Volume 32 Number 1 April 2008 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

1977

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 Higher Education, 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.

Intermediate Representations of Mobile Code

Wolfram Amme and Thomas S. Heinze
Friedrich-Schiller-Universität Jena, Germany
E-mail: {amme,theinze}@informatik.uni-jena.de

Jeffery von Ronne
The University of Texas at San Antonio, USA
E-mail: vonronne@cs.utsa.edu

Overview paper

Keywords: mobile code, intermediate representation

Received: April 3, 2007

Over the past decade, since Java was first introduced and integrated into the Netscape web browser, several intermediate representations have been developed that might be potentially used for mobile code applications. This paper examines the requirements for a mobile code representation, presents several examples of stack-based, tree-oriented, and proof-annotating mobile code representations, and evaluates each of these representations according to the requirements.

Povzetek: Članek podaja pregled mobilnih kod.

1 Introduction

In this era of the Internet, we increasingly come across mobile code applications (i.e., programs that can be sent in a single form to a heterogeneous collection of processors and will then be executed on each of them with the same semantics [1]). Such mobile code is usually intended to be loaded across a network and executed by an interpreter or after dynamic compilation on the target machine.

Unlike traditional monolithic, statically-compiled applications, many modern applications are designed to be dynamically composed from or extended with new components at runtime. An example of this is the Eclipse software development platform [17] that allows new plugins written in Java to be integrated into the environment. This dynamic extensibility is enhanced when the plugins can be described by executable code deployed in a mobile code representation that has a greater compactness, portability, and safety than native binaries.

The Java Virtual Machine's bytecode format ("Java Bytecode") has become the de facto standard for transporting mobile code across the Internet. However, in the last decades several intermediate representations of mobile code have been developed, each of which could be used as an alternative to Java Bytecode. In the paper we give an overview of common intermediate representations, discuss the strengths and weaknesses of each, and finally compare its attributes with that of the other representations.

The intermediate representations designed for mobile code are complex and usually combine multiple features and mechanisms. Therefore, a clear classification of mobile code representations is awkward. In contrast to other surveys, in which mobile code is examined from a programming language perspective [69] and for their verification time [45], respectively, our categorization emphasizes the structure of the intermediate representation. In particular the overview in [69] focuses on several programming languages (Java, Objective Caml, Telescript, etc.) and their suitability in mobile code environments. These languages are not classified by a taxonomy, but are introduced sequentially and evaluated according to some of the requirements imposed by the mobile code setting. In contrast, the article [45] centers on compiling safe mobile code, stressing the importance of safety in the mobile code setting. It discusses the safety issues present in several intermediate representations and compilation techniques, and classifies intermediate representations of mobile code according to their safety checking mechanisms, differentiating static, dynamic and hybrid mechanisms. The static mechanisms check critical safety properties at compile time (e.g., by static program analysis), while dynamic mechanisms rely on runtime safety checks (e.g., by inserting runtime checks into the code). Hybrid mechanisms apply a combination of static and runtime safety checks. In contrast, our classification does not focus on a single aspect (like safety) but highlights the general design of intermediate representations used for mobile code and classifies them as stack-based, tree-oriented, or proof-annotated.

The paper is structured as follows: In Section 2, we introduce a general framework for program transport by means of mobile code, and specify requirements this imposes on intermediate representations of mobile code. Section 3 presents our taxonomy and lists the primary representatives of each category. An evaluation and comparison of these intermediate representations is given in Section 4, and Section 5 concludes with a summary and a discussion about future directions in the area of mobile code representations.

2 Mobile code and its requirements

A system for transporting programs as mobile code can be partitioned into a producer side and a consumer side (see Figure 1). These two components communicate through files containing the mobile code in some intermediate representation (IR). The first step on the producer side is to analyze the input program syntactically and semantically and to transform it into an abstract syntax tree (AST). In the next stage, platform-independent optimizations can be performed, and annotations supporting consumer-side program analysis and optimization may be added to the abstract syntax tree in order to speed up dynamic code generation. Finally, the program is transformed into the chosen intermediate representation, and after being encoded as a—possibly compressed—binary, they are stored into files.

These files containing mobile code are then transferred to the consumer side where they are decoded. Next, the transmitted program has to be examined to determine if it adheres to the security requirements of the mobile code format. This verification process can use a variety of mechanisms ranging from simple type checks to validation of digital signatures or even the verification of proofs about program properties. If no violations are found, the program is executed on the target machine. The execution environment can execute the program by interpreting it or using a just-in-time (JIT) compiler to generate native machine code that runs directly on the target machine. In order to improve performance, JIT compilers often perform machine-dependent optimizations on the program code; this consumer-side optimization is sometimes enhanced by producer-side program annotations. To fulfill the requirements of a mobile code framework, special attention needs to be paid to the choice of intermediate representation. A candidate intermediate representation of mobile code can be evaluated on its ability to satisfy several desirable properties [25]:

Portability: An important property of an intermediate representation of mobile code is high portability. The mobile code needs to be able to execute on different target platforms, so the intermediate representation must be independent from any specific target machine's architecture.

Compactness An intermediate representation should also be dense. Originally, this requirement was due to restricted memory on some of the target code consumers, but today it is more important for reducing transmission times. This property is still critical, especially with respect to dynamically loaded mobile programs.

Flexibility: If an intermediate representation is not bounded to a specific input programming language, it can be used for a wide range of languages. This implies the advantage, as stated in [35], of implementing only *n* code-producers and *m* code-consumers instead of implementing *n***m* compilers. To attain high flexibility, the intermediate representation must support a versatile instruction set and an abstract type model.

Safety: In a mobile code system, the partitioning into code-producer and code-consumer leads to situations, in which the mobile code is not delivered directly by a trusted code-producer. Therefore, the question comes up as to how the code-consumer can ensure that the execution of the mobile code does not maliciously or accidentally affect the local machine in an unauthorized manner. Hence, an appropriate intermediate representation must support verification techniques to ascertain safety properties as type and memory safety.

Efficiency: Finally, although efficiency ultimately depends on the quality of the mobile code system implementation, the intermediate representation of a mobile code system should facilitate, or at least not hinder, the efficient execution of mobile code applications. This property is affected by the way the mobile code is executed: interpreted or compiled just in time. An interpreter-based implementation usually yields lower memory and other resource usage, and is often the most appropriate implementation for embedded systems. An implementation based on just-in-time compilation, however, usually results in faster execution of frequently executed mobile code and also supports machine-dependent program optimizations. Features that make a program representation easier to interpret may make it more difficult to optimize during JIT compilation or vice versa.

An optimal intermediate representation for mobile code should satisfy all of these properties, however in practice, the representation's designer may have to make design decisions that prioritize one over the other. As an example, increases in the safety guarantees often incur a loss of efficiency due to the increased costs of the verification process. Therefore, even though a representation cannot maximize all of these properties simultaneously, a mobile code representation can be evaluated on the basis of how well it satisfies these requirements.

3 Verifiable mobile code representations

Since mobile code is often received through untrusted channels, it is critically important to preserve the mo-

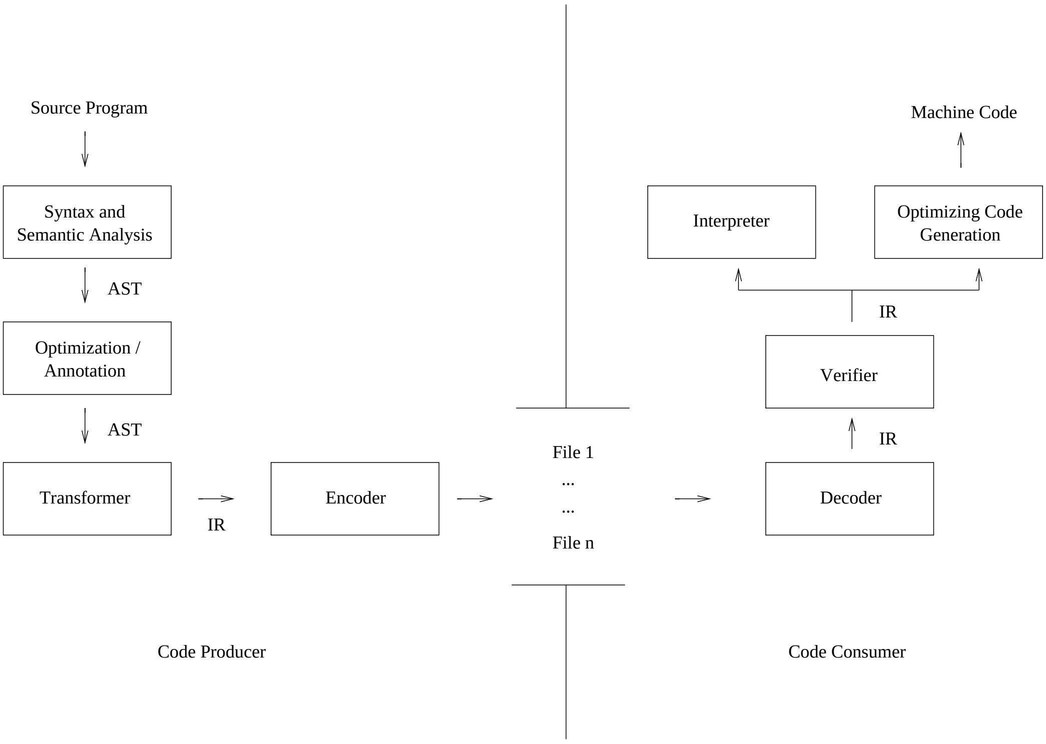


Figure 1: A general system for program transport by means of mobile code.

bile code consuming host system's security in the presence of malicious code. There are three main strategies that have been used—sometimes in isolation, sometimes in combination—to address this risk: cryptographic authentication, sand-boxing, and verification.

The first strategy is to use cryptographic signatures to authenticate the mobile code's producer and to prevent the mobile code from being tampered with during transit from the code producer. The code consuming system can then make decisions about the execution of the mobile code based on the trustworthiness of the code producer. In the simplest form (e.g., Microsoft's ActiveX Controls [53]), this may simply be used to run or not to run the mobile code. In more complex situations, this is used in combination with a security policy and "sand-boxing" to prevent mobile code from performing unauthorized actions.

A second strategy is to create a "sandbox" around the executing mobile code and mediate all access to parts of the code consuming system outside of the sandbox. This allows the sandbox to prohibit those interactions that violate the code consuming system's security policies. This sandboxing can be implemented using operating-system level (e.g., VMWare, Xen, User Mode Linux) or process-level isolation (e.g., BSD jail, SELinux), but these techniques are too heavy-weight and too loosely integrated for use in many mobile code applications (e.g., a Java applet running on a cell phone). These problems can be addressed by using light-weight, fine-grain isolation integrated into the execution environment.

A third strategy, often used to implement fine-grain isolation, is to analyze the mobile code and reject code that violates certain safety properties. Most commonly, this "verification" checks that the code is syntactically correct, has legal control flow, and that it is correctly typed. If the mobile code representation itself is type safe, this will guarantee the possible behavior (especially, with respect to memory accesses) of the mobile code to be constrained by the underlying mobile code representation's type system. This in turn will allow the execution environment (e.g., the Java Virtual Machine) to sandbox mobile code components without necessitating the runtime overhead of operating system or process level techniques.

