n = c_n)$, and the same method as that of [13] to choose the encoder for the random variables.

For the sake of convenience, suppose the encoders of both X and $P(X)$ use directly symbols as $x_1, x_2, ..., x_n$, and $y_1, y_2, \cdots, y_n$ as follows,

E_X	x_1	x_2	$\cdots$	x_n
$E_{P(X)}$	y_1	y_2	$\cdots$	y_n

where E_X is the encoder of X and $E_{P(X)}$ is the encoder of $P(X)$.

Suppose $D = \{(x_i, y_i), 1 \leq i \leq n\}$ indicate the set of all above data. It needs a preprocess to this data.

3.2 Pre-process of Data

Suppose $D = \{(x_i, y_i), 1 \leq i \leq n\}$ indicate the set of all above data. It needs a preprocess to this data.

First, a set D is divided along the y-axis into k groups of data as,

$$D_1, D_2, ..., D_k.$$

This should satisfy 3 conditions as follows.

(1) $\forall(x_{ij}, y_{ij}) \in D_i, y_{ij} \in [\underline{y_i}, \overline{y_i}]$, where, $\underline{y_i}$ and $\overline{y_i}$ are *supremum* and *infimum* of y_{ij} in D_i, respectively, $j = 1, 2, \cdots, n_i$, where n_i is the observed point numbers in D_i, and $n = \sum_{i=1}^{k} n_i$. And for $y \in [\underline{y_i}, \overline{y_i}]$ and $\forall y' \in [\underline{y_{i+1}}, \overline{y_{i+1}}]$, it must hold $y \leq y'$;

(2) If D_i border D_j, suppose $j = i + 1$, then $[\underline{y_i}, \overline{y_i}] \bigcap [\underline{y_j}, \overline{y_j}]$ exist at most one point $\underline{y_j} = \overline{y_i}$;

(3) y and x are approximately linear-relationship on $[\underline{y_i}, \overline{y_i}]$, that is, there exist two constants a and b so that $y = a + bx$ can close to the points that $y \in [\underline{y_i}, \overline{y_i}]$.

In fact, it can adopt a linear-relational coefficient as a measure to divide the set D.

After D is divided, the data in D_i is a linear-relationship. Hence, we can use the extreme points of each set D_i as its deputy points. In other words, to select the points $\underline{y_1'}, \overline{y_1'}, \underline{y_2'}, ..., \overline{y_k'}$ as the deputy points. Let m be the number of deputy points, and we rename them respectively as $y_1, y_2, ..., y_m$, the corresponding point values of X are $x_1, x_2, \cdots, x_m$, writing the point values of $P(X)$ as $G_1(x_1), G_1(x_2), ..., G_1(x_m)$. Therefore, we have constructed a deputy value set $U = (x_i, G_1(x_i))$, or

E_X	x_1	x_2	$\cdots$	x_m
$E_{P(X)}$	$G_1(x_1)$	$G_1(x_2)$	$\cdots$	$G_1(x_m)$

3.3 Constructing the Polynomial Function

Theorem 1 *For U, the approximating polynomial function for compressing probabilities can be constructed as*

$$F(x) = F_1(x) + \sum_{i=1}^{N}(F_{i+1}(x)(\prod_{j=1}^{2i}(x - x_j)))$$

where,

$$F_k(x) =$$

$$\frac{(x - x_{2k})}{(x_{2k-1} - x_{2k})}G_k(x_{2k-1}) + \frac{(x - x_{2k-1})}{(x_{2k} - x_{2k-1})}G_k(x_{2k}),$$

$k = 1, 2, \cdots, N$, N is the fitting times, G_k is the fitted data.

□

We will now show how we construct this approximating polynomial function as the proof of Theorem 1.

For U, suppose the polynomial function be

$$F(x) = F_1(x) + (x - x_1)(x - x_2)G_2(x)$$

where

$$F_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}G_1(x_1) + \frac{(x - x_1)}{(x_2 - x_1)}G_1(x_2),$$

and where the values of $G_1(X) = F(x)$ are listed in the above table. $(x-x_1)(x-x_2)G_2(X)$ represents a remainder term. Then the values of $G_2(x)$ at the other points can be solved as,

$$G_2(x_i) = \frac{G_1(x_i) - F_1(x_i)}{(x_i - x_1)(x_i - x_2)}$$

where $i = 3, 4, \cdots, m$.

If $|(G_2(x_3) + G_2(x_4) + ... + G_2(x_m))(x - x_1)(x - x_2)| < \delta$, where $\delta > 0$ is a small value determined by an expert, or $m - 2 \leq 0$, then we end the procedure, and get $F(x) = F_1(x)$. The term $G_2(x)$ is neglected. Otherwise, go on the above procedure for the remaining data. For the following data:

E_X	x_3	x_4	$\cdots$	x_m
$G_2(x)$	$G_2(x_3)$	$G_2(x_4)$	$\cdots$	$G_2(x_m)$

Let

$$G_2(x) = F_2(x) + (x - x_3)(x - x_4)G_3(x)$$

where

$$F_2(x) = \frac{(x - x_4)}{(x_3 - x_4)}G_2(x_3) + \frac{(x - x_3)}{(x_4 - x_3)}G_2(x_4).$$

And $(x-x_1)(x-x_2)(x-x_3)(x-x_4)G_3(x)$ is a remainder term. Then the values of $G_3(x)$ at the other points can be solved as,

$$G_3(x_i) = \frac{G_2(x_i) - F_2(x_i)}{(x_i - x_3)(x_i - x_4)},$$

where $i = 5, 6..., m$.

If $(G_3(x_5)+G_3(x_6)+...+G_3(x_m))(x-x_1)(x-x_2)(x-x_3)(x-x_4) < \delta$, where $\delta > 0$ is a small value determined by experts; or $m - 4 \leq 0$, then end the procedure, and obtain $F(x) = F_1+(x-x_1)(x-x_2)F_2(x)$, the term $G_3(x)$ is neglected. Otherwise, go on with the above procedure for the remaining data.

We can obtain a function after repeating the above procedure several times. However, the above procedure is repeated N times ($N \leq [m/2]$) at the most. Finally, we can gain the approximating function as follows.

$$\begin{aligned} F(x) &= F_1(x) + G_2(x)(x - x_1)(x - x_2) \\ &= F_1(x) + (F_2(x) + \\ &\quad G_3(x)(x - x_3)(x - x_4))(x - x_1)(x - x_2) \\ &\quad \cdots \\ &= F_1(x) + \sum_{i=1}^{m}(F_{j+1}(x)(\prod_{j=1}^{2i}(x - x_j))). \end{aligned}$$

$F(X)$ is the approximating function of $E_{P(X)}$ that we require. It is a polynomial of which the order is not over $2N + 1$.

$F(x)$ is the approximating function what we want. It is a polynomial which order is not over $2N+1$.

According to the encoders of E_X and E_Y, if $E_X(a_1) > E_X(a_2)$, then $E_Y(b_1) > E_Y(b_2)$. So, if $F(x)$ can fit perfectly the data of E_X and E_Y, then $F(x)$ must be an increasing function. Now we have the following theorem.

Theorem 2 *$F(x)$ is an increasing function under the above encoders, or for $\forall x', x''(x' > x'') \longrightarrow F(x') > F(x'')$.*

Proof: First, we prove $F_i(x)$ $(i = 1, 2, \cdots, m)$ are increasing functions. We apply induction to $F_i(x)$.

(1). When $i = 1$,

$$F_1(x) = \frac{(x-x_2)}{(x_1-x_2)}G_1(x_1) + \frac{(x-x_1)}{(x_2-x_1)}G_1(x_2).$$

For $\forall x', x'', x' > x''$,

$$F_1(x') - F_1(x'') =$$
$$\frac{(x'-x_2)}{(x_1-x_2)}G_1(x_1) + \frac{(x'-x_1)}{(x_2-x_1)}G_1(x_2)$$
$$-(\frac{(x''-x_2)}{(x_1-x_2)}G_1(x_1) + \frac{(x''-x_1)}{(x_2-x_1)}G_1(x_2))$$
$$= \frac{(x'-x'')(G_1(x_2)-G_1(x_1))}{(x_2-x_1)}$$
$$> 0.$$

That is, $F_1(x') > F_1(x'')$, or $F_1(x)$ is an increasing function.

(2). Suppose $F_{k-1}(x)$ is an increasing function, we want to prove $F_k(x)$ is an increasing functions. Because

$$F_k(x) =$$
$$\frac{(x-x_{2k})}{(x_{2k-1}-x_{2k})}G_k(x_{2k-1}) + \frac{(x-x_{2k-1})}{(x_{2k}-x_{2k-1})}G_k(x_{2k}).$$

and $F_{k-1}(x)$ is an increasing function, according to the method of constructing $G_k(x)$; G_k is an increasing function, or for $\forall x', x''(x' > x'') \longrightarrow G_k(x') > G_k(x'')$, so for $\forall x', x'', x' > x''$ we have

$$F_k(x') - F_k(x'') =$$
$$\frac{(x'-x_{2k})}{(x_{2k-1}-x_{2k})}G_k(x_{2k-1}) + \frac{(x'-x_{2k-1})}{(x_{2k}-x_{2k-1})}G_k(x_{2k})$$
$$-(\frac{(x''-x_{2k})}{(x_{2k-1}-x_{2k})}G_k(x_{2k-1}) + \frac{(x''-x_{2k-1})}{(x_{2k}-x_{2k-1})}G_k(x_{2k}))$$
$$= \frac{(x'-x'')(G_k(x_{2k})-G_k(x_{2k-1}))}{(x_{2k}-x_{2k-1})}$$
$$> 0.$$

That is, $F_k(x') > F_k(x'')$, or $F_k(x)$ is an increasing function. Hence, $F_i(x)$ $(i = 1, 2, \cdots, m)$ are increasing functions.

□

Because $H_i(x) = \prod_{j=1}^{2i}(x - x_j)$ $(i = 1, 2, \cdots, m)$ are increasing functions, so $H_i(x)F_i(x)$ $(i = 2, 3, \cdots, m)$ are increasing functions. That is, $F(x)$ is an increasing function, or for $\forall x', x''(x' > x'') \longrightarrow F(x') > F(x'')$

Furthermore, we have the following theorem.

Theorem 3 *Let $a_1, a_2, a_3 \in S(X)$, $b_1 = a_1 M_{Y|X}, b_2 = a_2 M_{Y|X}, b_3 = a_3 M_{Y|X}$, and $(E_X(a_1) > E_X(a_2) > E_X(a_3)) \bigwedge (E_Y(b_1) > E_Y(b_2) > E_Y(b_3))$, then $F(E_X(a_1)) > F(E_X(a_3)) > F(E_X(a_3))$.*

Proof: It can be proven immediately from the procedure of constructing $F(x)$ and Theorem 2.

□

Suppose $\delta = 0.1$ for the following examples. It is important to demonstrate the efficiency that the above function fits a given data set D, so the following data set is not the encoders of some rule. Now we select a data set of $x^2 + x + 2$ as,

X	0	1	2	3	4	5	6	7	8	9
Y	2	4	8	14	22	32	44	58	74	92

For the above data, the approximation function is

$$F(x) = F_1(x) + (x-0)(x-1)G_2(x)$$

where

$$F_1(x) = \frac{(x-1)}{(0-1)}2 + \frac{(x-0)}{(1-0)}4$$
$$= 2x + 2$$

$$G_2(2) = 1, G_2(3) = 1, G_2(4) = 1, G_2(5) = 1,$$

$$G_2(6) = 1, G_2(7) = 1, G_2(8) = 1, G_2(9) = 1.$$

Because $(2-0)(2-1)G_2(2) + (3-0)(3-1)G_2(3) + (4-0)(4-1)G_2(4) + (5-0)(5-1)G_2(5) + (6-0)(6-1)G_2(6) + (7-0)(7-1)G_2(7)(8-0)(8-1)G_2(8) + (9-0)(9-1)G_2(9) > 1$ and $10 - 2 > 0$, so for the following data:

X	2	3	4	5	6	7	8	9
$G_2(x)$	1	1	1	1	1	1	1	1

Let

$$G_2(x) = F_2(x) + (x-2)(x-3)G_3(x)$$

where

$$F_2(x) = \frac{(x-3)}{(2-3)}1 + \frac{(x-2)}{(3-2)}1$$
$$= 1$$

$$G_3(4) = 0, G_3(5) = 0, G_3(6) = 0,$$
$$G_3(7) = 0, G_3(8) = 0, G_3(9) = 0.$$

Because $(4-0)(4-1)(4-2)(4-3)G_3(4) + (5-0)(5-1)(5-2)(5-3)G_3(5) + (6-0)(6-1)(6-2)(6-3)G_3(6) +$

$(7-0)(7-1)(7-2)(7-3)G_3(7)(8-0)(8-1)(8-2)(8-3)G_3(8)+(9-0)(9-1)(9-2)(9-3)G_3(9)=0$, so

$$G_2(x) = F_2(x) + (x-2)(x-3)G_3(x)$$
$$= F_2(x)$$
$$= 1.$$

Furthermore, we can gain a polynomial function as

$$F(x) = F_1(x) + (x-0)(x-1)G_2(x)$$
$$= 2x + 2 + x(x-1)1$$
$$= x^2 + x + 2.$$

4 The Algorithm and Its Efficiency

This section shows the algorithm, then demonstrates the efficiency of the approximation function using examples and experiments.