Successful implementation of this third strategy requires that the program representation is designed with verification in mind. The verifiable mobile code representations that have been used in mobile code frameworks can be classified as being stack-based, tree-oriented, or proof-annotated representations.

3.1 Stack-based types

Most mobile code systems are based upon virtual machines. In such mobile code systems, programs are translated not into machine code for a specific target machine but rather into a platform independent intermediate representation. This intermediate representation consists of instructions for an idealized "virtual" machine.

Code consumers simulate the virtual machine by interpreting the transmitted intermediate representation or by

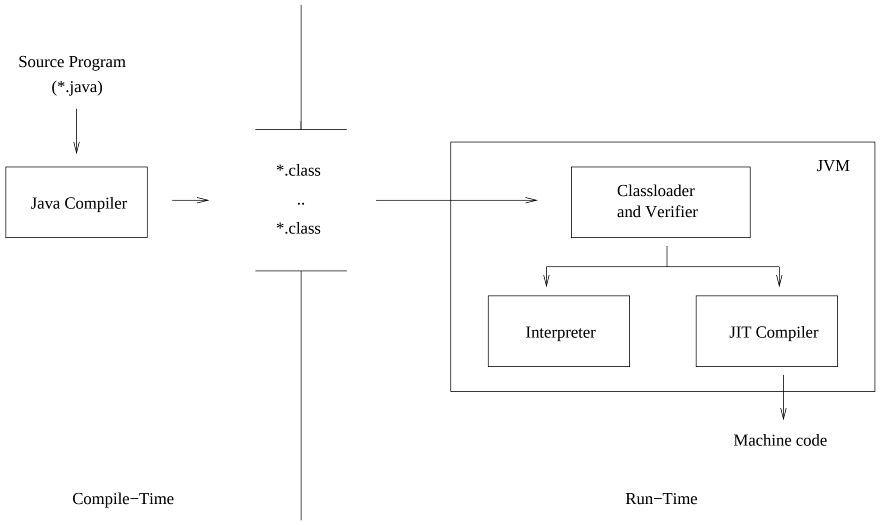


Figure 2: Java language infrastructure.

compiling it into equivalent machine code. Most often the virtual machine utilizes a stack-based architecture. In these virtual machines, most instructions implicitly take most of their operands from a stack and store most of their results back onto the same stack. One advantage of this architecture is the compact encoding of instructions; since most operands are implicit, many instructions can be represented by a single opcode without any operands. These representations are often designed so that each instruction can be encoded using a single byte, and for this reason the instruction sets of stack-based virtual machines are often called bytecode.

Virtual machines have long been used in compiler construction. Starting in the 1970's, compilers have used this concept to organize machine-dependent and machine-independent phases into front end and back end of a compiler. A representative example is P-Code [60], a stack-based intermediate representation used in some Pascal compilers. In the 1990's, *Sun MicroSystems* revived interest in stack-based virtual machines with the Java programming language and its portable intermediate representation, Java Bytecode [34, 33]. Microsoft's .NET Framework [63] also uses a stack-based intermediate representation called the Common Intermediate Language.

3.1.1 Java bytecode

Java Bytecode is a stack-based intermediate representation that was developed as type-safe program representation for Java programs. The instruction set and data types of Java's Virtual Machine (JVM)[1] are designed specifically for the Java programming language. In principle, Java Bytecode can also be used for other programming languages, but field reports show that a use of Java Bytecode for languages other than Java often can cause problems [23].

The architecture for the typical deployment of Java as a mobile code system is given in Figure 2. On the producer-side of this system, a compiler translates a source program into portable Java Bytecode representing each Java method; all the methods in each class (with associated symbolic information) are stored together in a Java "class file." After successful transmission, the consumer-side JVM verifies the code to determine if it is safe to execute the mobile program. If the verification succeeds, the Java Bytecode is interpreted or executed directly after JIT compilation into machine code from the Java Bytecode.

The JVM's most important components are a runtime stack, a program counter, and heap storage, which store objects, code segments, and symbolic information. If a method is invoked, a new method frame is created and placed by the JVM onto the top of the runtime stack. This method frame contains the values of parameters and local variables as well as information about the caller. In addition, each frame contains an operand stack which is accessed as Java Bytecode instructions need input operands and produce output results. Each slot of the operand stack can hold a 32-bit word, and two slots are needed for long or double values.

Figure 3(b) depicts the Java Bytecode program generated for a simple source code. In the program, the contents of local variables *a* and *b* are added and its result afterwards is stored in local variable *c*. JVM assigns indices[2] to all parameters and local variables within a method. These indices are used instead of their symbolic names to reference local variables and parameters. The density of Java Bytecode is increased by the inclusion of instructions which implicitly encode the indices of the first variables defined in a method.

In the sample program, the instructions iload_1 and

[1] A detailed description of JVM is given in [49].

[2] The index 0 is reserved for the object reference in case of virtual methods.

```
(a)
int a,b,c;
    c = a + b;

(b)

iload_1    ; push local variable a onto stack
iload_2    ; push local variable b onto stack
iadd       ; add topmost stack elements
istore_3   ; store topmost stack element into local variable c

(c)

ldloc.1    ; push local variable a onto stack
ldloc.2    ; push local variable b onto stack
add        ; add topmost stack elements
stloc.3    ; store topmost stack element into local variable c
```

Figure 3: Java Bytecode (b) and CIL Bytecode (c) for a simple program (a).

iload_2 are used to push the values of *a* and *b* onto the operand stack. In contrast, the instruction istore_3 takes the topmost element from the operand stack and stores its value in variable *c*. Most of the bytecode instructions are typed (i.e., only accept operands of a specific type). The operations that start with *i* generally indicate that they only accept values of type *int* as operands. Thus, in the example program, instruction *iadd* takes the two top-most int values from the operand stack, adds them, and stores the result back on the top of the operand stack.

The primary design consideration during the development of the JVM was its usefulness as a runtime environment for Java. Therefore, the JVM's instruction set is specialized for the representation of Java programs. Java Bytecode supports four different method invocation instructions implementing the virtual, super, static, and interface method calls of the Java programming language. For each method call, parameters are passed by value only, reference parameters are not supported directly. The JVM's flexibility with respect to running programs written in other languages is also limited by the JVM's provision of only single-inheritance for classes and multiple-inheritance for interfaces, respectively. Another disadvantage is the absence of arithmetic exceptions beside the division-by-zero exception for integers.

Java Bytecode's verification process includes static and dynamic checks and basically operates in four separate passes:

- Examination of general class file format
- Examination of additional structural properties of the class file
- Verification of the bytecode for each method
- Verification of inter-class dependencies during the execution of particular bytecode instructions

The examination of the transmitted Java Bytecode method (3rd pass) is performed by a data flow analysis, which verifies that certain behaviors, which might violate the virtual machine's type discipline (e.g., operand stack over- and underflows, unequal sizes of the operand stack on different control paths, usage of uninitialized local variables, operands of incorrect types for the operation) cannot occur. Overall, the data flow analysis is quite complex and requires, in the worst case, quadratic time in the number of verified instructions[66].

In contrast to the first three passes (which are performed during loading and linking process), the last verification pass, which checks properties about external classes referred to by bytecode instructions, occurs at runtime. In principle, all of the properties that are checked during this pass (e.g., that the classes referred to by an instruction exists) could be performed also during pass 3. But the JVM specification allows these checks to be deferred until run time, so that the loading of additional classes can be deferred until the instructions that refer to these additional classes need to be executed. If one of these dynamic check fails, the execution of the instruction being checked is aborted and an exception is thrown.

3.1.2 Common intermediate language

Microsoft Corporation's Common Language Infrastructure Platform (CLI) is a runtime environment that has been developed for running applications written in several different programming languages, including C#. CLI includes a stack-based virtual machine, called the Common Language Runtime (CLR), which can be used for execution of bytecode programs written in Common Intermediate Language (CIL). In contrast to JVM, the CLR standard does not anticipate execution with interpreter, but rather assumes all applications will be executed using JIT or ahead-of-time compilation.[3]

The .NET-Framework is Microsoft's proprietary implementation and extension of the CLI. In its current version,

[3] Mono, the CLR (ECMA-335) implementation from Novell, however, does include an interpreter [16].

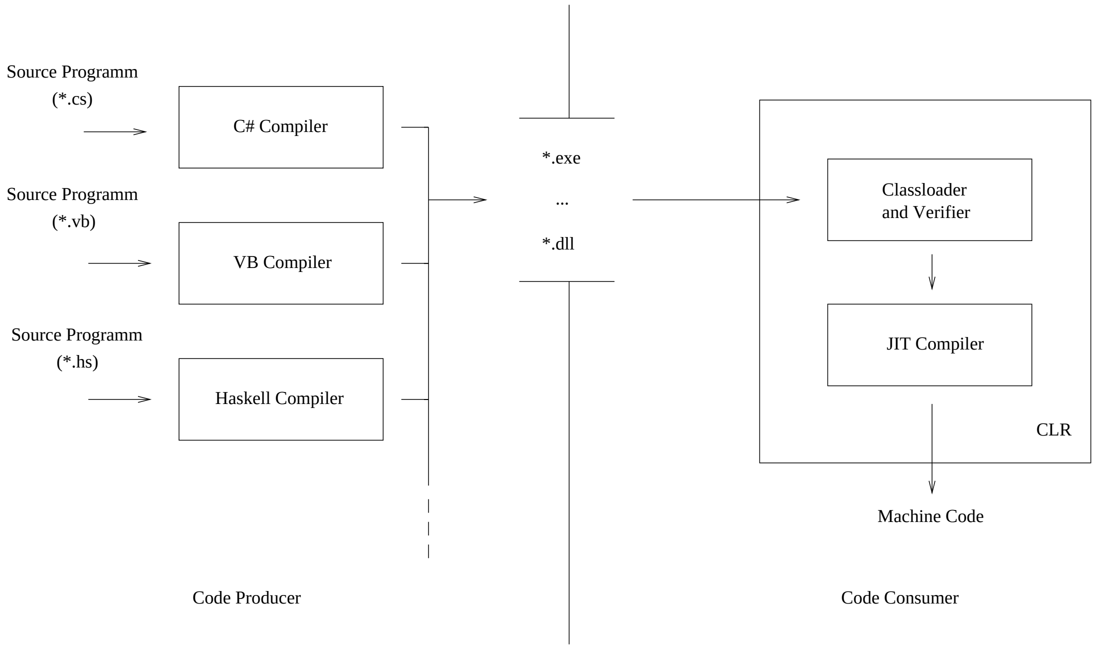


Figure 4: .NET framework.

.NET uses two JIT compilers: A standard JIT compiler and Econo. .NET's standard JIT compiler is an optimizing compiler that supports several optimizations (e.g., constant propagation, method-inlining, common subexpression elimination). In contrast, Econo is a non-optimizing JIT compiler that requires few system resources and, therefore, is especially suited for deployment on mobile platforms with limited resources. In addition, in the .NET-Framework, programs (or parts of programs) may be compiled in advance by the Pre-JIT compiler. Programs, that have been compiled with this compiler, are stored on the file system permanently, so that they can be executed directly when needed in the future without needing to be recompiled at runtime by the JIT compilation.