4.1 Algorithm

Let probability table be as follows:

X	$X[1]$	$X[2]$	$\cdots$	$X[n]$
Y	$Y[1]$	$Y[2]$	$\cdots$	$Y[n]$

After the data is pre-processed, the table become into

X	$X[1]$	$X[2]$	$\cdots$	$X[m]$
Y	$Y[1]$	$Y[2]$	$\cdots$	$Y[m]$

The compressing algorithm is constructed below.

Algorithm 1 *PFCPI*

begin
(1) **readln**(δ);
(2) **for** $i := 1$ **to** n **do**
begin
readln$(X[i])$;
readln$(Y[i])$
end;
(3) **preprocess** *the data*;
let $N1 \leftarrow 0$;
for $i := 1$ **to** n **do**
for $j := 1$ **to** n **do**
let $GG[i,j] \leftarrow 0; FF[i,j] \leftarrow 0$;
(4) **for** $j := 1$ **to** m **do**
begin
let $X[j] \leftarrow$ *encoder* $X[j]$;
let $Y[j] \leftarrow$ *encoder* $Y[j]$;
let $GG[1,j] \leftarrow Y[j]$;
end;
(5) **if** $2N+1 \geq m$ **then goto** *(6)*;
(6) **for** $j := 2N+1$ **to** m **do**
begin
let $FF[N,j] \leftarrow \frac{X[j]-X[2N]}{X[2N-1]-X[2N]} GG[N,2N-1] + \frac{X[j]-X[2N-1]}{X[2N]-X[2N-1]} GG[N,2N]$
let $GG[N+1,j] \leftarrow \frac{GG[N,j]-FF[N,j]}{(X[j]-X[2N-1])(X[j]-X[2N])}$;
end;
(7) **if** $\sum |GG[N+1,j]| > \delta$ **then**
begin
let $N \leftarrow N+1$;
goto *(5)*;
end;
(8) **output** *the result* $F(x)$;
(9) **end** *of all.*

The algorithm $PFCPI$ is to generate polynomial function $F(x)$ for compressing Baysian computations, where function $FF[i,j]$ is the same as F_j described in Theorem 1, function $GG[i,j]$ is also the same as G_j described in Theorem 1. The initialization of the algorithm is done in Step (1) and (2); Step (3) pre-processes the given data; Step (4) encodes the given data; Step (5) checks the condition $2N+1 \geq m$; Step (6) calculates Nth function; Step (7) checks the condition $\sum |GG[N+1,j]| > \delta$; Step (8) outputs the result $F(x)$.

4.2 Examples

Now, we demonstrate the efficiency of the approximation function by an example.

It is important to demonstrate the efficiency with which the above function fit a given data set D. The following data set is directly from the example in Section 2, and the encoders of the data of A and $P(A)$ are as $(E_A(yellow) = 1, E_A(red) = 2, E_A(blue) = 3, E_A(purple) = 4, E_A(green) = 5$ and $E_{P(A=yellow)} = 0, E_{P(A=(red)} = 6, E_{P(A=blue)} = 16, E_{P(A=purpl)} = 30, E_{P(A=green)} = 48)$,

E_A	1	2	3	4	5
$E_{P(A)}$	0	6	16	30	48

For the above data, the approximation function is

$$F(x) = F_1(x) + (x-1)(x-2)G_2(x)$$

where

$$F_1(x) = \frac{(x-2)}{(1-2)}0 + \frac{(x-1)}{(2-1)}6$$
$$= 6x - 6.$$

$$G_2(3) = 2, G_2(4) = 2, G_2(5) = 2.$$

Because $G_2(3) + G_2(4) + G_2(5) > 1$, therefore for the following data:

E_X	3	4	5
$G_2(x)$	2	2	2

Let

$$G_2(x) = F_2(x) + (x-3)(x-4)G_3(x)$$

where

$$F_2(x) = \frac{(x-4)}{(3-4)}2 + \frac{(x-3)}{(4-3)}2$$
$$= 2$$

$$G_3(5) = 0.$$

Because $G_3(5) = 0$, so

$$G_2(x) = F_2(x) + (x-2)(x-3)G_3(x)$$
$$= F_2(x)$$
$$= 2.$$

Furthermore, we can gain a polynomial function as,

$$F(x) = F_1(x) + (x-1)(x-2)G_2(x)$$
$$= 6x - 6 + (x-1)(x-2)2$$
$$= 2x^2 - 2.$$

So, $E_{P(A)}(x) = 2E_A^2(x) - 2$ and $P(A = a) = E^{-1}_{P(A)}(a) = E_{P(A)}(a)/100$.

We have seen, that we need only twice the approximation procedures to acquire the final result, and this result fits the above data completely. In the other hand, this method doesn't require the encoders to be constructed sophisticatedly.

Now we give a demonstration to how the data of the given random node in a Bayesian network are fitted by the method. Let one conditional probability matrix of this node be,

$$M_{Y|X} = \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.3 & 0.4 & 0.3 \\ 0.5 & 0.2 & 0.3 \end{bmatrix}$$

According to our encoder method, the order of the point variables must be rearranged as x_3, x_2, x_1. Then them are renamed as $z_1 = x_3, z_2 = x_2, z_3 = x_1$. Now we select a random data set as

E_Z	1090	2080	8020	9010	800020	900010	1000000
E_Y	302248	302446	303634	303832	383626	393823	404020

For above data, the approximation function is

$$F(x) = F_1(x) + (x-1090)(x-2080)G_2(x)$$

where

$$F_1(x) = \frac{(x-2080)}{(1090-2080)}302248 + \frac{(x-1090)}{(2080-1090)}302446$$
$$= 0.2x + 302030$$

$$G_2(8020) = 0, G_2(9010) = 0,$$
$$G_2(800020) = -1.229933 * 10^{-7},$$
$$G_2(900010) = -1.092822 * 10^{-7},$$
$$G_2(1000000) = -9.83214561 * 10^{-8}$$

Because $||G_2(8020)(8020 - 1090)(8020 - 2080)|| + ||G_2(9010)(9010 - 1090)(9010 - 2080)|| + ||G_2(800020)(800020 - 1090)(800020 - 2080)|| + ||G_2(900010)(900010 - 1090)(900010 - 2080)|| + ||G_2(1000000)(1000000 - 1090)(1000000 - 2080)|| > 1$ and $7 - 2 > 0$, so for the following data:

E_Z	800020	900010	1000000
$G_2(x)$	-1.229933E-7	-1.092822E-7	-9.83214561E-8

Notice: Because $G_2(8020) = 0$ and $G_2(9010) = 0$, they don't need to be fitted.

Let

$$G_2(x) = F_2(x) + (x - 800020)(x - 900010)G_3(x)$$

where

$$F_2(x) = \frac{(x-900010)}{(800020-900010)}(-1.229933E-7) +$$
$$\frac{(x-800020)}{(900010-800020)}(-1.092822E-7)$$
$$= 1.37124 * 10^{-13}x - 2.326947 * 10^{-7}$$

$$G_3(1000000) = -1.375653 * 10^{-19}$$

Because $||G_2(1000000)(1000000 - 1090)(1000000 - 2080)(1000000 - 800020)(1000000 - 900010)|| > 1$ and $7 - 6 > 0$, so for the following data:

E_Z	900010	1000000
$G_2(x)$	0	$-1.375653E-19$

Let

$$G_3(x) = F_3(x) + (x - 900010)(x - 1000000)G_4(x)$$

where

$$F_2(x) = \frac{(x-1000000)}{(900010-1000000)}0 +$$
$$\frac{(x-900010)}{(1000000-900010)}(-1.375653E-19)$$
$$= 1.37579 * 10^{-24}x + 1.238225 * 10^{-18}$$

Because $7 - 7 = 0$ and $G_4(x)$ can be neglected, so we can gain a polynomial function as

$$F(x) = 0.2x + 302030 +$$
$$(x-1090)(x-2080)$$
$$(1.37124 * 10^{-13}x - 2.326947 * 10^{-7}) +$$
$$(x-1090)(x-2080)(x-800020)(x-900010)$$
$$(1.37579 * 10^{-24}x + 1.238225 * 10^{-18})$$

4.3 Analysis

To study the effectiveness of our model, we have performed several experiments. The algorithm is implemented on Sun SparcServer using Java.

The basic parameters used to generate the groups of data, which are belief networks, are listed as follows.

Table 1 *The basic parameters of belief network*

object	*the nodes of belief networks*
range	*the average range of all nodes in a belief network*

For the convenience of comparison, we randomly generate four groups of data. The main properties of the data sets are the following. The first group consists of 5 objects (nodes), which the average range of objects is 8; The second group consists of 10 objects, which the average range of objects is 10; The third group consists of 15 objects, which the average range of objects is 12; The fourth group consists of 20 objects, which the average range of objects is 10. The comparison of our model (written as PPF) with LPF on running time and space are listed in Table 2 and Table 3, respectively. And they are also illustrated in in Figure 1 and Figure 2, respectively.

Table 2 *The running time*

object	*5*	*10*	*15*	*20*
LPF	*17*	*32*	*50*	*68*
PPF	*14*	*27*	*39*	*52*

Table 3 *The running space*

object	*5*	*10*	*15*	*20*
LPF	*110*	*207*	*324*	*453*
PPF	*75*	*154*	*230*	*318*

As have seen, our experimental results demonstrate that the proposed approach is efficient and promising. This is due to the facts that (1) the proposed model is simpler than Santos' method [13] and (2) the selected data in the above experiments is more suitable to be fitted by polynomial functions than by linear functions.

5 Conclusion

Santos' method [13] of linear potential functions, can only compress information with linear form in Bayesian networks. However, much of the information in Bayesian networks, in which there are no linear potential functions, would be fitted by polynomial approximating functions rather than by reluctantly linear functions. This kind of information contained in Bayesian networks can be compressed into an approximating polynomial function in the above model. We construct a polynomial method to compress the conditional probability table in this paper. We evaluated the proposed technique, and our experimental results demonstrate that the approach is efficient and promising.

Acknowledgments

We would like to thank the anonymous reviewers for their good comments on this paper.

In addition, this research is partially supported by the small grant from the Australian Research Council and partially supported by the large grant from the Australian Research Council (A49530850).

References

[1] Becker A. and Geiger D., Optimization of Pearl's method of conditioning and greedy- like approximation algorithms for the vertex feedback set problem, *Artificial Intelligence*, **83**(1996): 167-188.

[2] Breese J. and Blake R., Automating computer bottleneck detection with belief nets, *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal, Que, 1995: 36-45.

[3] Cooper G., The computational complexity probabilistic inference using belief networks, *Artificial Intelligence*, **42**(1990): 393-405.

[4] Dagum P. and Luby M., An optimal approximation algorithm for Bayesian inference, *Artificial Intelligence*, **93**(1997): 1-27.

[5] Ezawa K. and Schuermann T., Fraud/uncollectible debt detection using a Bayesian network based learning system: a rare binary outcome with mixed data structures, *Proceedings Eleventh Conference on Uncertainty in Artificial Intelligence*, Seattle, WA, 1994: 227-234.

[6] Heckerman D., Horvitz E. and Nathwani B., Toward normative expert systems: Part I. The pathfinder project, *Meth. Inform. Med.*, **31**(1992): 90-105.

[7] Kirman J., Nicholson A., Lejter M., Santos J. and Dean T., Using goals to find plans with high expected utility, *Proceedings of the 2nd European Workshop on Planning*, 1993.

[8] Kleiter G., Propagating imprecise probabilities in Bayesian networks, *Artificial Intelligence*, 88(1996): 143-162.

[9] Kwoh C. and Gillies D., Using hidden nodes in Bayesian networks, *Artificial Intelligence*, **88**(1996): 1-38.

[10] Pearl J., Probabilistic reasoning in intelligent systems: Networks of plausible inference, *Morgan Kaufmann Publishers*, 1988.

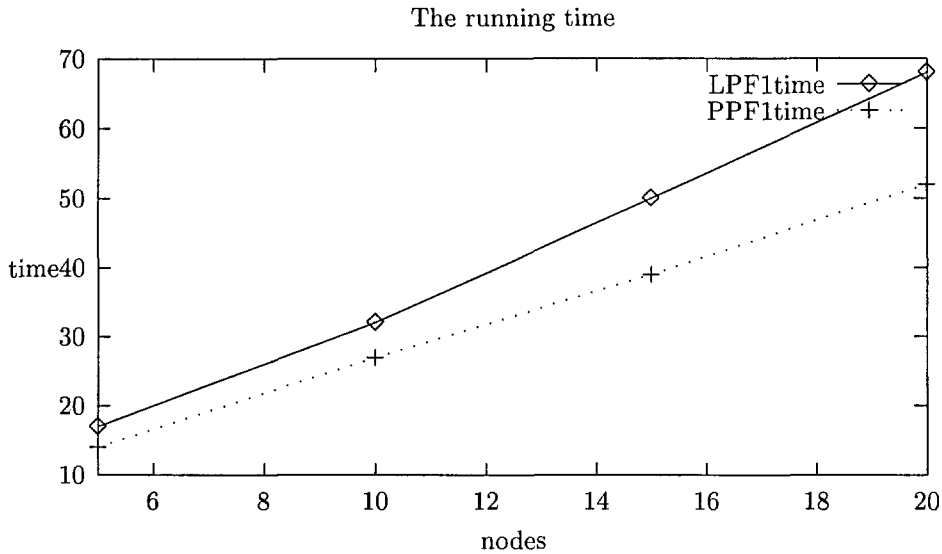


Figure 1: The comparison on time

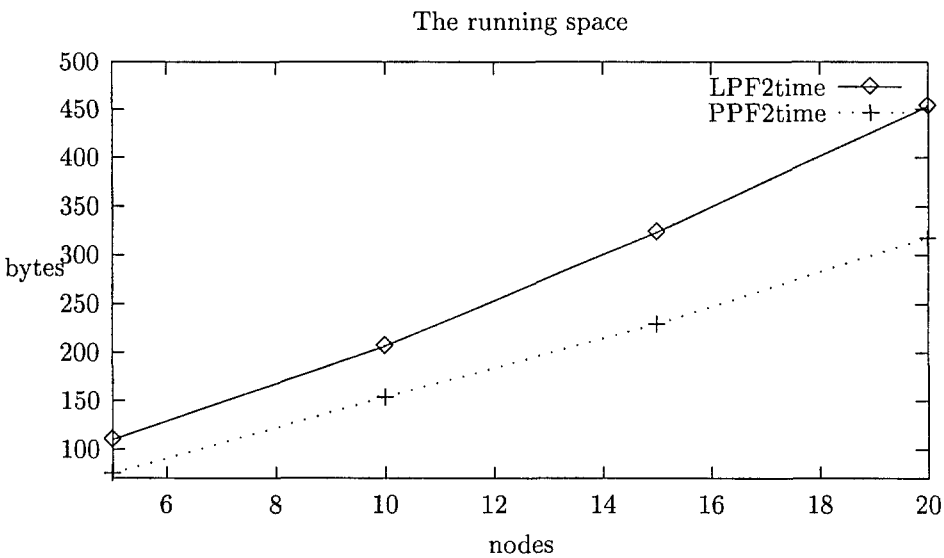


Figure 2: The comparison on space

[11] Peng Y., and Reggia J., Plausibility of diagnostic hypotheses: The nature of simplicity, *Proceedings of AAAI'86*, Menlo Park, Calif.,1986: 140-147.

[12] Poole D., Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks, *Artificial Intelligence*, 88(1996): 39-68.

[13] Santos J., On linear potential functions for approximating Bayesian computations, *Journal of The ACM*, 43(1996): 399-430.

[14] Shimony S., The role of relevance in explanation, I: Irrelevance as statistical independence, *Int. J. Approx. Reasoning*, 1993.6.

[15] Shimony S. and Charniak E., A new algorithm for finding map assignments to belief networks. *Proceedings of the conference on uncertainty in artificial intelligence*, Morgan Kaufmann, san Francisco, calif., 1990.

[16] Srinivas S., A generalization of the noisy-or model. *Proceedings of the conference on uncertainty in artificial intelligence*, Morgan-Kaufmann, San Francisco, Calif., 1993: 208-215.

[17] Shichao Zhang and Chengqi Zhang, A Model for Propagating Probabilities, *Proceedings of ICCIMA'98*, Australia, 1998.

[18] Shichao Zhang and Chengqi Zhang, A Method of Learning Probabilities in Bayesian Networks, *Proceedings of ICCIMA'98*, Australia, 1998.

Population migration: a meta-heuristics for stochastic approaches to constraint satisfaction problems

Kazunori Mizuno, Seiichi Nishihara, Hitoshi Kanoh and Isao Kishi
Institute of Information Sciences and Electronics, University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan
mizuno@algor.is.tsukuba.ac.jp, http://www.npal.is.tsukuba.ac.jp/

Keywords: constraint satisfaction, search algorithms, stochastic search, meta-heuristics

Received: July 25, 2000

A meta-heuristics for escaping from local optima to solve constraint satisfaction problems is proposed, which enables self-adaptive dynamic control of the temperature to adjust the locality of stochastic search. In our method, several groups with different temperatures are prepared. To each group the same number of candidate solutions are initially allotted. Then, the main process is repeated until the procedure comes to a certain convergence. The main process is composed of two phases: stochastic searching and population tuning. As for the latter phase, after evaluating the adaptation value of every group, migration of some number of candidate solutions in groups with lower values to groups with higher values are induced. Population migration is a kind of parallel version of simulated annealing, where several temperatures are spatially distributed. Some experiments are performed to verify the efficiency of the method applied to constraint satisfaction problems. It is also demonstrated that population migration is exceptionally effective in the critical region where phase transitions occur.

1 Introduction

A constraint satisfaction problem (CSP) involves finding values for problem variables which are subject to constraints specifying the acceptable combinations of values. Such combinatorial search problems are ubiquitous in artificial intelligence and pattern analysis, including scheduling and planning problems. Most of the previous work on CSP algorithms has adopted a systematic backtracking-based approach in which a partial assignment to the variables is incrementally extended. However, this approach often needs too much time to find a solution on large-scale problems due to their exponential complexity. In contrast, a repair-based stochastic approach, which starts with a complete but inconsistent assignment and then repeats repairs of constraint violations until a consistent assignment is achieved[1], has recently made remarkable progress because this approach may sometimes solve large-scale problems in a practical time. However, this approach has a drawback of getting caught in locally optimal states that are not acceptable as solutions. Therefore, many techniques to escape from local optima have recently been proposed for stochastic approaches[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. We call such techniques meta-heuristics.

Simulated annealing (SA) has been studied as a kind of meta-heuristics which is widely applicable to stochastic approaches[2, 11, 12, 14]. SA involves a unique operation, after which it was named, that gradually reduces the value of a parameter, or a temperature, for determining a state transition probability. Accordingly, as the search proceeds, its focal area varies from global to local. The main drawback to SA, however, is its difficulty in deciding in advance the schedule of temperature reduction because this depends on each problem.

In this paper, we propose a new meta-heuristics with a self-adjustment mechanism which automatically, but implicitly, controls its temperature schedule for a given problem during the search. First, several groups preassigned with different temperatures are created, in each of which an equal number of candidate solutions are stored. Then, the main process is repeated until the system comes to a certain convergence. The main process is composed of two phases: searching and population tuning. As for the latter phase, after evaluating all the adaptation values of groups, migration operations are executed, in which a proper number of candidate solutions in groups with low adaptation values are moved into groups with high adaptation values taking account of how far those values differ from the average.

CSP is well-known as an NP-complete problem, but actual problem instances with such computational complexity are found only in a locally limited region of the problem space. Recent studies have revealed that really hard problems tend to happen in situations very similar to physical phase transitions. Hence it is important for the studies of meta-heuristics to place their interests on how well they cope with phase transitions. We show that our method is efficient especially for hard problems found in the region of phase transitions.

In section 2, after reviewing stochastic approaches to

CSPs and meta-heuristics for them, we explain the phase transition of graph-colouring problem, which has become one of the standard benchmark problems for testing CSP algorithms. In section 3, we give the basic idea of population migration and propose a concrete algorithm. In section 4, we show experimental results for our method from two viewpoints: its efficiency compared with other meta-heuristics and its effect on phase transitions.

2 Stochastic Approaches and Phase Transition

2.1 Stochastic Approaches

Many methods to solve CSPs have been proposed. Two approaches can be distinguished: systematic and stochastic. The former is the constructive approach to a solution based on tree search with backtracking. Many heuristics like forward checking, partial and full-looking ahead, back marking, arc and path-consistency have been developed. However, it is still difficult to solve really hard CSPs completely within tractable time. In contrast, the stochastic approach based on the repair-oriented search starting from an initial candidate solution (i.e., an inconsistent complete assignment of values to all variables) often gives a final or semi-optimal solution in a practical time. Hill-climbing is one of the standard search algorithms that navigates the search space while attempting to minimize the total constraint violations included in the present candidate solution. It is efficient when the landscape of search space is simple, i.e. single peaked or so, but, at times, hill-climbing tends to get caught in local optima that are not acceptable as solutions. Stochastic hill-climbing, abbreviated SHC[2], is a revocable hill-climbing that permits random shifts to directions with no improvement in some non-zero probability depending on a given temperature, which may help the search escape from local optima.

Up to now, many meta-heuristics exist that give ways to avoid local optima, like restarting with another candidate solution generated randomly[3, 6, 9], adjusting the evaluation function by increasing the weight of unsatisfied constraints[4, 5, 8, 13], introducing a state transition probability to determine the next state[2, 10], and simulated annealing (SA)[2, 11, 14].

SA, modeled after the annealing process of statistical mechanics, is a general-purpose stochastic technique that is effective in approximating global optima for many NP-hard combinatorial problems[2, 12]. Figure 1 shows the SA algorithm to which SHC is incorporated as the basic local search method. SA works on SHC as meta-heuristics by generating monotonically decreasing temperature values, T, which are iteratively used to control the transition probability in the SHC procedure. A temperature decrease corresponds to a narrowing of the search area of SHC from global to local.

As a fundamental result, it is known that SA certainly guides to a global optimum when the temperature is set initially to a large enough value and then reduced logarithmically. However, since logarithmic reduction is too slow for practical use, the decay rate $\gamma(0 < \gamma < 1)$ is generally used instead to control the temperature from T_max to T_min, as shown in Figure 1. Determining the best decay rate γ in advance is difficult because it depends on each problem instance. Further, there is always the risk that SA may freeze before it finds a global optimum when the starting state is not chosen appropriately.

```
procedure simulated annealing(){
  generate a candidate solution, s ;
  for ( T = T_max ; T ≥ T_min ; T := T × γ ) {
    SHC( T ) ;
  };
}

procedure SHC( T ) {
  for ( hc = 1 ; hc ≤ hc-max ; hc := hc + 1){
    calculate the constraint satisfaction ratio ;
    randomly select a variable v with constraint violations ;
    randomly select a value c for v ;
    assign c to v with probability p = 1 / ( 1 + exp( Δ / T ) ;
  };
}
```

Figure 1: The SA algorithm, in which Δ indicates how the number of constraint violations changes by replacing the value of v in s by c.