For each method invocation the CLR creates a new activation record. An activation record consists of fields containing method information, an instruction pointer, arrays for local variable and parameter definitions, and an evaluation stack. The stack-architecture of the Common Language Runtime is realized by the evaluation stack, which is used like the operand stack of the JVM to store the operands and the results of CIL instructions. In contrast to the JVM operand stack, the CLR evaluation stack is capable of storing elements of variable size.

Figure 3 (c) shows the CIL bytecode generated for the sample program from (a). Similar to local variable access in Java Bytecode, local variable and parameter accesses in CIL occur through indices assigned to variables and parameters in the order of their declaration. In the example program, instruction *ldloc* is used to push the value of a variable onto the evaluation stack. In contrast, instruction *stloc* takes the topmost element of the evaluation stack and stores it in a variable. For each variable, there is a corresponding load instruction and a corresponding store instruction to access that variable. These instructions are named by adding the variable number as a suffix (e.g. .1, .2, and .3,) to the operation name. In the example program, instruction *stloc.2* stores the topmost stack element in the local variable *b* (i.e., the local variable with associated index 2.) Unlike Java Bytecode, CIL offers the developer typed and untyped instructions. In the example program, the generic *add-operation* is used. Uses of this generic add-instruction require the CLR to infer the type of add-instructions during JIT compilation from its actual operand types.

In contrast to JVM, the CLI was developed with the intent of supporting many different programming languages. Therefore, the instruction set of CLR is designed around a general type system that is called Common Type System (CTS). Beside the standard primitive and reference types found in Java, the CTS also includes value types. A value type is essentially a restricted class, that is similar to a structure or enumeration type. Like the Java Virtual Machine, the CTS offers only single-inheritance of classes but multiple-inheritance of interfaces. The flexibility of this type model is further enhanced by the instruction set of the CLR, which includes several instructions to make the execution of programming languages other than C# more efficient. For example, a .tail suffix can be appended to a method call instruction, causing it to discard the stack frame of the calling method; this is particularly important for the efficient implementation of functional languages, which make heavy use of recursive calls that would otherwise overflow the runtime stack.

For method invocations there are two call instructions (*callvirt* and *call*) that can be used for virtual, non-virtual and static method calls. For parameter passing CLI offers call-by-reference and call-by-value mechanisms. In addition, parameters can be characterized as result parameters. Standard exception handling for operations on primitive data types are supported only for integer null-division. However, in contrast to JVM, in CLI add-, sub- and mult-instructions can be extended with special postfix operands to handle overflow exceptions.

When the producer side of the mobile code system translates source programs into CIL, it packages them into "assemblies." An assembly contains a set of modules bundled together along with meta-data describing the classes and types defined in and used by those modules [50]. In contrast to Java Bytecode's class files, which contain only a single Java class, a CIL assembly is able to contain several classes. This facilitates the composition of application programs out of multi-module components and allows the producer-side compiler greater scope for inter-class and inter-procedural optimizations. The code within the modules provides sequences of Common Intermediate Language instructions defining the behavior of the methods declared in the assembly.

The CLR uses a verification process, similar to that of the JVM, to determine if it is safe to execute CIL programs. Unlike the JVM, the CLR can be configured to allow certain programs to use "unmanaged" instructions, which can break the type safety of the runtime environment. These are provided in order to support a wide range of programming languages, including languages with unsafe features like pointer arithmetic. Normally, these unsafe instructions would be disabled when running mobile code.

The verification process is performed in two passes: validation and verification. In the validation pass, the general assembly format and the proper use of the meta-data format is ensured. Therefore, the validation pass corresponds to the first two passes of the Java Bytecode verification. In addition, a successful validation is a prerequisite for the verification pass, which is used to verify the control flow and then type-check the CIL module. This verification pass mirrors the last two passes of the Java Bytecode Verification and uses similar mechanisms.

3.2 Tree-oriented representations

Many compilers translate source programs into intermediate representations based on abstract syntax trees. Tree-oriented mobile code representations are derived from these internal data structures, linearized into a stream of binary data so that they can be transmitted in files or across the network. Due to their close relationship to internal compiler structures, tree-oriented intermediate representations are especially well-adapted to execution through JIT compilation, but they can also be interpreted.

A typical tree-oriented mobile code representations compilation unit consists of a source module's abstract syntax tree and symbol table of a program (which would typically be generated during the compilation of the source program even if native machine code were to be targeted) [12, 29, 39, 28]. Since abstract syntax trees are typically machine-independent, tree-oriented intermediate representations are often very portable. In addition, the semantic gap between source language and mobile code representation is minimized compared to a translation into stack-oriented bytecode [66]. The advantages of this approach include the retention of high-level program information (e.g., types and control structures), that can be useful for program optimizations, and a verification process that more closely resembles the type-checking of the source language. The primary disadvantage of this approach is that because it is closely tied to a single source language, it tends not to be very flexible with respect to supporting other source languages.

Though not a true mobile code representation (since it does not address network transportation or verifiability), the Architecture Neutral Distribution Format (ANDF) demonstrates the portability benefits of platform-independent tree-oriented program representations. The compact tree-oriented representation, Slim Binaries, demonstrated the viability of transporting mobile code applets over networks using a tree oriented rather than a stack-oriented representation (like Java Bytecode). The SafeTSA representation is a hybrid representation that combines tree-oriented control structures with blocks of instructions in static single assignment form, which is commonly used as an intermediate representation of the back end of optimizing compilers.

3.2.1 ANDF

The Open Software Foundation's Architecture Neutral Distribution Format (ANDF) [61] was a subset of the Ten15 Distribution Format (TDF)[4] developed by the Defense Research Agency in the UK (DRA). TDF [13] is a tree structured language, that is defined as a multi-sorted abstract algebra. It was originally designed for the compilation of sequential languages such as C and Lisp.

The intended usage was that programs would be distributed in the ANDF, and then compiled into native code at installation time. As such, ANDF was designed solely as a distribution format with a tree-oriented program representation that supports several source programming languages.

Inside of the ANDF infrastructure (see Figure 5), the producer-side translates a program to distribute into ANDF, expressing platform specific information by standard application programming interfaces (API's) [8]. Thereafter, the generated ANDF program is encoded into files and transmitted to the consumer-side, called installer. To install the transferred program, the ANDF files are compiled into target platform's machine code, and the installer replaces calls to an API with implementations provided by the target platform. Although originally developed for the C language,

[4] later renamed to the TenDRA Distribution Format.

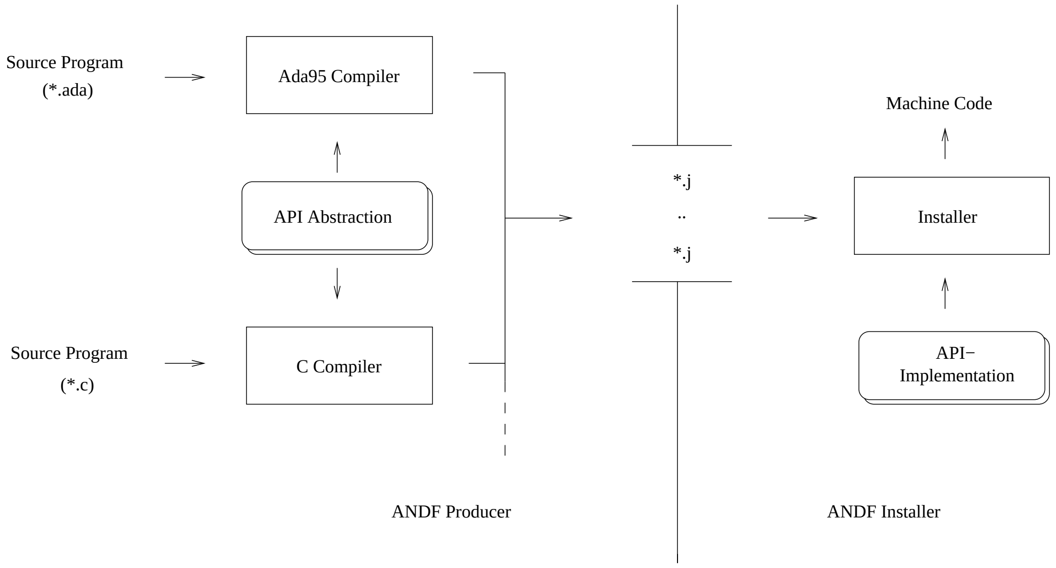


Figure 5: ANDF Scenario.

```
(a) int i,j;

       i = i + 1;
       j = j + 1;
       if (i <= j)
          i = i + 1;
       else
          i = i - 1;
       j = j + 1;

(b) sequence(
      assign(
         obtain_tag(~tag_1),
         plus(contents(integer(~signed_int*),obtain_tag(~tag_1)),
              make_int(~signed_int*,1))),
      assign(
         obtain_tag(~tag_2),
         plus(contents(integer(~signed_int*),obtain_tag(~tag_2)),
              make_int(~signed_int*,1))),
      conditional(
         ~label_0,
         sequence(
           integer_test(
              less_than_or_equal,
              ~label_0,
              contents(integer(~signed_int*),obtain_tag(~tag_1)),
              contents(integer(~signed_int*),obtain_tag(~tag_2))),
           assign(
              obtain_tag(~tag_1),
              plus(contents(integer(~signed_int*),obtain_tag(~tag_1)),
                   make_int(~signed_int*,1))),
           assign(
              obtain_tag(~tag_1),
              minus(contents(integer(~signed_int*),obtain_tag(~tag_1)),
                    make_int(~signed_int*,1))))),
      assign(
         obtain_tag(~tag_2),
         plus(contents(integer(~signed_int*),obtain_tag(~tag_2)),
              make_int(~signed_int*,1))))
```

Figure 6: A sample program (a) and its ANDF output (b).

ANDF producers and installers are available for other programming languages and several machine architectures [9].

In TDF, the original program structure is maintained within the intermediate representation. The base element of the TDF is the sort constructor. Instances of this constructor represent abstractions of expressions, descriptors, and data types. The shape constructor is used to describe data types within the TDF, including procedures, pointers, and recursive data types beside the primitive data types. Generic types can be defined in order to support platform specific data types, like the native integer type. Other sort constructors can be used to define specific memory layouts for data structures, exception handlers, and runtime stacks.

TDF includes various operations, which can be separated into arithmetic, memory, pointer, and control flow operations. Each operation is described using the expression constructor. Figure 6 (b) shows some simplified ANDF output for the example program given in (a). Descriptors (e.g., variables) in TDF are defined by the tag constructor. In this ANDF sequence, a unique integer is assigned to each tag constructor, in which *tag_1* stands for variable *i* and *tag_2* describes variable *j*.

Platform independence is achieved in TDF through the provision of two constructs: the token constructor and the conditional constructor. The token constructor is essentially a parameterized placeholder, which can be replaced with an arbitrary sort constructor. Therefore, the token constructor is used within TDF to hide platform specific program information by substituting calls to an API. In addition, the conditional variant of several constructors allows one to specify platform specific installation tasks. A conditional constructor includes two constructors and a condition: the installer evaluates the condition and maintains one of the constructors, corresponding to the result.

In general, the installation process of ANDF programs is separated into two steps. In the first step, calls to API's, denoted by token constructors, are replaced with its corresponding implementation. In the second step, conditional constructors of the program are evaluated. As a result, the platform independent ANDF program is transformed on the consumer-side into a platform dependent ANDF program, which then is compiled into the machine code of the target platform and installed.

For the transport of ANDF programs, the algebraic TDF is linearized and stored in a *capsule file*. A capsule file consists of a byte array structured into sections. The first section includes the definitions of visibility rules for the encoded ANDF program and acts as an interface. All of the token constructors used in the capsule are specified in the next section with the definitions of token constructors following their declarations in order to simplify the encoding and decoding process. After all the token constructors have been specified, the program is stored in the following sections using a linearized version of its TDF representation. A capsule file normally contains a single program, but it is possible to merge several capsule files into a single capsule library.

Verification of capsule files by the installer on the consumer-side is not integrated into the ANDF scenario, due to its development as a program distribution format. Instead, ANDF producers and installers are validated with respect to their conformity to the ANDF specification during a certification process [8]. For certification, ANDF producers and installers are validated separately. Validation of an installer is based on a number of hand-written programs (i.e., the ANDF Validation Suite [40]), which must be executed accurately by an ANDF installer. Validation of a producer is more difficult, because the produced ANDF code must execute correctly in any runtime environment for which there is an ANDF installer. Therefore, an architecture-independent high-level interpreter is used to evaluate the correctness of ANDF code generated by an ANDF producer for the ANDF Validation Suite.

3.2.2 Slim binaries

The Slim Binary format [30, 21, 22] was originally developed as an extension of the modular Oberon system, in which this format was used to provide architecture-independent distribution of Oberon modules. The name Slim Binaries was chosen to contrast with that of Fat Binaries [46], a name used for commercial distribution formats from Apple and Next, which stored binaries for multiple program architectures in a single file. Since Fat Binaries store one version of the entire program executable for each machine architecture, Fat Binaries tend to be large and require a complex build process [28].

Slim Binaries avoid these disadvantages by using a portable and high-level intermediate representation, that is based on the encoded abstract syntax tree and symbol table of a program. In the extended Oberon system (see Figure 7), the producer-side translates Oberon modules into Slim Binary files and distributes them to several consumers. After successful transmission, a code-consumer can restore the syntax tree and symbol table from the r Slim Binaries and then verify its correctness. If this verification succeeds the syntax tree and symbol table are then used to generate the machine code of the target platform. In the actual implementation, a single code generating loader decodes Slim Binary files and generates code in an unified process.

The program representation contained in Slim Binary files consists of a compact description of the symbol table and a syntax-oriented encoding of the abstract syntax tree that is based on a technique called Semantic Dictionary Encoding (SDE).[5] In SDE the encoding is performed using a dynamically generated semantic dictionary table, in which each entry stands for a special type of node used in the abstract syntax tree. As a consequence, the abstract syntax tree of a program in Slim Binary format is not described through nodes directly, but through a sequence of indices, where each index stands for an entry in the dictionary table. The resulting sequence of indices is stored, conjoined with

[5] In principle, SDE is a clever application of the well-known LZW compression algorithm [71] on expressions.

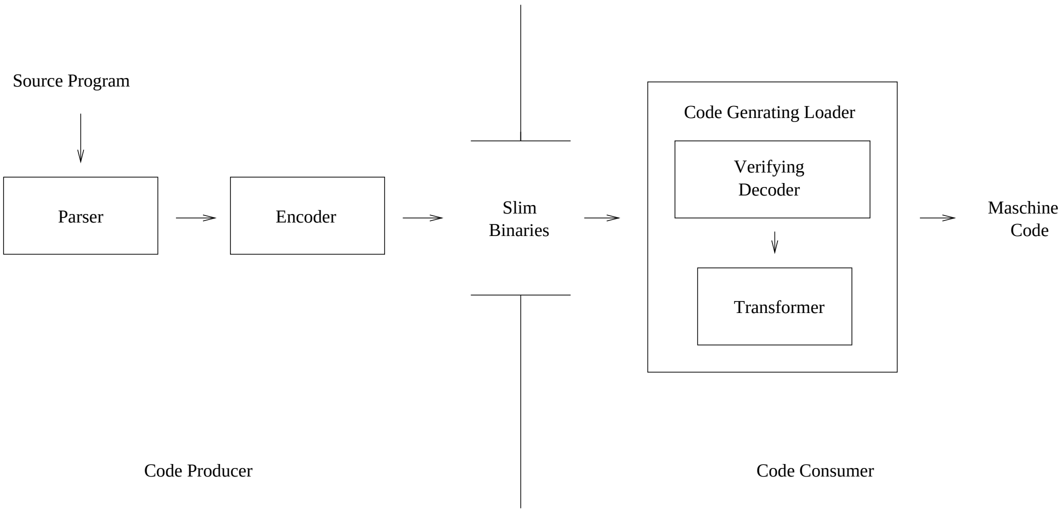


Figure 7: A Slim Binary Scenario.

the symbol table, in a file, which then can be transmitted to the code-consumer.

In SDE the dictionary table is generated in the exact same manner during encoding and decoding processes, therefore it is not necessary to store the dictionary table itself into a Slim Binary file. Instead, the dictionary table of a program is rebuild automatically during decoding of the abstract syntax tree on the consumer-side. Construction of the dictionary table is always performed in three steps. First, the dictionary table is filled up with entries that describe the control structures and operators (e.g., *if*, *while*, *for*, +, –, *, and /) of the used programming language. Second, the dictionary is augmented with entries from the symbol table for the variables and constants defined in the program. Finally, the dictionary can be enhanced with special entries, which we will describe in detail later.

Figure 8 (a) contains the abstract syntax tree and the corresponding symbol table for the same sample source program that was shown in Figure 6 (a), and Figure 8 (b) shows the sequence of indices resulting from applying the SDE. To simplify matters, the dictionary table shows only those entries which appear in the abstract syntax tree. In actuality, in order to describe all control structures and operators significantly more entries must be placed into the table. In SDE, a '.' stands for operands that have not yet been processed, e.g. if the entry '.=.' is selected, the left and right operands will need to be read. There are also dictionary entries (8 to 10 in our example) for each of the entries in the symbol table.

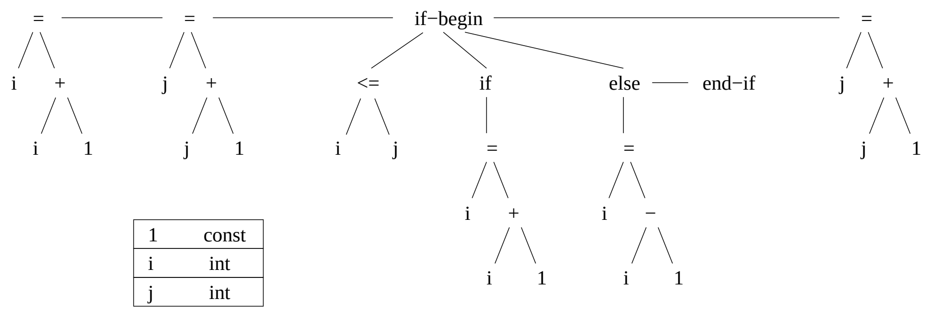
The dictionary table generated for our sample program can then be used for encoding the abstract syntax tree. For that purpose, the nodes of the abstract syntax tree are traversed in pre-order, and as each node is processed, the index of its corresponding node class is written out. For example, the expression *i=i+1* can be encoded as the sequence of indices, *4-9-5-9-8*, corresponding to this expression in prefix notation: *=i+i1*. Encoding of expressions in prefix notation allows the abstract syntax tree of a program to be rebuilt directly as the Slim Binary file is processed.

Application of this simple SDE encodes each assignment of the sample program using at most 6 indices. On closer inspection, it is apparent that certain sequences reappear multiple times within the Slim Binary file (e.g., the encoding of the first and third assignment are identical). The Slim Binary format allows for the compression of recurrences of similar patterns, by adding additional entries to the SDE during the encoding process that express patterns of nodes that have already been seen. As an example, after processing the assignment *i=i+1*, entries for the subexpressions *i=.*, *i+.*, *.+1*, *i+1,.=i+1* and *i=i+1* are inserted into the dictionary table. Figure 8 (c) shows excerpts of the dictionary table extension that would be adaptively built up during the encoding of our sample program. As can be seen, this SDE dynamic extension mechanism reduces the number of indices required for the sample program from 32 to 24 indices.

The insertion of additional entries for the description of these subexpressions increases the size of the dictionary table and with it the number of bits that are required for table index representation. However, in an optimized SDE, not all above discussed dictionary entries must be inserted into the dictionary table. Ref. [22] contains a detailed description of insertion strategies that can be used for effective construction of dictionary tables for the Slim Binary format. During decoding of an abstract syntax tree that has been encoded by a dynamic SDE, the same adaptive construction of the dictionary table must be performed. As a consequence, after the recovery of the expression, *i=i+1*, entries for subexpressions *i=.*, *i+.*, *.+1*, *i+1,.=i+1* and *i=i+1* must be inserted into the dictionary table in the same order on the consumer side as they were on the producer side, since otherwise the abstract syntax tree cannot be regenerated correctly.

The effectiveness of Slim Binaries as intermediate rep-

(a)



1	const
i	int
j	int

(b)

```
int i,j;
...
i = i + 1;          4    9    5    9    8
j = j + 1;          4   10    5   10    8
if (i <= j)         0    7    9   10
   i = i + 1;       1    4    9    5    9    8
else
   i = i − 1;       2    4    9    6    9    8
end if;             3
j = j + 1;          4   10    5   10    8
```

(c)

```
int i,j;
...
i = i + 1;          4    9    5    9    8
j = j + 1;          4   10   13   10
if (i <= j)         0    7    9   10
   i = i + 1;       1   16
else
   i = i − 1;       2   11    6    9    8
end if;             3
j = j + 1;         21
```

Index	Meaning
0	if−begin
1	if
2	else
3	end−if
4	.=.
5	.+.
6	.−.
7	.<=.
...	...
8	1
9	i
10	j
11	i=.
12	i+.
13	.+1
14	i+1
15	.=i+1
16	i=i+1
17	j=.
18	j+.
19	j+1
20	.=j+1
21	j=j+1
22	i<=.
23	.<=j
24	i<=j
25	i−.
26	.−1
27	i−1
28	.=i−1
29	i=i−1
...	...

Figure 8: A Slim Binary Example.

resentation for mobile code was demonstrated by the Juice browser plug-in [27], which allowed Oberon applets (compiled into Slim Binaries) to be executed locally inside the web browser (via Juice's code generator) just like Java applets. Since the Slim Binary format results in smaller file sizes than corresponding Java Bytecode files, transmission times of Juice applets are shorter than for equivalent Java applets [47]. Furthermore, many optimizations can be performed on Juice applets due to the retention of high-level program information.

Program optimizations that are performed on the consumer-side impose additional runtime costs. Therefore, instead of enforcing optimizations during load time, they can be performed as background process while the mobile code is already executed. The Slim Binary format is well suited for this kind of runtime optimization [48].