2.2 Phase Transition of Graph-Colouring Problems

As a well studied NP-complete problem, the graph-colouring problem has often been used to evaluate combinatorial algorithms empirically. We also employ the 3-colouring problem, GCP for short, to test the efficiency of the method that we propose in Section 3. An instance of GCP is defined as a triple (V, C, E), where $V = \{v_1, \cdots, v_n\}$ is a set of variables, $C = \{red, blue, green\}$ is a set of values (different colours) which should be assigned to each variable, and $E = \{e_1, \cdots, e_m\} \subseteq V \times V$ is a set of binary constraints. Notice that (V, E) corresponds to an undirected graph, where V is the set of nodes, and E is the set of edges. An edge $e = (v_p, v_q)$ in E stands for the constraint claiming variables v_p and v_q should not have the same value.

Several recent papers have observed phase transitions: matter commonly undergoes dramatic changes in its qualitative properties when certain parameters pass through particular values[15, 16, 17, 18, 19]. In GCPs also, the solution cost follows an easy-hard-easy pattern[17] as a function of the constraint density, d, which is the ratio of the number of constraints m to the number of variables n. Actually, when the density d is increased gradually, GCPs suddenly become hard to solve in the sense of the computational complexity in the region where d varies from 2 to 3[15]. These surprising phenomena are understood to

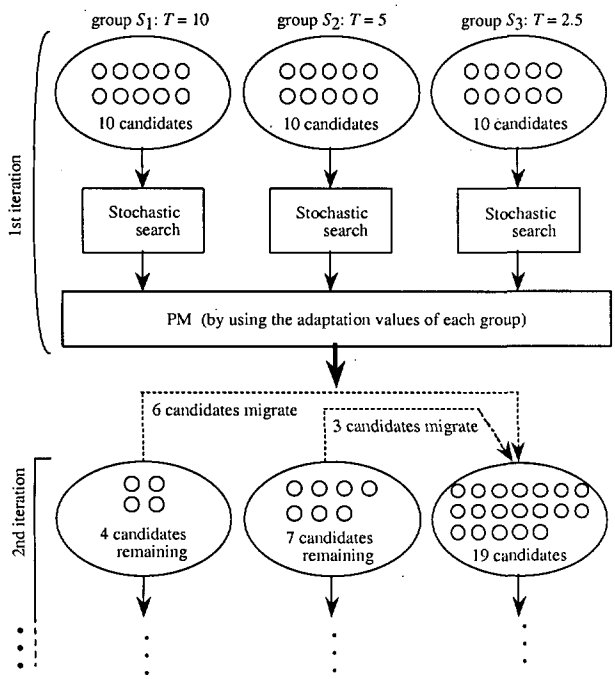


Figure 2: Population migration dynamically updates the allocation of candidates among group.

happen due to the competition between a decreasing number of solutions and an increasing number of prunings[16]. Mammen and Hogg have found another kind of easy-hard-easy pattern which is observed even when the number of solutions is held constant; the pattern in these cases appears to be due to changes in the size of the minimal unsolvable subproblems rather than the changing number of solutions[17].

For any systematic or stochastic method getting complete solutions for critically constrained problems, which are usually found around the transition point, would take an exponential order of computation time. Some known stochastic methods boosted by appropriate meta-heuristics like SA can often find optimal or semi-optimal solutions in an acceptable time. However, most of the meta-heuristics have not necessarily been proposed to cope with phase transitions.

3 Population Migration Strategy — A Meta-Heuristics

3.1 Basic Ideas

Instead of the temporal reduction of temperature in SA, in our method we prepare a set of initial temperature values and for all values a normal stochastic search is performed independently in parallel. We summarize our method in three points as follows:

(i) We determine a set of temperature values that do not change throughout the search process.

(ii) For each temperature, a finite set of candidate solutions is created initially and processed by some basic stochastic search algorithm, SHC in our case.

(iii) The size of each set, called a group, of candidate solutions is adjusted periodically taking its current adaptation value into account.

Periodical tuning of population distribution introduced in (iii) is the key operation and the reason for calling our method population migration, abbreviated as PM. Thus our method can be regarded as a parallel version of SA where the set of available temperatures is fixed in advance. Figure 2 illustrates the mechanism of population tuning. First, three groups, S_1, S_2 and S_3, allotted with the same number, say 10, of candidate solutions, called candidates shortly, are created. To each group a different temperature value is assigned: 10, 5 and 2.5, respectively. For each group, the stochastic search is performed for a predefined period of time, and then a new adaptation value is calculated. In our example, let us assume the updated values in the first iteration become $g(S_1) < g(S_2) < g(S_3)$ and $g(S_2) < \widetilde{g}$, where $g(x)$ is the adaptation value of group x, and $\widetilde{g}$ is the mean value of $g(S_1)$, $g(S_2)$ and $g(S_3)$. At this point PM is started to reorganize the allocation of candidates for the next iteration: as can be seen from Figure 2, a proper number of candidates are moved randomly from the groups with lower adaptation values to the groups with higher ones in proportion to the difference from the mean value $\widetilde{g}$. As a result, PM works as a meta-heuristics that enables implicit self-adaptive temperature scheduling or dynamic control of search ranges.

3.2 The Algorithm

Figure 3 shows an outline of the population migration algorithm, where the meta-heuristics PM is integrated with the stochastic search algorithm SHC. In the following we give supplementary explanations for the numbered statements in Figure 3, assuming GCP as the CSP to be solved.

(1) generate k groups, $S_1, \cdots, S_k$:

As defined in 2.2, let (V, C, E) an instance of GCP with $n = \mid V \mid$ and $m = \mid E \mid$. A candidate, s, is a complete set of assignments of randomly selected values in C for all variables. This statement generates k groups with different temperatures, to each of which the same number of random candidates are allotted.

(2) SHC(T_i):

T_i is the temperature assigned previously to group S_i, which is used by SHC in determining the sigmoidal probability function, shown in Figure 1, to enable stochastic moves to the next candidates. SHC is performed per candidate in S_i. Notice that the whole procedure of Figure 3 terminates whenever a final solution is found during the search of SHC.

(3) calculate the adaptation value $g(S_i)$:

Let $s = (c_1, \cdots, c_n)$ with $c_i \in C$ for $1 \leq i \leq n$ be a candidate. For each constraint $e = (v_p, v_q)$ in E, let $conf(e)$ be equal to 0 when $c_p \neq c_q$, indicating that e is satisfied, or 1 otherwise. Then the constraint satisfaction ratio f of s is given as

$$f(s) = 1 - \frac{\sum_{i=1}^{m} conf(e_i)}{m}. \tag{3.1}$$

Thus, the average ratio $\widetilde{f}$ for S_i is

$$\widetilde{f}(S_i) = \frac{\sum_{s \in S_i} f(s)}{\mid S_i \mid}, \qquad \text{for } 1 \leq i \leq k. \tag{3.2}$$

After the execution of SHC, the adaptation value for each group S_i is calculated as

$$g(S_i) = a \times \widetilde{f}(S_i) + b \times (\widetilde{f}(S_i) - \widetilde{f}_i), \qquad \text{for } 1 \leq i \leq k, \tag{3.3}$$

where a and b are non-negative constants and $\widetilde{f}_i$ is the average satisfaction ratio in the previous iteration of the outermost loop in Figure 3. The second term in (3.3) reflects how much the average ratio $\widetilde{f}$ of the i-th group has been improved by SHC in statement (2) in Figure 3.

(4) divide the groups into two classes, G_{high} and G_{low}:

Let $\widetilde{g}$ be the mean of all the adaptation values calculated as

$$\widetilde{g} = \frac{\sum_{i=1}^{k} g(S_i)}{k}. \tag{3.4}$$

By using $\widetilde{g}$, the groups $S_1, \cdots, S_k$ are classified into two classes as follows:

$$G_{high} = \{S \mid g(S) \geq \widetilde{g}\}, \tag{3.5}$$

$$G_{low} = \{S \mid g(S) < \widetilde{g}\}. \tag{3.6}$$

(5) migrate proper number of candidates :

Let μ be the sum of differences between $\widetilde{g}$ and $g(S)$ for $S \in G_{low}$:

$$\mu = \sum_{S \in G_{low}} (\widetilde{g} - g(S)) \quad \left(\equiv \sum_{S \in G_{high}} (g(S) - \widetilde{g}) \right). \tag{3.7}$$

Population migration is performed from groups in G_{low} to groups in G_{high}. The number of candidates to be removed from group S in G_{low} is determined as

$$\nu = \frac{\widetilde{g} - g(S)}{\mu} \times \mid S \mid, \tag{3.8}$$

except that at least one candidate must remain in S. Each removed candidate determined by (3.8) goes to one of the groups in G_{high}, say S, with probability

```
procedure population-migration(){
    generate k groups, S_1, ..., S_k ;                          (1)
    for ( j = 1 ; j ≤ max ; j := j + 1) {
        for ( i = 1; i ≤ k ; i := i + 1) {
            for (each candidate, s in S_i ) {
                SHC( T_i) ;                                     (2)
            };
            calculate the adaptation value g(S_i) ;             (3)
        };
        PM-meta-heuristics() ;
    };
}

procedure PM-meta-heuristics(){
    divide the groups into two classes, G_high and G_low ;      (4)
    for ( each S in G_low ){
        migrate proper number of candidates;                    (5)
    };
}
```

Figure 3: Population migration algorithm incorporating SHC as the basic local search algorithm.

$$\rho(S) = \frac{g(S) - \widetilde{g}}{\mu}, \tag{3.9}$$

which means that the higher adaptation value a group has, the more candidates the group tends to be allotted.

In the example of Figure 2, $G_{low} = \{S_1, S_2\}$, $G_{high} = \{S_3\}$, $\nu_1 = 6$, $\nu_2 = 3$ and $\rho(S_3) = 1.0$. As a result, population migration dynamically tries to keep an optimal allocation of the limited resources (or candidates) by recruiting promising groups.

4 Experiments

We evaluate the effectiveness of the population migration as a meta-heuristic from two major points of view: we compare its efficiency with SHC and SA, and we investigate in detail its behavior around the critical region where phase transitions may occur. Throughout the experiments, we use solvable GCPs (i.e., graphs colourable with 3 colours) which are generated randomly by using the procedure given in [1]. As to the calculation of the adaptation values of groups defined in equation(3.3), the ratio of coefficient b and a is set to 5 to boost rapid movement of population to the promising groups. All algorithms are implemented in the language C on an IBM Aptiva B75.

4.1 Comparison with SHC and SA

4.1.1 Comparison with SHC

SHC, a naive stochastic search method without heuristics, is adopted to evaluate the efficiency of PM. Fixing the number of nodes n and the number of edges m to 150 and 375 respectively, we generated 100 solvable GCPs. Thus the constraint density $d(= m/n)$ is equal to 2.5, around which it is known that GCPs tend to become hard to solve.

Table 1: Experimental results for SHC, parallel SHC, and the proposed method.

method	SHC with T =						parallel SHC	PM
	0.313	0.625	1.25	2.5	5	10		
%-solved	12	77	99	16	0	0	80	93
time	2.89	2.77	2.08	3.67	–	–	2.78	2.77
σ	1.38	1.10	0.78	1.23	–	–	1.24	1.13

%-solved : percentage of success(%)
time : mean solution time(min)
σ : standard deviation

In our implementation of the PM procedure of Figure 3 we prepared 5 groups with temperatures 10, 5, 2.5, 1.25 and 0.625 respectively and we allotted 20 random candidates are allotted to each group. The PM procedure is performed once per GCP. The maximum number of iterations of the outer loop, max in Figure 3, is set to 100. SHC (statement (2) in Figure 3) performs 100 hill-climbing operations (referred to as hc-steps hereafter) for each candidate as far as a final solution is not attained.

Simple SHC is performed for six different temperatures T: from 10 to 0.313 by halving the value. For each GCP, simple SHC is repeated 100 times with different starting candidates, where the upper limit of total hc-steps is fixed to 0.1×10^5 in each repetition, to make the amount of computation equal to that of PM.

We also tested a mixture of simple SHCs, called parallel SHC, in which 5 simple SHCs with different temperatures are performed in parallel practically under the same conditions as described above except that each simple SHC is repeated only 20 times at most.