Within an environment that supports runtime optimizations, Slim Binaries are first transformed into the machine code of the target platform during load time without applying any program optimizations. Subsequently, while the machine code executes, additional transformations and program optimizations can be performed in a separate thread. With each transformation, the quality of the generated machine code is enhanced, until a certain level of optimization is achieved.

Runtime optimizations are also able to support complex transformations (e.g., inter-modular and approximative program optimizations). Extended variants (see, for example, [7, 62]) use adaptive analysis to identify frequently executed parts of the mobile code. Using this information, the optimizations can be performed more efficiently.

A variant of Slim Binaries for the Java language is implemented by the ASTCode format [66]. The main objective of this approach was to produce a more compact intermediate representation than Java Bytecode and to simplify the verification process on the consumer-side. In ASTCode the class file format has been changed slightly. In particular, the constant pool of the class files is used as a symbol table, and instead of Java Bytecode sequences, in ASTCode class files contain sequences of indices of the Semantic Dictionary Encoding. In order to simplify the verification process, the decoding process of a class file in ASTCode is extended by a type-checking procedure. As a result, the complexity of the verification process, which is quadratic for Java Bytecode, is reduced to a linear function of code length.

3.2.3 SafeTSA

We also classify SafeTSA (which stands for Safe Typed Static Single Assignment Form) as a tree-oriented intermediate representation for mobile code, even though it is actually a hybrid format that combines high-level control structures in a AST-like form (called the Control Structure Tree) with individual instructions in static single assignment form [3, 70]. The format was designed as a drop-in replacement for Java Bytecode[6] providing for more efficient just-in-time compilation and an innovative approach to safety based on an *inherently safe* encoding.

SafeTSA's control structure tree provides for all of the non-linear intra-procedural control flow in SafeTSA. The instructions (which only perform computations, manipulate data on the heap, and call methods) are embedded as leaves of the control structure tree with their execution being controlled by their parents in the tree. The high-level control structures provided by SafeTSA (which mirror those provided by the Java programming language), restrict SafeTSA programs to reducible intra-procedural control flow. They also make it possible to do a syntax directed derivation of the control flow graph and dominator tree, and also allow for the possibility of high-speed single-pass syntax-directed JIT compilation of SafeTSA code.

The primary driver of enhanced efficiency for just-in-time compilation of SafeTSA, however, results from the use of Static Single Assignment Form (SSA). Static Single Assignment Form guarantees that each instruction's result variable is unique (i.e., assigned to at only that static location in the program) [14]; this discipline (which is facilitated by special ϕ-functions that merge alternative values that reach a program point on different control flow paths) enables a variety of optimizations that are now standard in state-of-the-art optimizing compilers. In SafeTSA, the use of SSA facilitates producer-side machine-independent optimization and speeds up several consumer-side optimizations. As reported in [4], the net result is that JIT compilers for SafeTSA can deliver the same quality code in less time than a JIT compiler for JVML.

Static single assignment form also plays a key part in SafeTSA's inherently safe encoding. The binary on-the-wire SafeTSA is designed such that it only uses the number of bits required to represent possible program symbols that might result in a syntactically valid and correctly typed program [70]. In this way, the program is more dense, because it is not wasting bits that do not differentiate between correctly typed programs.

In addition, a separate verification phase is unnecessary, because the decoding process only ever produces syntactically valid and correctly typed programs. There are a couple of mechanisms that enable this. Perhaps the most important mechanism is the implicit naming and enumeration of variables according to *dominator scoping* and the *type separation*. The implicit naming is based on the property that, in static single assignment form, each variable is only ever assigned at a single location, so by enumerating the locations where variables are created, one can create names for the variables. In static single assignment form, a variable is live at a program point, if and only if, its defining instruction dominates that program point. Therefore, SafeTSA limits the scope of all SSA variables to the program region dominated by its definitions, and the implicit enumeration takes advantage of this so that variables are enumerated consecutively along the path of the dominator tree to the point the variable is being accessed.

In addition SafeTSA's variable enumeration is *type separated*. That is, there are no implicit coercions, so the variables of each type can be enumerated independently. These mechanisms enable all symbols representing operands to be selected from a list of candidate operands that would be legal in that program location. Simpler mechanisms are used for symbols and other kinds of program elements, and a binary prefix code is generated for each position in the program based on an implicit enumeration of the possible alternative symbols for that position.

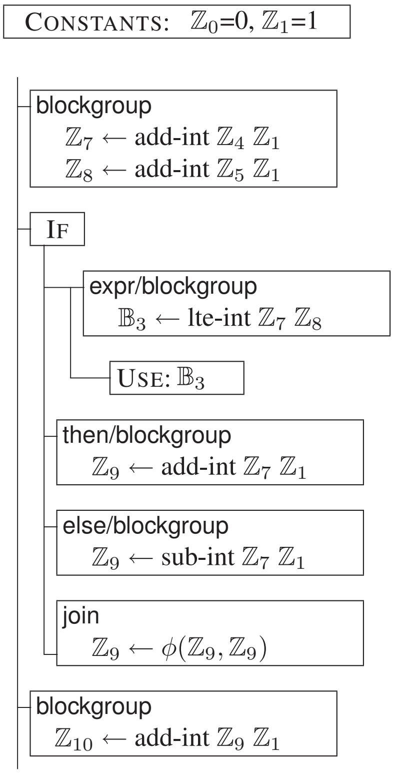
All of these mechanisms can be seen in Figure 9. The section of the control structure tree shown in Figure 9(b) contains a node for the IF statement, a node naming the boolean value that should be "used" to control the IF statement, and several blockgroups, which contain instructions, and some of which are subordinate to the IF statement in particular ways (e.g., the THEN-statement). In order to make the code easier to read, the variables are named with a

[6] And, in fact, the prototype implementation of SafeTSA based on the Jikes Research Virtual Machine supports intermixing classes loaded from both SafeTSA and JVML class files within a single executing virtual machine [4].

```
int i,j;

i = i + 1;
j = j + 1;
if (i <= j)
    i = i + 1;
else
    i = i - 1;
j = j + 1;
```

(a) in Java


CONSTANTS: $\mathbb{Z}_0$=0, $\mathbb{Z}_1$=1
blockgroup
$\mathbb{Z}_7 \leftarrow$ add-int $\mathbb{Z}_4$ $\mathbb{Z}_1$
$\mathbb{Z}_8 \leftarrow$ add-int $\mathbb{Z}_5$ $\mathbb{Z}_1$
IF
expr/blockgroup
$\mathbb{B}_3 \leftarrow$ lte-int $\mathbb{Z}_7$ $\mathbb{Z}_8$
USE: $\mathbb{B}_3$
then/blockgroup
$\mathbb{Z}_9 \leftarrow$ add-int $\mathbb{Z}_7$ $\mathbb{Z}_1$
else/blockgroup
$\mathbb{Z}_9 \leftarrow$ sub-int $\mathbb{Z}_7$ $\mathbb{Z}_1$
join
$\mathbb{Z}_9 \leftarrow \phi(\mathbb{Z}_9, \mathbb{Z}_9)$
blockgroup
$\mathbb{Z}_{10} \leftarrow$ add-int $\mathbb{Z}_9$ $\mathbb{Z}_1$

(b) in abstract SafeTSA

Symbol	Index/Choices	Encoding
statement blockgroup	1/12	001
apply	19/20	11111
add-int	89/185	10100000
$\mathbb{Z}_4$	4/7	101
$\mathbb{Z}_1$	1/7	010
add-int	89/185	10100000
$\mathbb{Z}_5$	5/8	101
$\mathbb{Z}_1$	1/8	001
end blockgroup	0/20	0000
IF	3/12	011
expression blockgroup	1/3	10
apply	19/20	11111
lte-int	104/185	10101111
$\mathbb{Z}_7$	7/9	1110
$\mathbb{Z}_8$	8/9	1111
end blockgroup	0/20	0000
use:	—	—
$\mathbb{B}_3$	3/4	11
then:	—	—
apply	19/20	11111
add-int	89/185	10100000
$\mathbb{Z}_7$	7/9	1110
$\mathbb{Z}_1$	1/9	0001
end blockgroup	0/20	0000
else:	—	—
apply	19/20	11111
sub-int	111/185	10110110
$\mathbb{Z}_7$	7/9	1110
$\mathbb{Z}_1$	1/9	0001
end blockgroup	0/20	0000
join:	—	—
ϕ	0/2	0
$\mathbb{Z}_9$	9/10	1111
$\mathbb{Z}_9$	9/10	1111
end join	1/2	1
statement blockgroup	1/12	001
apply	19/20	11111
add-int	89/185	10100000
$\mathbb{Z}_8$	8/11	1101
$\mathbb{Z}_1$	1/11	001

(c) in SafeTSA's Binary Encoding

Figure 9: The Example Program Fragment in SafeTSA

symbol representing the type ($\mathbb{Z}$ for integer, $\mathbb{B}$ for boolean) and a subscript indicating the variables position in 0-based implicit enumerations. The integer constants, 0 and 1, are declared to be represented by $\mathbb{Z}_0$ and $\mathbb{Z}_1$, respectively. The initial values of i and j are assumed to be $\mathbb{Z}_4$ and $\mathbb{Z}_5$, and it is assumed that there are 7 integers and 2 booleans defined before the first instruction shown. With these assumptions, the first instruction in the first blockgroup adds 1 to $\mathbb{Z}_4$ (i.e., the old i) and puts the result in $\mathbb{Z}_7$ (i.e., the new i). Note that there are several definitions of $\mathbb{Z}_9$, which appears to be a violation of the single assignment property, but none of these definitions dominates any of the others so their scopes do not overlap and they are distinct variables. In fact, this mechanism effectively prohibits accessing non-dominating variables, since their names get re-used by those that do dominate a particular access. Due to the peculiarities of ϕ-functions in SSA, the definition of the third $\mathbb{Z}_9$ actually refers to the first $\mathbb{Z}_9$ and the second operand refers to the second $\mathbb{Z}_9$, but according to SafeTSA's rules [70], only the correct $\mathbb{Z}_9$ is in scope at each of those positions. The rendering of the tree representation into a sequence of symbols, and the binary encoding of those symbols is shown in Figure 9(c).

3.3 Proof-annotated representations

In the past decade, there have been several research projects aiming at the development of certifying compilers. Certifying compilers differ from traditional compilers in that in addition to producing executable code, they also produce an additional annotation (i.e., a certificate) containing a proof that the executable code respects certain safety properties (usually type and memory safety). The proof-annotated code format is designed so that all proofs can be automatically checked in a bounded amount of time. In a mobile code context, such a proof-annotated format can be used to only allow the execution of mobile code for which it is determined that the proofs are correct and that the proofs are sufficient to guarantee that the annotated code satisfies the safety properties required by the mobile code system.

Proof-Carrying Code [55] and Typed Assembly Language [51] are the two primary representatives of proof-annotated mobile code formats. As introduced by George Necula in 1996, proof-carrying code utilizes certificates written in a formalism based on first-order logic. This proof can be generated by the code producer and shipped along with the program code. The code consumer then validates the proof to ascertain the safety of the transmitted mobile code. Due to its foundation in first-order logic, proof-carrying code is quite flexible in terms of the types of safety properties that can be checked using first-order logic; the limiting factors on flexibility are the kinds of properties for which proofs can be generated automatically. The Touchstone compiler[7] is the front-end of a prototype proof-carrying code system that compiles from a safe subset of the C programming language into machine code and certifies that the resulting machine code is type and memory safe [55, 58].

Typed Assembly Language extends traditional untyped assembly languages with typing annotations, memory management primitives, and a sound set of typing rules. These typing rules guarantee the memory safety, control flow safety, and type safety of the transmitted program.

3.3.1 Proof-carrying code

Proof-Carrying Code (PCC) was originally developed as a mechanism for safe operating system kernel extensions, but was later adapted to the area of mobile code [54, 56]. Founded on formal program verification theory, PCC allows the code consumer to check the safety of programs by checking machine-readable proofs that are generated by the code producer and shipped along with the program code. After checking the validity of the proofs, code consumers are then assured that the program execution will not violate the verified properties. The desired safety properties depend on the code consumers safety policy, which acts as a contract between the code producer and code consumer, and defines which conditions must be satisfied by safe mobile programs.

In a proof-carrying code system, the role of the code producer is fulfilled by a certifying compiler, consisting of an annotating compiler and a proof-generator. While the certifying compiler translates the program that is to be transmitted into machine code of the target platform or any other executable code representation, it also annotates the program with additional information (e.g., types) that would otherwise be lost. After this, a theorem (possibly-specialized) prover is used to generate a proof that the generated code complies with the mobile code system's safety policy, and this proof is transmitted along with the code to the code consumer in a PCC binary. The code consumer validates the safety proof based on the actual machine code and the conditions defined in the safety policy. The validation algorithm and the safety policy are the only parts of the system which have to be trusted, minimizing the size of the trusted code base (TCB)[8]. If proof validation is successful, the safety of the transferred mobile program is guaranteed and the machine code can be executed as shown in Figure 10.

Safety conditions, which have to be satisfied by the transferred mobile code are defined within the safety policy and are shared between code producer and code consumer. This safety policy is based on first-order logic and consists of three parts: a verification condition generator (VCG) [19], a set of axioms, and pre- and post-conditions.

The verification condition generator is used to derive a safety predicate (i.e., a verification condition), from the annotated program code. The safety predicate is derived such that it will only evaluate to true if every condition specified

[7] A variant of Touchstone, called SpecialJ, has been developed for the Java programming language [10, 11].

[8] The paper [5] describes an even further reduced trusted code base which incorporates the safety policy into the proof.

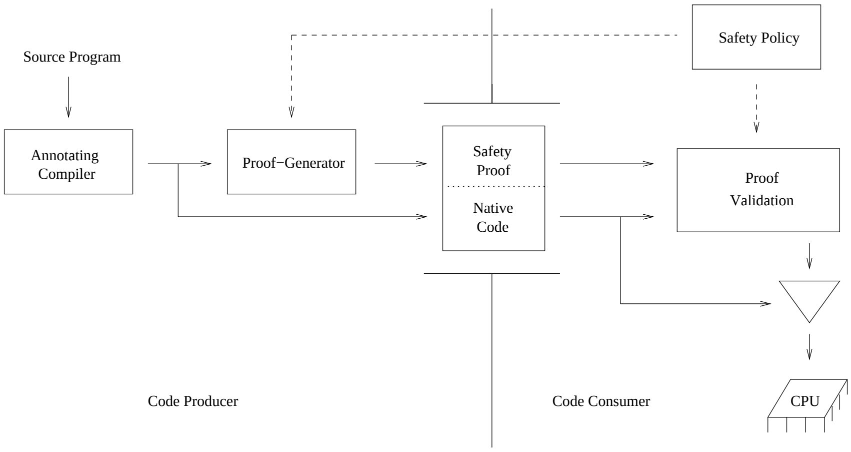


Figure 10: Proof-Carrying Code Architecture.

in the safety policy is satisfied. The pre- and post-condition included in the safety predicate, express constraints on the machine state that must hold, respectively, before and after program execution. The safety predicates, which are defined in first-order logic, are derived from a set of axioms and derivation rules that model the state transitions of the target machine associated with each instruction.

Both, code producer and code consumer, derive the safety predicate from the annotated program code. The code producer generates a proof of the safety predicate, indicating the safe execution of the program code, and the code consumer derives the safety predicate in order to check the matching of program code and safety proof. Thus, the verification condition generator traverses the program code and creates predicates for each critical instruction (e.g., memory access) using a symbolic interpreter, such that a proof of these predicates ensures that executing the corresponding instruction does not invalidate the conditions defined in the safety policy.

To support complex program structures like method calls and loops, the verification condition generator uses *invariants* annotated in the program code. These annotations are frequently required to mark loop invariants, which cannot normally be automatically derived by the verification condition generator. The invariants are included among the predicates which must be verified. Thus, the code consumer does not need to trust the program annotations, and the invariants are only used as hints supporting the generation, and the validation of the safety proof. These predicates must be proved to hold for every control path between two distinct invariants, starting with the pre-condition and finishing with the post-condition. As a consequence, the safety predicate of the whole program is the conjunction of all predicates derived from the invariants and the individual instructions.

After the safety predicate has been derived by the verification condition generator, the code producer creates a proof, which shows the correctness of the generated safety predicate. This safety proof is represented using the Edinburgh Logical Framework or LF notation [42, 68]. The Edinburgh Logical Framework efficiently validates the proofs by reducing validation to a simple type-checking procedure [55, 57]. (In other words, in the Edinburgh Logical Framework, only correct proofs are correctly typed.) Thus, the code consumer of a PCC system can be guaranteed that a mobile code program satisfies its safety policy prior to its execution.

As a concrete example of a PCC system, let us examine the output of the Touchstone certifying compiler. The Touchstone certifying compiler might translate the source program, shown in Figure 11 (a), into the slightly optimized DEC ALPHA assembly code as shown in Figure 11 (b). Note that a pre- and post-condition are annotated, both stating that registers v0 and t0, which represent variables i, j of the source program respectively, contain integer values. In order to verify the safety of the machine code, the Touchstone certifying compiler generates a verification condition and a safety proof, indicating the validity of the verification condition. The verification condition, shown in Figure 11 (c), states type and memory safety of the machine code. Therefore, the verification condition denotes the implication that: assuming the registers v0 and t0 hold integer values initially, their values after manipulation will still be of type integer after the machine code has been executed. Thus, the post-condition can be derived from the pre-condition, and the validity of the verification condition is proven. The safety proof shown in Figure 11 (d) states the validity of the verification condition and therefore guarantees type and memory safety of the machine code.

One drawback of PCC is the size of the generated safety proofs, especially since the proofs have to be transmitted along with the code to the code consumer. The transfer of the safety proof conjoined with the program code is performed using a PCC binary, which consists of three parts. First, the program to be transferred is included using an intermediate representation or the machine code of the target platform. In the latter case, the program can be directly loaded and executed after the mobile code has been suc-

```
(a) source program

    int i, j;
        i = i + 1;
        j = j + 1;
        if (i <= j)
                i = i + 1;
        else
                i = i - 1;
        j = j + 1;

(b) annotated machine code

        ANN_PRE(example, LF_
                        (/\ (of t0 int)
                            (of v0 int))_LF)

        subl    t0, v0,t1
        blt     t1, L2
        lda     v0, 2(v0)
    L2:
        lda     t0, 2(t0)

        ANN_POST(example, LF_
                        (/\ (of t0 int)
                            (of v0 int))_LF)

(c) verification condition

pf (all ([X0 : exp]
         (all ([X1 : exp]
              (=> (/\ (of X1 int) (of X0 int))
                  (/\ (=> (>= (- X1 X0) 0)
                          (/\ (of (+ X1 2) int)
                              (of (+ X0 2) int)))
                      (=> (< (- X1 X0) 0)
                          (/\ (of (+ X1 2) int)
                              (of X0 int))))))))

(d) proof

(alli [X0 : exp]
 (alli [X1 : exp]
  (impi [A1 : pf (/\ (of X1 int) (of X0 int))]
   (andi (impi [A2 : pf (>= (- X1 X0) 0)]
     (andi (ofIntAny (+ X1 2))
      (ofIntAny (+ X0 2))))
    (impi [A3 : pf (< (- X1 X0) 0)]
     (andi (ofIntAny (+ X1 2))
      (ander A1)))))))
```

Figure 11: A sample program (a) and its PCC output (b), (c) and (d).

cessfully verified. The second part of the Proof-Carrying Code binary contains a symbol-table, which is used to reconstruct the LF representation of the safety proof on the consumer side. The last part includes the safety proof in a binary encoding.

3.3.2 Typed assembly language

The use of a Typed Assembly Language as intermediate representation benefits a mobile code system in several ways. First of all, a number of program optimizations are enhanced by having type information available in the assembly code. In addition, the type annotations facilitate the verification of the mobile code's type safety. In order to gain these benefits, a type abstraction of assembly languages is required, which guarantees type safety of well-formed assembly programs and hence enables the transformation of well-formed input programs into safe assembly code [38]. Since the type annotations and type checking, serves to prove that a program in a Typed Assembly Language is type safe, it can be considered as a kind of proof-annotated representation, even though the proof is not that of a direct assertion of safety in a general logic as it is in proof-carrying code [52].

The feasibility of Typed Assembly Language was demonstrated by the reference implementation, TALx86

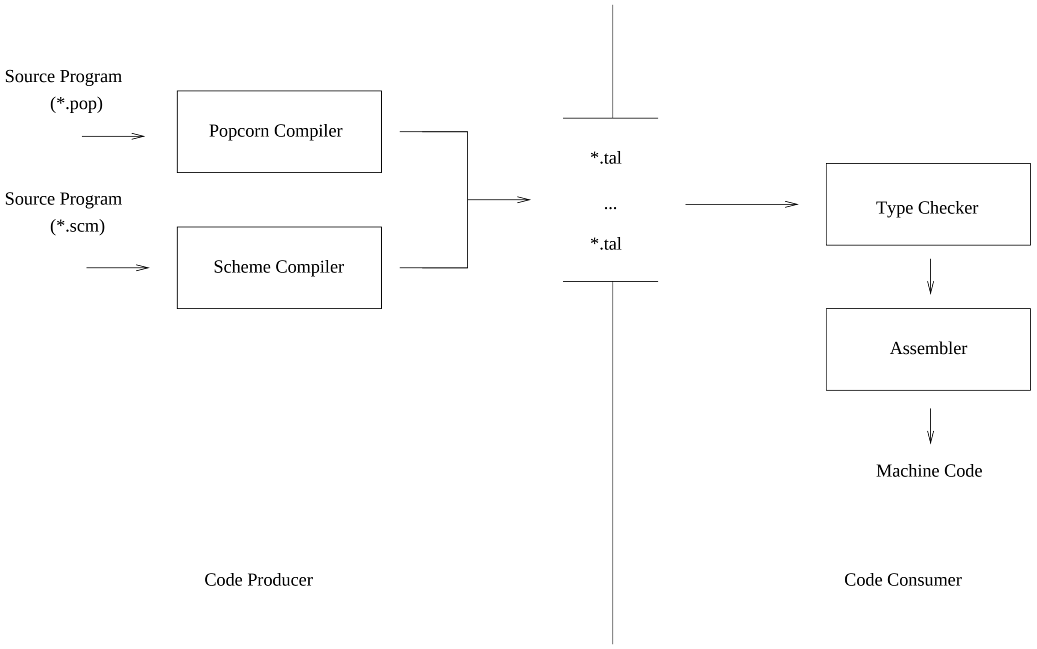


Figure 12: Talx86 system.