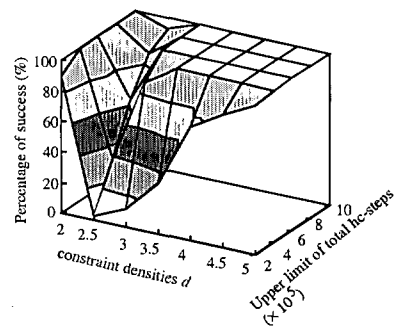
Table 1 summarizes the experimental results, where *%-solved* gives the percentage of solved GCPs, and *time* and σ shows the cpu time averaged over solved cases and its standard deviation.

In comparison with SHCs, PM is a robust method. In fact, simple SHCs do not complete successfully in most cases of temperature except a narrow range near $T = 1.25$, indicating that some mechanism (heuristic) of temperature control is necessary for SHC. Parallel SHC seems to give results comparable to PM from a computational time point of view. However, PM solves much more GCPs within the time limit. Thus, when compared to parallel SHC, PM solves difficult problems without increasing cpu time by help of the population migration.

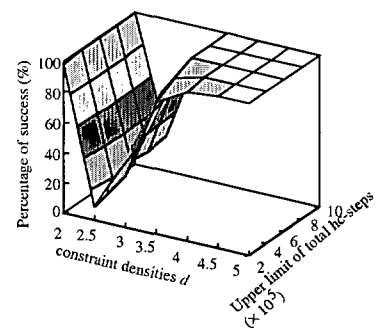
4.1.2 Comparison with SA

In order to compare PM with SA, we ran experimental simulations from two different viewpoints: the constraint density d and the number of variables n.

Let us clarify the PM and the SA used to solve GCPs. The PM is the same as the one used in 4.1.1 except that the



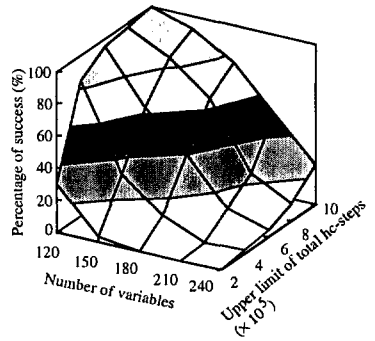
(a) The percentage of success for PM



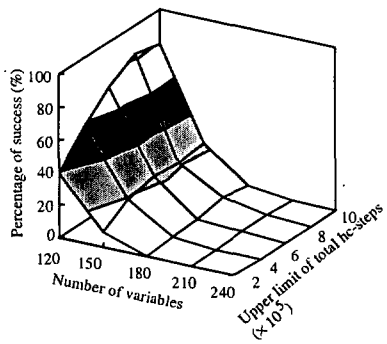
(b) The percentage of success for SA

Figure 4: Experimental results on the constraint densities d.

number max of iterations of the outer loop ranges from 20 to 100 at intervals of 20. Thus the available total computational cost ranges from 2×10^5 to 10×10^5 hc-steps since the hill-climbing effort in every iteration amounts to 0.1×10^5 hc-steps. The SA used in the experiments is based on the SA procedure in Figure 1, where we set $T\_max = 10.0$, $T\_min = 0.625$, $\gamma = 0.5$ and $hc\text{-}max$ in SHC equal to 1,000, in order to make the SA comparable to the PM above. As long as a final solution is not found the SA procedure is repeatedly restarted with a new initial candidate. In fact, we tested five different upper limits on the number of repetitions: 40 to 200 at intervals of 40, which corre-



(a) The percentage of success for PM



(b) The percentage of success for SA

Figure 5: Experimental results on the size of the search space.

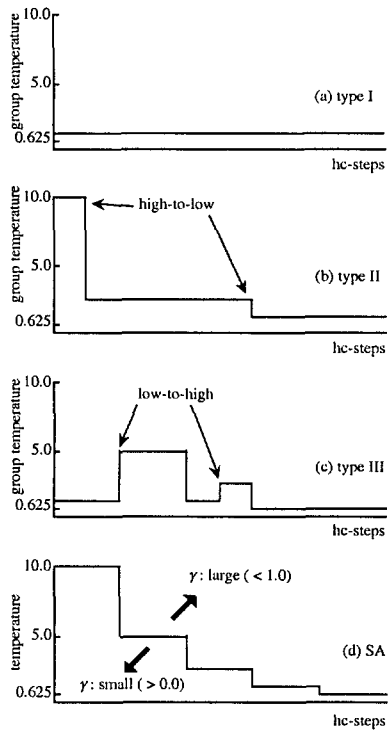


Figure 6: Migration patterns (a), (b) and (c), and SA temperature control (d).

spond to the 2×10^5 to 10×10^5 hc-steps of total computational costs described above.

In the first experiment, we tested seven different densities d varying from 2 to 5 at intervals of 0.5. For each density, we randomly generated 100 GCPs with a fixed number of variables, $n = 150$. Figure 4 gives the results. When the total computational cost limit is low, say less than 4×10^5, SA is slightly superior to PM. But, in the density range 2 to 3.5, where most GCPs are extraordinarily hard to solve, PM apparently gives superior results. In that range, SA fails to solve most of the GCPs and its performance is not improved even when the computational time is increased. In contrast, the percentage of success for PM remarkably increases as the available total hc-steps is increased.

In the second experiment we varied the number n of variables fixing the constraint density d to 2.5. We tested five cases of n: 120 to 240 at the intervals of 30. For each n, we generated 100 random GCPs. Figure 5 shows the results. We can clearly see that PM gives higher success ratios than SA everywhere except in the restricted case that both the problem size and the available computational cost are small. The size of the search space grows exponentially as n increases. As a result, the probability of success for SA declines rapidly and does not seem to improve even when the available computational cost is increased. In the case of PM, however, the percentage of success does decline slowly as n increases when the total computational cost is chosen to be large proportionally to the problem size.

To illustrate the wide applicability and reliability of PM as a meta-heuristics, we ran a set of supplementary experiments in which neural networks are used as the basic stochastic search technique to solve SAT problems. The results are shown in the Appendix.

4.2 Detailed Analysis of Temperature Control

4.2.1 Migration Patterns

We traced the behavior of all the candidates during the execution of the PM procedure. When the PM operation is performed in Figure 3, the population distribution of the groups is updated autonomously. Thus, for each candidate we get a transition pattern along which the candidate migrated among the five groups.

Observing these migration patterns, we found that they can be classified into three types as shown symbolically in Figure 6. Type I is the simplest where the candidate remains in its initial group. Type I corresponds to the simple SHC in which the initial temperature stays unchanged during the search. Type II is the pattern containing high-to-low migrations only: one or more migrations from groups with higher temperatures to groups with lower temperatures. Type II represents the temperature control similar to that of SA, whose typical pattern is given in Figure 6(d).

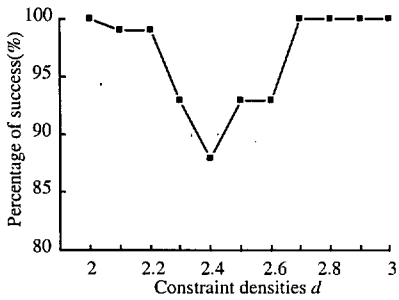


Figure 7: Experimental results from $d = 2$ to $d = 3$.

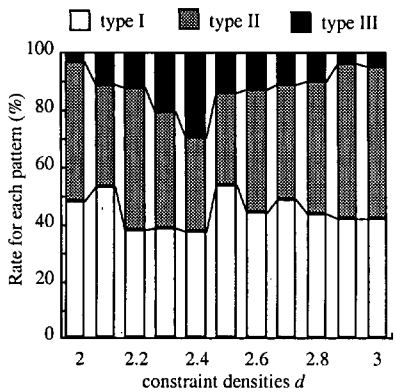


Figure 8: Distributions of temperature behavior in PM.

Various patterns with different decreasing speeds, which correspond to different values of γ in SA, are included in Type II. Therefore, PM is robust because it implicitly performs SA in various cases of γ in parallel. Type III is the pattern containing at least one low-to-high migration, which ensures that population migration enables dynamic self-adaptation control of temperature. In SA the temperature is controlled so that it monotonically decreases. Therefore, Type III is an especially interesting pattern specific to PM.

4.2.2 Discussions from the viewpoint of Phase Transition

The three types of migration patterns (Type I, II and III introduced in the previous section) are similar to the typical cases of temperature control realized by the three major search strategies that we are concerned with: simple SHC, SHC with SA, and SHC with PM, respectively.

To clarify how well these three major types affect the efficiency of the PM procedure, we ran further experiments. We fixed the number n of variables to 150 and varied the constraint density from 2 to 3 at small increments of 0.1. For each density, we tried to solve 100 solvable GCPs by the PM procedure under the same conditions as described in 4.1.1, with the maximum computational power limited to 10×10^5 hc-steps and 100 population migrations.

Experimental results are summarized in Figure 7 and Figure 8. Figure 7 shows the percentage of problems solved within 10×10^5 hc-steps. The percentage of success is low in the region around $d = 2.4$, where the phase transitions are expected to occur[15].

We traced back every candidate that led to a final solution and classified its migration pattern into the three types shown in Figure 6. Figure 8 shows the results. We see that the curve of Type III is quite similar to that of the success percentage in Figure 7. Actually, the percentage of Type III becomes the highest at $d = 2.4$, where hard problems are concentrated. Thus, it is expected that the Type III pattern, which is specific to the PM meta-heuristics, will be helpful to reduce the hardness of GCPs in the critical region.

5 Conclusions

We proposed a novel meta-heuristics named population migration (PM), which is applicable to stochastic search methods for constraint satisfaction problems including stochastic hill-climbing and neural networks.

It may be possible to view PM as a spatially parallel version of temperature control of SA in which temperature always decreases monotonically. The proposed meta-heuristics, however, enables a more sophisticated control of temperature since it implicitly conducts dynamic self-adaptive temperature control. Its effectiveness was verified by some experiments: (1) comparison between naive stochastic hill-climbing (SHC) and SHC assisted by PM, (2) comparison beween SA and PM applied to two basic methods: SHC and neural networks, (3) detailed investigation of the dynamic controllability of temperature from the viewpoint of computational complexity.

The last experiment is particularly interesting because efficiency of self-adaptive temperature control, which is specific to PM, is remarkable in the critical region where phase transitions occur.

Acknowledgment

This research was partially carried out while the second author was staying at the International Institute for Applied Systems Analysis (IIASA) in Laxenburg, Austria as a research scholar of RMP (Risk, Modeling and Policy) Project under the supervision of Dr. Marek Makowski.

References

[1] Minton, S., Johnston, M. D., Philips, A. B., and Laird, P. (1992) Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problem, *Artificial Intelligence*, Vol.58, pp.161–205.

[2] Ackley, D. H. (1987) *A Connectionist machine for Genetic Hillclimbing*, chapter 3, Kluwer Academic Publishers.

[3] Adorf, H. M. and Johnston, M. D. (1990) A Discrete Stochastic Neural Network Algorithm for Constraint Satisfaction Problems, *Proceedings of IJCNN'90*, pp. 917–924.

[4] Boyan, J. A. and Moore, A. W. (1998) Learning Evaluation Functions for Global Optimization and Boolean Satisfiability, *Proceedings of AAAI'98*, pp. 3–10.

[5] Davenport, A., Tsang, E., Wang, C. J., and Zhu, K. (1994) Genet, A Connectionist Architecture for Solving Constraint Satisfaction Problems by Iterative Improvement, *Proceedings of AAAI'94*, pp. 325–330.

[6] Frank, J. (1996) Weighting for Godot: Learning Heuristics for GSAT, *Proceedings of AAAI'96*, pp. 338–343.

[7] Frank, J., Cheeseman, P., and Stutz, J. (1997) When Gravity Fails: Local Search Topology, *Journal of Artificial Intelligence Research*, Vol.7, pp.249–281.

[8] Morris, P. (1993) The Breakout Method for Escaping From Local Minima, *Proceedings of AAAI'93*, pp. 40–45.

[9] Selman, B., Levesque, H., and Mitchell, D. (1992) A New Method for Solving Hard Satisfiability Problems, *Proceedings of AAAI'92*, pp. 440–446.