[36]. TALx86 incorporates support of the programming languages Scheme and Popcorn (a type-safe subset of the C language from which unsafe constructs such as pointer arithmetic and address operator have been removed). On the producer side of the prototype TALx86 system (see Figure 12), programs of the Scheme or Popcorn programming language are transformed into assembly code for the target platform, and the generated assembly code is annotated with type information resulting in Typed Assembly Language. After receiving a program, the code consumer uses the annotated type information to type-check the transferred assembly program. If the mobile code is successfully verified, an assembler transforms the assembly code into machine code for the target platform, which is then executed.

The TALx86 implementation is based on Microsoft's Macro Assembler Language, and therefore, TALx86 programs, after being type-checked, can be efficiently assembled with common commercial assemblers. Within the TALx86 system, the register-based Microsoft Macro Assembler Language is extended with annotations, which are mainly used as pre-conditions of code labels, assigning type information to registers.

Our sample program (now written in Popcorn) in Figure 13 (a) and its translation into TALx86 are shown in Figure 13 (b). The TALx86 sequence consists of two sections: the assembly language instructions and its corresponding type annotations. The assembler program starts by calculating increments of values contained in registers EBP and EDI, which represent the variables i and j, respectively. Subsequently, the values are compared, and if the value contained in EBP exceeds the value in EDI, execution continues at the code label ifFalse$49, which represents the else branch of the program. Otherwise, the then branch of the if statement is executed, and the value contained in EBP is incremented. After that, program execution proceeds to the label ifMerge$50, which marks the end of the if statement, and the value in register EDI is incremented again.

Both labels are annotated with the types of values contained in the registers at that point. As an example, annotation EDI:B4 denotes type B4 in register EDI, indicating a 4-byte integer value inside. On the consumer side, the annotation is then used by the type-checker, so that it only has to check that the register EDI contains a value of type B4 before control is given to label ifFalse$49 or ifMerge$50 (rather than propagating types around the control-flow graph until a fixed pointer is reached or a type error is detected).

Polymorphic types, required for describing high-order structures like stacks, are realized using placeholders, which are replaced by corresponding types before control is transferred to the associated code label. Type annotations are also used to define new types with type constructor declarations. This flexibility of the type system allows one to support a wide range of programming languages. TALx86 allows the code consumer to provide routines of instructions (i.e., macros) for manipulating complex data structures that can be typed and treated as atomic operations during verification. Programs may explicitly allocate such complex data structures using the macro malloc but are not allowed to explicitly de-allocate the structures; this is done implicitly, through garbage collection.

Stacks in TALx86 programs are modeled by abstractions that are based on lists. The expression t :: s denotes a stack,

```
(a)     int i, j;
        i = i + 1;
        j = j + 1;
        if (i <= j)
                i = i + 1;
        else
                i = i − 1;
        j = j + 1;

(b)     MOV     EDX,EBP
        ADD     EDX,1
        MOV     EBP,EDX
        MOV     ESI,EDI
        ADD     ESI,1
        MOV     EDI,ESI
        CMP     EDX,ESI
        JG      tapp(ifFalse$49,<r$9>)
        ADD     EBP,1
        JMP     tapp(ifMerge$50,<r$9>)
ifFalse$49:
LABELTYPE <All[r$9:Ts].{EDI: B4,EBP: B4,ESP: sptr {ESP: sptr r$9}::r$9}>
        MOV     ESI,EBP
        SUB     ESI,1
        MOV     EBP,ESI
        FALLTHRU        <r$9>
ifMerge$50:
LABELTYPE <All[r$9:Ts].{EDI: B4,EBP: B4,ESP: sptr {ESP: sptr r$9}::r$9}>
        ADD     EDI,1
```

Figure 13: Sample program in Popcorn (a) and corresponding TALx86 program (b).

consisting of a top-most element of type t and the rest of the stack described by s. Placeholders are applied in order to enable the polymorphic representation of stacks. A placeholder is of a general form All[s:Ts] where Ts denotes the abstract type, which is substituted by a corresponding instance s.

As a consequence, function calls are represented by the help of a runtime stack s, which is referenced by stack pointer sptr s contained in register ESP. This can bee seen at labels ifFalse$49 and ifMerge$50 of Figure 13 (b), where register ESP references the runtime stack, which contains another stack pointer representing the caller frame, and the rest of the stack denoted by the polymorphic type r$9.

Polymorphic types in combination with runtime stacks are also used to implement visibility rules, which make the actual representation of abstract types associated to local variables only resolvable by the authorized function. Furthermore, this mechanism supports exception handling by restricting register access. A dedicated register contains a stack pointer which indicates where to unwind the runtime stack to. This stack pointer is typed so that it is abstract and therefore unmodifiable by everything except for the exception code.

The time-critical processes in the TAL system include the code consumer's type-checking and the transfer of the mobile program to the code consumer. Thus, a compact encoding of Typed Assembly Language is needed for optimal performance. Since the annotations (e.g., pre-conditions of code-labels) increase the code size, various compression techniques can be applied to increase the density of the annotation format, so that it is more suitable for mobile code [37]. These techniques include, among other, the sharing of common sub-terms within annotations, the use of generic type abbreviations, and the elimination of unnecessary annotations.

4 Review and comparison

In the following sections, the mobile code representations that were presented earlier will be evaluated according to the requirements introduced in Section 2. Table 1 summarizes this evaluation and can be used as a guide to the discussion that follows.

4.1 Source language flexibility

Although the JVM was designed to support Java semantics, it can also be used as a target for other languages. Indeed, several compilers for C, C++, and Ada95, target the JVM. However, these language's intrinsic insecurities, and their semantic mismatch with Java, require the programmer to adhere to restrictive feature subsets [32]. In order to avoid such disadvantages, .NET and its intermediate representation CIL were designed to efficiently support a variety of object-oriented, functional, and procedural programming languages, including C#, C++, Java, Fortran, Cobol, Eiffel, Haskell, ML. Furthermore, the .NET platform's Common Type System serves as a common denominator that aids cross-language interoperability, so that .NET components can interact with each other even if they are written

	Stack-based		Tree-oriented			Proof-Annotated	
	JVML	CIL	ANDF	Slim Binary	SafeTSA	PCC	TAL
Flexibility							
Input-Languages	✓	+	✓	✓	✓	++	+
General type system	−	++	−	−	−	−	++
Portability							
Target-Architectures	++	++	+	+	+	−	−
Compactness							
Encoding Density	✓	✓	+	++	+	−	−
Efficiency							
Interpreter	++	++	✓	✓	✓	−	−
JIT compiler	✓	✓	+	+	++	✓	✓
Producer Optimizations	✓	✓	+	+	+	++	++
Producer Annotations	✓	✓	✓	✓	++	++	++
Consumer Optimizations	+	+	+	+	+	✓	✓
JIT Cost	✓	✓	+	+	++	✓	✓
Safety							
Safety	++	++	−	++	++	++	++
Automated	++	++	−	++	++	−	++
Runtime Complexity	✓	✓	N/A	+	+	+	+

Legend: −poor/no, ✓adequate, +good/yes, ++excellent.

Table 1: Intermediate representations in comparison.

in different CTS-supporting languages.

Tree-oriented intermediate representations tend to be more limited in their linguistic flexibility. The current ANDF-System supports Ada95 and C; Slim Binaries' and SafeTSA's prototype systems are built to support the source language Oberon and Java, respectively. Although, it seems that tree-oriented techniques are limited to programs written in predetermined languages, representatives of this kind of intermediate representation also can be extended for addressing a multiplicity of source languages. Basically, such an extension might be based on the construction of an unified abstract syntax tree and a more general type system.

In principle, the greatest source-language flexibility can be achieved with proof-annotating intermediate representations, since for most programming languages, a front end, which translates a source program into native code, can be easily constructed. Furthermore, a principle objective of the TAL project was the development of a statically typed, low-level intermediate representation, that could be used for multiple source languages and on which multiple program optimizations could be performed [64]. For the description of type systems of different source languages, the TAL system transforms a program internally into an intermediate representation that is based on a high-order λ calculus, from which eventually after several type-conserving restructurings the TAL program is derived.

4.2 Portability

A code consumer can execute mobile code applications using an interpreter, a JIT compiler or both. On most current desktop and server computer systems adaptive JIT compilation techniques provide the best performance. However, as small resource constrained devices (e.g., cell phones, PDA's, Java cards) become more and more ubiquitous, interpreters in mobile code systems have become more important, since compared to compilation, interpretation usually uses less resources.

Stack-oriented intermediate representations provide an excellent foundation for the development of fast and efficient interpreters. In contrast, interpretative program execution is supported from none of the tree-oriented prototype systems, as these mobile code formats mainly focus on JIT compilation. Nevertheless, tree-oriented systems can include interpreters (see e.g. [41, 31]), which may be slower than for stack-based counterparts, but could be used for program execution on platforms for which JIT compilers are not yet available.

JIT compilers that transform JVML and CIL programs, respectively, into machine code have been developed for the most common computer systems. Although all of the presented tree-oriented systems are developed for a restricted number of architectures, in principle, these intermediate representations can be considered just as portable. That is, since targeting other architectures needs only more engineering resources to implement new back-ends translating tree-based program representations into their corresponding machine code.

Although PCC and TAL, in their original incarnations, are based on the target machine assembly language, and thus are not portable. In principle, they could also be used as input for JIT compilation, in which case, the assembly language could be replaced with a more general register-based language. A candidate for such an all-purpose low-level language could be the intermediate representation used by the VCODE system (which is the machine code of an idealized RISC-Architecture) [18]. This would serve as a common target language for programs written in different programming languages and as input for an on-the-fly machine code generation of different architectures.

4.3 Compactness

Compactness of mobile code applications plays a major role, especially as many today's network connections are wireless and have a limited bandwidth. In such networks, raw throughput rather than network latency is the main bottleneck. Moreover, increasing use of mobile code on constrained devices, also puts attention on the size of program representation due to limited memory resources.

The use of tree-oriented intermediate representations usually leads to better file sizes than stack-based techniques. In particular, according to measurements described in [24], Slim Binaries are more dense than compressed JVML class files by a factor of 1.72. And for uncompressed JVML the ratio in file sizes in average even can increase up to 2.42. SafeTSA, which has a hybrid tree/SSA structure, is not quite as dense, but as reported in [70], has a binary on-the-wire file size similar to compressed JVML class files.

Stack-based intermediate representations, in turn, are often more compact than the corresponding machine code [24]. Proof-annotating intermediate representations are still larger, because in addition to be based on less compact machine code or assembly language, the file sizes of proof-annotating intermediate representations are also increased by their proof or type annotations. Unfortunately, there exists no measurements about file sizes of proof-annotated code compared to JVML programs. However, measurements in a prototype PCC system resulted in an average ratio of proof size to code size of 2.5 [58]. Comparable experiments performed in the TAL system led to a ratio of up to 0.67 [37]. These results indicate a significant increase in file sizes when applying proof-annotating techniques, and consequently a need of sophisticated compression techniques.[9]

4.4 Efficiency

We call the property of an intermediate representation to support a fast and resource-efficient program execution, the representation's efficiency. Although it is a matter of common knowledge, that fast program execution is primarily achieved through the use of JIT compilation techniques, the efficiency with which a mobile code format can be interpreted is also important, especially as resource-constrained devices become more ubiquitous.

Stack-oriented intermediate representations are excellent candidates for interpretation. The main advantage of this architecture as input for an interpreter, is the compact instruction encoding (due to most operands being taken off the stack). Although, tree-oriented mobile code formats can also be interpreted, tree-based interpreters are not such efficient than its bytecode counterparts, because of a higher storage consumption and slower execution times, which are direct consequences of its internal representation as pointer structures. Register-based interpreters are also possible [65, 15], but have not been employed in industrial strength mobile code systems.

In recent years a lot of powerful JIT compilers for stack-based mobile code formats have been developed; especially notable are Sun's HotSpot compiler [62, 6] and IBM's Jikes RVM [7, 44], respectively. However, the popularity of existing stack-based JIT compilers belies the limitations of stack-based intermediate representation when used as input for a JIT compiler. Certainly, simple machine code for stack-based programs can be generated quickly, but for aggressive JIT compilation (i.e., with several complex optimizations) stack-based representations have some disadvantages.

The main disadvantage for aggressive JIT compilation of stack-based code is the use of the stack model. This approach requires the compiler to generate optimized register-based machine code for a program that is expressed in terms of the manipulation of a virtual stack machine. Most existing stack-based JIT compilers solve this problem by expending compilation effort to transform their input programs into an internal three-address code representation (often in SSA form) on which the optimizations are performed.

A further disadvantage is the low-level character of stack-based program code, which often prevents reconstruction of high-level language information, which is essential for certain optimizations. In addition, performing machine independent optimization on the producer side of a stack-based system is difficult. For example, while a compiler generating stack-based JVML code could, in principle, perform common subexpression elimination and store the resulting expressions in additional, compiler-created local variables, this approach introduces additional instructions and temporary variables that may negate any improvements created by the common subexpression elimination.

In contrast, due to its high-level entities, tree-oriented code formats are excellent candidates for JIT compilation. In principle, JIT compilers based on these intermediate representations can be just as effective as static compilers. The main advantage of a tree-oriented JIT compilation is the preservation of high-level information that aides the quick generation of fast code (e.g., explicitly marked loops, loop invariant codes, exclusion of irreducible control flow graphs).

SafeTSA successfully augments a tree-oriented intermediate representation with instructions in SSA form, which is already used internally in several static and JIT compilers and that is considered the state-of-the-art intermediate rep-

[9] as described for TAL in [37] and PCC in [59]

resentation for intra-procedural scalar optimizations. Several efficient optimization techniques have been developed for SSA programs in the last decade. Experiments, described in [4], confirm that JIT compilers using SafeTSA run faster than those using JVML code, reducing the cost for dynamic optimization of some programs by up to 90%. As mentioned above, the machine-independent optimization of stack-based mobile code formats is often awkward, but for tree-oriented and proof-annotated formats, such optimizations cause no further difficulties.

In recent years program annotations have been suggested as a way to improve the code generation of JIT compilers. The term program annotation is used as a synonym for code information added to the mobile code during its generation. This information can be used by the consumer side of a mobile system to speed-up optimizations of a given program. In principle, all types of intermediate representations support the transport of program annotations. The main challenge after transferring mobile code to the runtime environment is the verification of the transmitted annotations. Conceptually, verifiable program annotations can be constructed for PCC and TAL programs through proof and type extensions, respectively. In SafeTSA programs the concept of type separation can be applied in a tamper-proof manner for the safe transport of program annotations [43].

4.5 Safety

Safety is an important criterion in a mobile code system due to the inherent separation of code consumers and code producers. In general, mobile code can be created by an untrusted code producer and transferred through insecure communication channels to the code consumer, so the code consumer needs to verify that the transmitted mobile code will not perform any unsafe actions when executed.

In addition to other mechanisms such as cryptographic signatures, intermediate representations of mobile code address the safety issue using several distinct approaches, ranging from implicitly legal program encodings to formal methods like program verification using first-order logic as applied by Proof-Carrying Code. Common to all of them is the focus on guaranteeing type and memory safety as well as a legal control flow of the verified mobile program in order to provide fine grain isolation of code within the execution environment.

Stack-based intermediate representations for mobile code utilize a data-flow analysis in the verification process. This data-flow analysis is required due to the semantic gap between high-level source language and low-level intermediate representation [66]. Hence, as in the case of Java Bytecode, well-formed Java Bytecode sequences do not necessary represent legal Java programs. Adherence to certain safety concepts of the high-level source language is therefore verified using the data-flow analysis, which is performed on the consumer side of the mobile code system and targeted at type and memory safety as well as a legal control flow of inspected bytecode.

In addition to some semantic errors in original specifications and implementations of data-flow analysis for Java Bytecode verification [67], this approach also suffers from its immense costs, requiring quadratic time regarding the number of verified instructions in the worst case [66]. Because all of the verification work has to be done by the code consumer, this factor introduces another point of attack. Furthermore, since Java Bytecode verification assumes the type system and other safety concepts of the Java programming language, extending the underlying data-flow analysis to other programming languages and safety concepts is complicated. Due to its support for a wide range of programming languages, the Common Intermediate Language is more flexible on this point.

Two of the tree-oriented intermediate representations for mobile code, Slim Binaries and SafeTSA, represent one approach to avoid the semantic gap between source language and intermediate representation, and consequently facilitate the verification process. ANDF, the third tree-oriented intermediate representation, does not integrate a verification mechanism and so will not be discussed further in this respect.

The Slim Binaries format, as well as SafeTSA, implements a program encoding which is based on the abstract syntax tree and hence close to the high-level source language. Furthermore, both formats restrict their expressiveness to legal programs, (i.e., code violating safety criteria of the source language like type or memory safety can not be encoded by a well-formed Slim Binaries or SafeTSA program [25]. Thus, verification of mobile code is essentially done by checking the adherence to the general format of the corresponding intermediate representation.

As a consequence, complexity of the verification process can be reduced to linear time with regard to code length, as in the case of ASTCode [66]. The Slim Binaries format and its variant ASTCode have been designed for the Oberon and Java programming languages, respectively, hence addressing other languages with differing safety properties is a non-trivial task. This drawback also relates to SafeTSA, though it provides a mechanism for safe program annotations, which may be utilized in a broadened verification process incorporating extended safety criteria [43].

Proof-Carrying Code is based on the concept of certifying compilers (i.e., that produce a machine-readable safety certificate to accompany the mobile code and guarantee its safe execution). Due to the formal representation of the certificate using first-order logic and the small trusted code base [58, 5], Proof-Carrying Code can be seen as an intrinsically safe intermediate representation for mobile code.

Furthermore, the genericity of the underlying approach allows the incorporation of extended safety criteria by adapting the logic of the safety policy and the proof generator. Current applications of Proof-Carrying Code, however, are typically limited to type and memory safety. The main drawback of Proof-Carrying Code also relates to its foundation on formal program verification theory: loop invariants must be annotated as part of the safety proof generation, but these invariants cannot always be automatically inferred from the program, so manual annotations may be necessary if the properties to be proved are more complex than type safety in a tractable type system.

Because creating the safety certificate is expensive, proof generation has been shifted to the producer side of the corresponding mobile code system. The code consumer needs only to verify the shipped proof using a type-checking procedure, requiring linear time with regard to the code length [10], and its consistency with the accompanied program.

Typed Assembly Language, as a variant of Proof-Annotating Code, restricts its safety guarantees to type and memory safety as well as the legal control flow of assembly programs, and stresses the translation of type-correct programs of the source language to type-correct assembly code [52]. The restricted scope of the verification process allows to automatically generate the safety certificate, in the form of type annotations, on the producer side of the corresponding mobile code system. Furthermore, the generation of a safety proof, as required by Proof-Carrying Code, is omitted, since verification on the consumer side is done by a type-checker which utilizes the annotated type information of the transmitted assembly code.

It should be noted, that the three presented safety concepts (i.e., verification based on data-flow analysis, implicitly legal program encodings, and certifying compilers) are orthogonal and may be combined in several ways. All of these concepts rely on representing only programs translated from a safe source language, and—with the exception of the Common Intermediate Language's unmanaged extensions for unverified programs—unsafe features like pointer arithmetic are not supported by any of the presented intermediate representations of mobile code.

5 Conclusion

In the paper, we have provided an overview of common intermediate representations of mobile code, discuss the strengths and weaknesses of each, and compare its properties with that of the other representations.

The comparison of different intermediate representations (see Table 1) leads us to the conclusion that there is no unqualified 'best' mobile code format. One reason for this may be that due to a tendency to focus on one single aspect of the mobile code framework. For example, the developers of PCC were mostly concerned with providing increased security but did not address portability. On the other hand, developers of ANDF provided a very portable distribution format but did not address advanced safety requirements.

Instead, it is obvious that for each intermediate representation, there are disadvantages, which cause it to fail to live up to the ideal. As a consequence, except for Microsoft's CIL representation, none of the suggested mobile code formats can be seen as a serious commercial challenger for Java's Bytecode format. This is also supported by the observation that, except for Java Bytecode and CIL, for none of the other intermediate representations a mobile code system other then of prototype status has been developed.

Because of the wide acceptance of Java Bytecode and because none of the alternative intermediate representations is the ne plus ultra, most current mobile code projects have shied away from the developing of novel intermediate representations. Instead, recent research projects in that area attempt to improve the JVM [20] or integrate features of some of the representations into Java Bytecode. In particular, representatives of this trend are projects that adapt the concepts of Proof-Carrying Code and type-separation to Java Bytecode [2, 26, 72].

Acknowledgment

This investigation has been supported in part by the Deutsche Forschungsgemeinschaft (DFG) under grants AM-150/1-1 and AM-150/1-3.

References

[1] A.-R. Adl-Tabatabai, G. Langdale, S. Lucco, and R. Wahbe. Efficient and language-independent mobile programs. In *Proceedings of the Conference on Programming Language Design and Implementation (PLDI'1996)*, volume 31 of *ACM SIGPLAN Notices*, pages 127–136, New York, May 1996. ACM Press.

[2] P. Adler and W. Amme. Improving the java virtual machine using type-separated bytecode. In *Proceedings of the Workshop on Compilers for Parallel Computers (CPC'2006)*, pages 256–263, Jan. 2006.

[3] W. Amme, N. Dalton, M. Franz, and J. von Ronne. SafeTSA: A type safe and referentially secure mobile-code representation based on static single assignment form. In *Proceedings of the Conference on Programming Language Design and Implementation (PLDI'2001)*, volume 36 of *ACM SIGPLAN Notices*, pages 137–147, Snowbird, Utah, USA, June 2001. ACM Press.

[4] W. Amme, J. von Ronne, and M. Franz. Ssa-