[10] Selman, B., Kautz, H., and Cohen, B. (1994) Noise Strategies for Improving Local Search, *Proceedings of AAAI'94*, pp. 337–343.

[11] Spears, W. M. (1996) A NN Algorithm for Boolean Satisfiability Problems, *Proceedings of ICNN'96*, pp. 1121–1126.

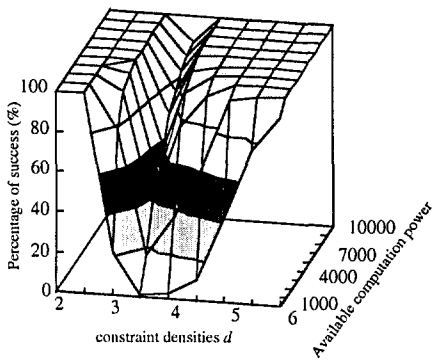
[12] Varanelli, J. M. and Cohoon, J. P. (1995) Population-Oriented Simulated Annealing: A Genetic / Thermodynamic Hybrid Approach to Optimization, *Proceedings of ICGA'95*, pp. 174–181.

[13] Wong, J. H. Y. and Leung, H. F. (1998) Extending genet to solve fuzzy constraint satisfaction problems, *Proceedings of AAAI'98*, pp. 380–385.

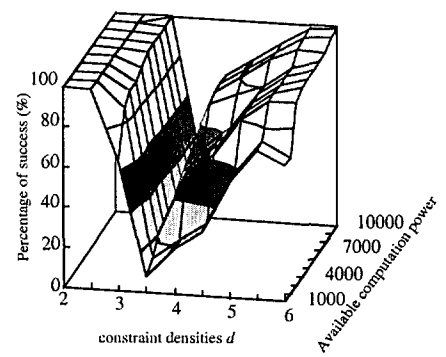
[14] Wah, B. W. and Wang, T. (1999) Simulated Annealing with Asymptotic Convergence for Nonlinear Constrained Global Optimization, *Proceedings of CP'99*, pp. 461–475.

[15] Hogg, T. and Williams, C. P. (1994) The hardest constraint problems: a double phase transition, *Artificial Intelligence*, Vol.69, pp.359–377.

[16] Hogg, T., Huberman, B. A., and Williams, C. P. (1996) Phase transition and search problem, *Artificial Intelligence*, Vol.81, pp.1–15.



(a) The percentage of success for PM



(b) The percentage of success for SA

Figure 9: Experimental results on the constraint densities d.

[17] Mammen, D. L. and Hogg, T. (1997) A New Look at the Easy-Hard-Easy Pattern of Combinatorial Search Difficulty, *Journal of Artificial Intelligence Research*, Vol.7, pp.47–66.

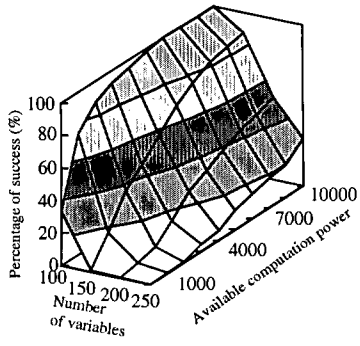
[18] Mitchell, D., Selman, B., and Levesque, H. (1992) Hard and Easy Distributions of SAT Problems, *Proceedings of AAAI'92*, pp. 459–465.

[19] Yokoo, M. (1997) Why Adding More Constraints Makes a Problem Easier for Hill-climbing Algorithms: Analyzing Landscape of CSPs, *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP'97)*, pp. 356–370.

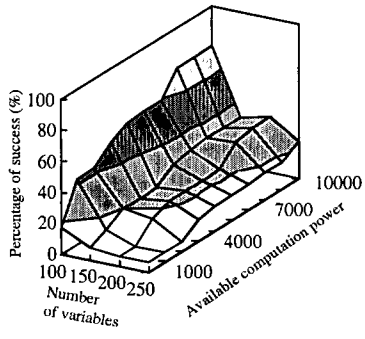
[20] Asahiro, Y., Iwama, K., and Miyano, E. (1996) Random Generation of Test Instances with Controlled Attributes, *DIMACS*, Vol. 26, pp. 377–393.

Appendix

In these supplementary experiments the basic stochastic search method to which the meta-heuristics SA and PM are applied is based on NN-SAT, the neural network proposed in [11] instead of SHC in Section 4. The standard CSP to



(a) The percentage of success for PM



(b) The percentage of success for SA

Figure 10: Experimental results on the size of the search space.

be tested is CNF 3-SAT, or 3-SAT shortly, instead of GCP in Section 4.

Let us clarify the parameters of PM and SA. The PM procedure is the one of Figure 3 except that instead of SHC, NN-SAT without temperature control is used as the basic stochastic search method. The number k of groups is set to five with temperatures fixed to 0.15, 0.08, 0.04, 0.02 and 0.01. To each group 8 candidates are alloted. The SA procedure we used is the same as NN-SAT[11], with the temperature T controlled like $T = T\_max \times \exp(-j/(restarts \times neurons))$ from $T\_max = 0.15$ to $T\_min = 0.01$. The parameters j, $restarts$ and $neurons$ indicate the j-th trial of search operation, the number of restarts with a new candidates, and the number of neurons, respectively. For a SAT problem, n is the number of variables in the propositional expression and d is the constraint density given by the number of disjunctive clauses divided by n. In each case 100 random 3-SAT problems were generated using the procedure in [20].

In the first experiment, fixing the number n of variables to 150, we tested nine cases of constraint density d: 2 to 6 at intervals of 0.5. Figure 9 shows the results. As was seen in Figure 4, PM again becomes apparently superior to SA as the available computational power increases.

In the second experiment, fixing $d = 3.5$, four values of n are tested: from 100 to 250 at intervals of 50. Figure 10 shows the results. We see the percentage of success of SA tends to decline more rapidly than that of PM as the problem size increases, which is similar to the result of Figure 5.

Let us notice finally that all three migration patterns shown in Figure 6 are also observed in these supplementary experiments.

Modeling shipping company information systems

Sonja Klenak, Sanja Bauk
Maritime Faculty, Dobrota 36, 85330 Kotor, Montenegro, Yugoslavia
fzpkotor@cg.yu and www.fzpkotor.cg.yu

Keywords: information system, modeling, shipping company

Received: October 2, 2001

The introduction of up-to-date information technologies in all aspects of business activities, maritime industry included, represents the basic prerequisite of success. Due to extremely difficult situation in maritime industry in most under-developed countries, there are no adequate information systems. Wishing to present the current situation and the possibilities of application of up-to-date scientific methods in projecting and implementing information systems, this paper deals with:

(a) Up-to-date methodological approach to projecting information systems through CSF (Critical success factor), E-M (End- Means), BSP (Business System Planning) methods and their comparative analysis;
(b) Present condition of information systems in shipping companies of most under-developed countries;
(c) Possibilities of modeling the existing systems through determining the business goals and projecting matrix relations between business organizational units, processes, and data classes;
(d) Upgrading of these relations by HIPO (Hierarchy Input Process Output) method, referring to input-output diagrams of certain business processes;
(e) Possible architecture of integrated shipping company information systems;
(f) Review of integrated ship information systems as a key subsystem of a shipping company.

1 Introduction

It becomes evident that maritime countries, which do not find solutions to the challenges of modern changes in information technologies, would find themselves on the way of stagnation and decline. This fact is based on the notion that the introduction of up-to-date information technologies is turning into the major factor of development. The significance of information is shifted from mass usage to the creation of new ideas and it enables a new relationship between a man and the technology he uses.

Information systems of a shipping company are usually organized as a group of more subsystems intensively interconnected. For proper operation of such a system it is of utmost importance for all systems to function by performing tasks defined according to set goals, that there is good communication among the subsystems and that the data and information are rationally used. Therefore, designing such a complicated system is a very complex task that should gather a large number of experts for particular fields in order to obtain a consistent project. This technology requires all working teams to perform their share of work in a homogenous way. Information system design should be done according to the overall project goal.

The integration of management of material, energy, financial and information resources has been made possible by the development of information system. Thus, the basic goal of information system design is to provide management of the above mentioned resources.

Business process may be defined as a group of logically connected tasks to be executed in order to achieve shipping company business results. A group of several processes comprises a business system or method of operation of a business unit or a group of shipping company units. All the processes, shipping company processes included, have two main characteristics:

1. processes have users (internal and external) or specified business outputs received by the users and
2. processes are not constrained by the organisational limits, i.e. they are performed among several organisational units, meaning that they are, in principle, independent of the formal organisational structure of a shipping company.

Typical examples in shipping industry to satisfy these requirements are as follows: development of new methods of business operation and creation of new types of services, ordering carriage as well as creating a marketing plan of a shipping company. Modelling of business processes with the support of information technologies is done mostly in five main steps:

1. creating business and process goals;
2. identifying shipping organisation processes that need to be reprogrammed;
3. understanding and measuring the existing processes;
4. identifying the possibilities of information technology and
5. projecting and designing prototypes of new processes.

2 Model of business processes

Business model of a system includes the application of methods for structural decomposing and the use of a reference model.
System decomposition enables a comprehensive system analysis and, as the next step, its modelling. In doing so, it is necessary to provide information regarding the following:
- the process itself, i.e. horizontal functional relations,
- system structure, i.e. vertical functional relations,
- data flow, i.e. data-functions relations,
- data structure, i.e. vertical data relations.

Structural systems disposition methods should determine the decomposition level and system limits, check input and output data consistency, as well as define precisely the system's general structure, and be applicable to a complex system – such as a shipping company.
CSF, E-M and BSP methods are easy to use, they enable quick interchanges, programming and application, as well as relevant documentation gathering.
Business system model, the kind a shipping company represents, has three basic components:
1. **entities**, i.e. organizational units – being the carriers, participants or partial participants in particular business processes, thus generating or using specific data;
2. **business processes** – groups of logically connected activities and decisions which are used for managing a shipping organization;
3. **data classes** – groups of data used to describe resources, activities and other terms of permanent significance for the business system.

A suitable matrix could present relations between business systems and organizational units, i.e. entities. A matrix like that gives the answer to the following questions: what responsibility for performing particular business processes has specific entities, i.e. organization units? Organizational structure model is subject to change, but what is to be done remains the same. The things that change are operation and management methods, as well as the intensity of work, but the model of business processes has to maintain the independence regarding the change ability of business terms and organizational structure.

Separate entities, i.e. organizational units and business processes are interrelated by three types of relations:
1. Organizational unit has the **primary** responsibility for the implementation of a business process;
2. Organizational unit has the **significant** responsibility for the implementation of a business process
3. Organizational unit has the **marginal** responsibility for the implementation of a business process.

A suitable matrix could also present relations between business processes and data classes. Then separate business processes generate or simply use specific data classes.

Due to its complexity, the relation matrices of business processes, entities and data classes would not be presented as the integral part of the paper, but it is important to stress the fact that they are essential in the process of defining the architecture of an information system.

3 Methodology of projecting

The methodology of projecting information system of a shipping company is rather based on business processes than on the changeable organization structure. Consequently, the following methods have been covered in the paper:
- BSP (Business System Planning)
- E-M (End – Means)
- CSF (Critical Success Factor) and comparative analysis has been made. BPS method was given the preference.

3.1 Critical success factor analysis

Critical success factor analysis, first proposed by Rockart [7], is business-aligning rather than business-impacting technique, based upon three stage process. Firstly the identifying of a number of critical success factors is to be done, secondly the critical decision is to be made and finally the information required to support those decisions are to be defined. This process can be applied at each management level: strategic, tactical or operational one. Critical success factors are those handfull of things that within someone's job must go right for the organization to flourish. The process of critical success factors analysis allows managers, initially senior ones, to articulate their needs in terms of the information that is absolutely critical to them.

Example 1: Problem of ship spare parts stocks optimization.

CSF (Critical Success Factor) - ship spare parts number (quantity) optimization,
KD (Key Decision) - supply optimization and
IR (Information Required) - statement facts, information about damages, suppliers, expenses, etc.

Example 2: Problem of ship route optimization.

CSF (Critical Success Factor) - ships' schedule optimization,
KD (Key Decision) - ship's assignment to a particular line and
IR (Information Required) - information about lines, channels, tariffs, freights, etc.

It is to be identify what is necessary for business' success, how it can be achieved and by what information.

3.1.1 Critical set analysis

Henderson's [2] work on critical sets is variant of CSF analysis which includes analysis of the critical assumptions made as well as the critical success factors and the key decisions. Basically the model suggests that a three-stage process can identify the information system strategy.
These stages are:
1. Understanding the business using,
2. Identifying information needs using and
3. Ranking the information system/information technologies (IS/IT) opportunities.

This approach is very much about business alignment and it is not going to be used as a tool of impacting or business re-engineering. This is an approach for ensuring that information system vision aligns with senior management vision and any such revolutionary steps as process redesign are not likely to be adopted.

3.2 Ends-means analysis

Drawn from the general system theory model of first and second order control systems, this technique requires that managers define not only their information requirements, the outputs, but also measures of efficiency (first order feedback) and effectiveness (second order feedback).
This technique, therefore, aims to identify information requirements. The outputs of one business process form the inputs of another, making explicit the internal customer relationship modeled by Poter's [5] value system, and the organization has to define effectiveness, how good the outputs are at being the next inputs, as well as efficiency, that is the minimum use of resource to perform the move.

The stages in Ends-Means Analysis are as follows:
1. Specify ends,
2. Specify means,
3. Specify efficiency measures (what information is needed to know that the organization is efficient) and
4. Specify effectiveness measures (what information is needed to know that the organization is effective).

This is not particular widely used tool, but it is practical for anyone familiar with system theory, particular control systems. E-M analysis is directly connected with effectiveness and efficiency as bases of control systems. It is particular good for re-engineering or redesigning process of existing IS/IT systems.

3.3 BSP method

BSP was developed initially for IBM internal use and then sold as a service to their customers in the mid-1970's. It is fairly lengthy process that offers a structured approach to planning via number of fairly rigorously defined stages that lead from the identification of business processes to definition of required data structures. The steps needed to conduct BSP method are summarized as follows:
1. starting the study,
2. defining business processes,
3. defining data classes,
4. analyzing current systems support,
5. determining the executive perspective,
6. defining findings and conclusions,
7. defining the information architecture,
8. determining architectural priorities,
9. reviewing information resource management,
10. developing recommendations and action plan,
11. reporting results.

Data are tracked as they flows throughout the organization by the business activity they support or result from. Outputs then become inputs so data use and creation can be mapped for the whole organization. It provides a bottom-up view of information for the organization that take as it basic premise the note that data is corporate resource and so it should be managed from an overall organizational viewpoint. Therefore BSP involves top-down planning (1-5) and bottom-up implementation (6-11).

The process of BSP method implementation in IS projecting can be supported by HIPO (Hierarchy Input Process Output) method. IBM develops this method within IPT (Improved Programming Technique) program. According to this method functions at each level are defined like processes that owing certain inputs give appropriate outputs. The adequate medium, i.e. input/output units could be determining by HIPO overview and detail diagrams. HIPO is usually used in further stages of BSP method, i.e. during the process of its implementation.

3.4 Comparation of CSF, E-M, BSP

The difference between these three methods can be highlighted by examples of following questions:
CSF: What is the CSF for certain business area? What information is required to ensure the CSF is well managed?
E-M: What makes services effective to users? What information is required to ensure effectiveness?
BSP: What are the major problems in accomplishing the purposes of certain business area? What are good solutions for these problems? What role does information play in those solutions?

Where data detail is wanted then BSP can be used. Where business direction focuses is wanted CSF will be applied. E-M analysis may score most highly when the objective is not only to improve current processes, but also to provide their continual monitoring.

4 Architecture of shipping company information systems

Basic aim in conducting this study is to create a model based on the main functions of a business system, so that on the basis of the model transfer to information domain could be performed and the appropriate architecture of the information system defined. In order to reach the model starting from its business system the "top-down" methodological approach is used; then the transfer from business to informational domain is done and accomplishment of particular information sub-systems and the unique information system is achieved through "bottom-up" processing.

4.1 Information sub-systems

Information sub-system is a group of logically rounded business processes and data classes, which is formed around one or more key business resources or entities. Within each sub-system there are several modules which represent a complete group of applications functionally separable from the sub-system, which means that one module could be used for more sub-systems.
Information sub-systems with following modules, which clearly stand out in functioning of a shipping company, are the following:

1. Sub-system Administration
- business politics
- plann and analyze
- ships' bourse tracking
- distribution

2. Sub-system Commerce
- common data
- bourse
- calculation
- contract
- contract realization
- invoice
- report

3. Sub-system Ship (part of Tecnical Deparment IS)
- tecnical characteristics
- certificates
- security system
- inspection and maintenance
- events

4. Sub-system Personnel
- basic data
- seamen
- contracts
- other employee

5. Sub-system Finance
- basic data
- financial operations
- bookkeeping
- financial reports
- currency rates

6. Sub-system Information Bourse
- countries
- ports
- channels
- ship remonts
- brokers
- agents
- producers
- providers
- market
- low regulations
- embassies - consulates
- air companies

7. Sub-system Low and Insurance
- basic data
- low acts
- insurance
- contracts
- complains

8. Sub-system Purchasing
- basic data
- demands (from ships)
- offers
- inventory books
- reports

9. Sub-system Agency
- data related to the links (destinations)
- reservations and tickets delivery
- providing tickets for seamen
- reports

10. Sub-system Business Activities
- stockholders
- share of stockholders
- meetings

11. Sub-system Common Module
- codes
- editors
- connections with other networks, etc.

In order to illustrate the possibility of implementing the information system for the activities of a technical

department of a shipping company, the appropriate application has been prepared in MS Access (Appendix 1).

5 Integral information system of shipping company

On the basis of a defined model of the shipping organization, identified entities, business processes, data classes, their interrelations and separate information modules, insight into an integral information system of the shipping company could be, at least partly, achieved. It is evident that it is a very complex, multi-function system requiring distributed organization, i.e. apart from central data bank and central processor, and it requires the existence of developed functions of separate information sub-systems.
In order to meet these requirements, the necessary operational systems are designed to:
- manage very large, relational, object-oriented data bases,
- create forms for the development of interactive, dynamic Web presentations,
- create reports and graphic applications.

5.1 Integral information system of a ship

For a ship, as a cardinal subsystem of shipping company, appropriate software are developed for integration of particular information systems in integrated one - Ship Control Center (SCC), like SCC GEAMAR developed by STN ATLAS ELECTRONIC. Within ship integrated information system following key modules could be set apart.

1. Ship operations:
- Keeping ship diary
- Technical and financial operations tracking
- Operations decision support
- Demand for compensation support

This module enables:
- ship velocity optimization in relation to revenue/expenses
- ship fuel supply optimization
- data checking
- post optimal analyzes, etc.

2. Ship control:
- Engine control
- Power system control, by:
  - measuring all relevant values for ship operation
  - damages simulations with instructions for its elimination.

3. Ship maintenance:
- Specifications of necessary maintenance (preventive, periodical, corrective and predictive)
- Hull and engine maintenance
- Maintenance and reparation reports
- Inspectory
- Keeping continuity in ship maintenance

4. Ship supply:
- Ship supply circle (demand-calculation- order-invoice-delivery)
- Items grouping and coding process
- Items substitution
- Data integration of all ships under the same administration

5. Crew:
- Identity cards
- Qualifications, licenses, certificates
- Crew schedule program
- Account of wages and expenses

6. Accountancy:
- Financial transactions
- Bookkeeping
- Financial reports
- Comparative financial reports

7. Communications:
- Ship-to-ship
- Ship -to-shore
- Shore-to-ship

Communications ship-shore mean ship conections with fix and mobile users of telecommunications network. Links ship-shore and shore-ship are usually realised by classic radio at shorter distances and almost by satellite links at L-band (1,5-1,6 GHz).

8. Navigation:
Key modules for supplying navigation subsystem are:
- Positioning,
- Route determination,
- Colision avoidance,
- Pilotage, etc.

For the purpose of supporting these modules, following systems are developed:

- Electronic positioning systems: GPS, GLONASS, DGPS, DGLONASS, INMARSAT-3, etc.,
- Electronic charts (ECDIS-Electronic Chart Display Information System) with posibilities of graphic and numeric route planning, processing and memorising route with waypoints, diametres and curves,
- Integrated route control system (TRACKPILOT) that enables automaticaly ship tracking according to predefine course,
- Colision avoidance systems: 3CM, ARPA, RASTER-SCAN (with resolution system

overheaded ARPA), etc. This system not only enable colision avoidance, but even integration of RADAR and electronic chart displays.

All these modules and applications are integrated in unique ship informatin system and by satellite links, coast Earth stations and compatibile softwares with shipping company headquarter (Figure 1).

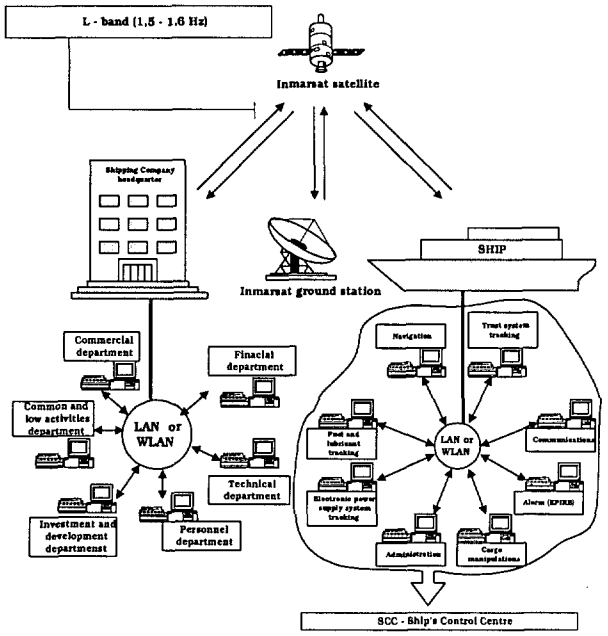


Figure 1: Integral information system of ship and shipping company

6 Conclusion

This paper has attempted to present the possibility of an all-inclusive approach to projecting and implementation of an information system of the shipping company. In the process, the BSP method was preferred in the phase of modeling the business system of a shipping company, i.e. the decomposition of business processes and observing their mutual relations, and those between organizational units and data classes.

Decomposition and defining the inputs and outputs of separate business processes is not enough, since there is the need to define the architecture of the integrated information system. Therefore, the review of information sub-systems forming the integral information system of a shipping company is given. These subsystems should be gathered within distributed information system with central data bank and central processor, but having also the possibility of unobstructed development of data bases and processes for the automation of performing specific business functions.

The aim of information system is to process data generated or used in a shipping company, as well as to use them in transformed form as a back up to the operations of a shipping company and in decision-making aiming at achieving competitiveness in the international maritime trade. Strategic planning of information system should follow the technology development and anticipate business changes, since only the information system, which supports changes, could be of use to a shipping company. Consequently, shipping company management is not only expected to be an *ordering customer*, but also an active participant in information system planning, projecting and implementation.

References

[1] Bauk S., Avramovic Z. (1999) Savremene komunikacije u pomorstvu, *Tehnika 4*, II kongres o saobraćaju, Beograd, pp 262-266.

[2] Henderson J. C, Treacy, M.E. (1986) *Managing End User Computing for Competitive Advantage*, Sloan Management Review.

[3] Klenak S. (2001) *Informaciona tehnologija u funkciji efikasnog upravljanja brodarskom organizacijom*, Maritime faculty, Kotor.

[4] Perovic M., Klenak S. (1999) Primjena informacionih sistema u pomorskom saobraćaju, *Tehnika 4*, II kongres o saobraćaju, Beograd, pp 257-261.

[5] Poter M.E., Technology & Competitive Advantage, *Journal of Business Strategy*, 1985, pp 60-78.

[6] Robson W. (1997) *Strategic Management & Information Systems*, Financial Times, Pitman Publishing.

[7] Rockard J. (1979) Chief Executives Define their own Information Needs, Harvard Business Review

7 Appendix

The following application in Microsoft Access 97 illustrates the possibility for implementing information system for the activities of a technical department of a shipping company. Relations between tables are presented in Figure 1.

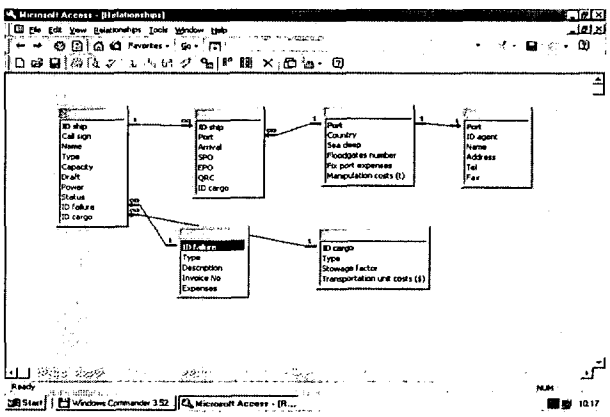


Figure 1: Relationships between tables

The fields and contents of individual tables are given bellow (Table 1-5)

ID	Call sign	Name	Type	Capacity	Draft	Power	Status	ID failure	ID
1	mvyubb	Boka	freighter	35000	25	2500	P		C1
2	mvyuzz	Zeta	freighter	65000	12	4500	P		C2
3	mvyupe	Pelinovo	passenger	25000	12	3500	P		C2
4	mvyuko	Kotor	passenger	34000	13	2500	R	f11	
5	mvyurr	Rumija	freighter	34000	14	3540	P		C1
6	mvyuor	Orjen	freighter	28700	13	2400	R	f11	
7	mvyukm	KapMarti	freighter	34000	10	3570	S		C1
8	mvyuli	Link	passenger	34600	13	2400	S		C2
9	mvyulo	Lovcen	freighter	32000	11	2300	P		C3
10	mvyudu	Durmitor	freighter	34500	13	2890	R	f12	

Table 1. Ship

ID ship	Port	Arrival	SPO	EPO	QRC	ID cargo
1	Rotterdam	12/01/01	10.45	11.55	20000	C1
2	New York	13/01/01	11.3	13.45	15000	C2
3	Bar	14/01/01	12.3	15.25	12000	C2
5	Bar	18/01/01	9.45	11.35	13000	C1
9	Barri	13/01/01	12.45	16.45	12000	C3

Table 2. L/U Operations

Port	Country	Sea deep	Floodgates number	Fix port expenses	Manipulation costs (t)
Amstterdam	Holand	4.5m	16	450	10
Bar	Yugoslavia	3.6m	5	560	12
Barri	Italy	4.5m	13	780	8
New York	New York	5.5m	22	900	9
Rijeka	Hrvatska	3.5m	10	567	7
Rotterdam	Holand	5m	15	890	10

Table 3. Port

ID failure	Type	Description	Invoice No	Expenses
f11	Main engine damage	Cylinder damage	5678	$ 35,000.00
f12	Auxiliary engine damage	Axis damage	3425	$ 15,000.00
f31	Propeller damage	Propeller banding	4356	$ 500.00
f71	Ship hull damage	Hull banging	3214	$ 10,000.00

Table 4. Failure

ID cargo	Type	Stowage factor	Transportation unit costs ($)
C1	bauxite	0.5	12
C2	coal	0.7	10
C3	profile	0.9	13
C4	crude oil	0.6	15
C5	steel	0.8	15

Table 5. Cargo

Some of the queries which are possible to be performed on this basis are given bellow in form of adequate SQL queries:

Query 1: Ship - Failure
Select Ship.Name where Ship.ID failure is "f11" or "f12", and select Failure.Type, Failure.Description and Failure.Expenses.

Query 2: Ship
Select Ship.ID ship, Ship.Call sign, Ship.Name, Ship.Type and Ship.ID failure where Ship.ID failure is Not null.

Query 3: Ship - Failure
Select Ship.ID ship, Ship.Name and Ship.Power where Ship.ID failure is Not null and Failute.Type is like "M*" or "A*".

Query 4: Ship - Failure
Select Ship.Name, Ship.Type, Ship.Capacity and Ship.Draft where Ship.ID failure is Not null and Failure.Expenses are equal or greater than 500 ($).

Query 5: Ship - Failure
Select Ship.ID ship, Ship.Capacity, Ship.Draft, Ship.Power, Ship.ID failure where Ship.Draft is equal or greater then 11 (feets).

Query 6: Ship - L/U Operations
Select Ship.ID ship, Ship.Name and Ship.Status where L/U Operations.Port is Not null and select L/U Operations.Port, L/U Operations.Arrival and L/U Operations.QRC.

Query 7: Port - L/U Operations
Select Port.Port where L/U Operations.Port is Not null and Port.Port is like"H*" or like"Y*".

Query 8: L/U Operations - Cargo
Select L/U Operations.ID ship, L/U Operations.Arrival, L/U Operations.ID cargo where Cargo.Type is like "b*".

Query 9: Port - L/U Operations.
Select Port.Port, Port.Sea deep and Port.Floodgates number where L/U Operations.ID ship is Not null and L/U Operations.Arrival is "*/01/01".

Query 10: L/U Operations - Cargo
Select SumOf(L/U Operations.QRC) group by Cargo.ID cargo and Cargo.Type.

Query 11: Ship - Cargo
Select Ship.Name, Ship.ID cargo, Ship.Capacity, Cargo.Transportation unit costs ($) and Expr: Rvenue: [Capacity]*[Transportation unit costs ($)].

Query 12: Port - L/U Operations
Select L/U Operations.Port, L/U Operations.SPO, L/U Operations.EPO, L/U Operations.QRC, Port.Fix port expenses, Port.Manipulation costs ($/t), Expr:Variable expenses:QRC*[Manipulation costs ($/t)] and Expr: Total port expenses: [Fix port expenses]+[Variable costs].

Query 13: Query 10 - Cargo
Select Cargo.ID cargo, Cargo.Type, Cargo.Stowage factor, Cargo.Transportation unit costs, Query10.SumOf(QRC) and Expr: Expected Revenue: [Transportation unit costs]*[SumOf(QRC)].

Query 14: Query 9 - Ship
Select Query 9.Port, Query 9.Sea deep, Query 9.Floodgates number, Query 9.Arivall, Query 9.ID ship, Ship.Name, Ship.Call sign and Ship.Capacity.

Query 15: Query 8 - Cargo
Select Query 8.ID ship (arrived *at one of ports*), Query 8.Type (of cargo that ship transported), Cargo.Stowage factor and Cargo.Transportation unit costs ($/t).

The outlook of project form of one of mentioned queries (Query 9) in MS Access is represented in Figure 2.

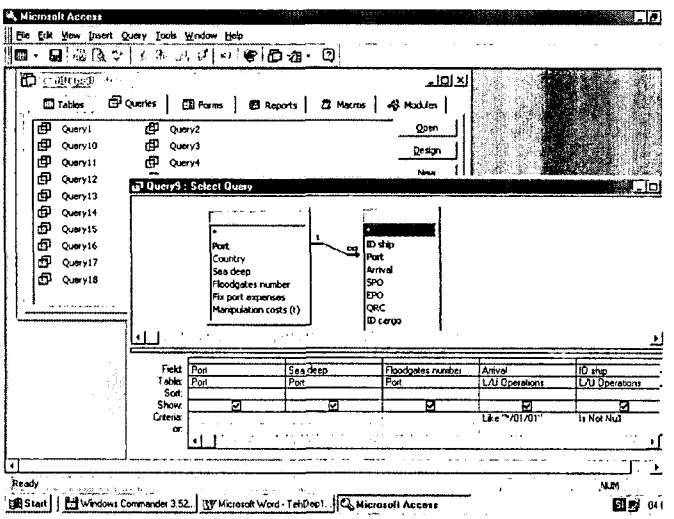


Figure 2: Query 9 - Design form

INFORMATION SOCIETY 2001
INFOS, Cankarjev dom, Ljubljana, Slovenia
22.-26. October 2001

You are kindly invited to cooperate on multi-conference Information Society – IS 2001, which will be held under INFOS from 22nd to 26th of October 2001 in Cankarjev dom in Ljubljana. The multi-conference will include important achievements on the fields mentioned below. Emphasis will be given on the exchange of ideas and particular suggestions, which will be included in the final paper of individual conferences.

IS 2001 exists of nine carefully chosen conferences:

Collaboration and information society
Data mining and warehouses
Development and reingeeniring of information systems
Education in information society
Intelligent systems
Management and information society
Medical and cognitive science
Speech technologies
New information technologies in fine arts

Further information is available at http://is.ijs.si/ or http://ai.ijs.si/is/is2001/index01.html.

Institutions, enterprises and donators are invited to present interesting new developments on their fields of work as 'normal' contributions. They can make a review of new developments and existing situation in their institutions and talk about problems of development in Slovenia, attitude of governmental institutions, and about the way Slovenia should be developing in the direction of information society. They can grant certain interesting activities, related to their work (please turn to the organizator, for example matjaz.gams@ijs.si).
The emphasis is on development, new ideas and trends in information society. If you have something interesting to tell or show to Slovenia, Information Society is the right place to be.

Invited are primarily all those, who have some knowledge about information society. Presentations of enterprises are welcome, especially from the functional point of view. To summarize, we will meet to tell what can we do in Slovenia, to exchange our experiences and to help Slovenia make a step forward in the direction of information society.

You are kindly invited to make a presentation and actively take part in the open exchange of ideas with your knowledge and achievements. The submission deadline is fall 2001.

Pictures from the IS 2000 conference can be found at http://ai.ijs.si/IS/is2000/index00.html.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of **Slove**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

In the last year on the site of the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica LaTeX format and figures in .eps format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

Please, complete the order form and send it to Dr. Rudi Murn, Informatica, Institut Jožef Stefan, Jamova 39, 1111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than five years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (optional): .....

Signature and Date: .....

Informatica WWW:

http://ai.ijs.si/informatica/
http://orca.st.usm.edu/informatica/

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Vladimir Bajič, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszoermenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Pavel Brazdil, Jerzy Brzezinski, Marian Bubak, Davide Bugali, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Netiva Caftori, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, Mel Ó Cinnéide, David Cliff, Maria Cobb, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Deborah Dent, Andrej Dobnikar, Sait Dogru, Peter Dolog, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Jozo Dujmović, Pavol Ďuriš, Amnon Eden, Johann Eder, Hesham El-Rewini, Warren Fergusson, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Hugo de Garis, Eugeniusz Gatnar, James Geller, Michael Georgiopolus, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Inman Harvey, Elke Hochmueller, Jack Hodges, Rod Howell, Tomáš Hruška, Don Huch, Alexey Ippa, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričič, Sabhash Kak, Li-Shan Kang, Ivan Kapustøk, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Kevin Korb, Gilad Koren, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Phil Laplante, Bud Lawson, Ulrike Leopold-Wildburger, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Matija Lokar, Jason Lowder, Kim Teng Lua, Ann Macintosh, Bernardo Magnini, Andrzej Małachowski, Peter Marcer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Daniel Memmi, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Gautam Mitra, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Hari Narayanan, Jerzy Nawrocki, Rance Necaise, Elzbieta Niedzielska, Marian Niedq'zwiedziński, Jaroslav Nieplocha, Roumen Nikolov, Mark Nissen, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, Jens Penberg, William C. Perkins, Warren Persons, Mitja Peruš, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Gabika Polčicová, Gustav Pomberger, James Pomykalski, Dimithu Prasanna, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Luc de Raedt, Ewaryst Rafajlowicz, Sita Ramakrishnan, Wolf Rauch, Peter Rechenberg, Felix Redmill, David Robertson, Marko Robnik, Wilhelm Rossak, Ingrid Russel, A.S.M. Sajeev, Bo Sanden, Vivek Sarin, Iztok Savnik, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer, Zhongzhi Shi, Mária Smolárová, William Spears, Hartmut Stadtler, Olivero Stock, Janusz Stokłosa, Przemyslaw Stpiczyński, Andrej Stritar, Maciej Stroinski, Tomasz Szmuc, Zdzislaw Szyjewski, Jure Šilc, Metod Škarja, Jiří Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, Gheorge Tecuci, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Wieslaw Traczyk, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Zygmunt Vetulani, Olivier de Vel, Valentino Vranić, John Weckert, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Zonling Zhou, Robert Zorc, Anton P. Železnikar

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Volume 25 Number 3 October 2001 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics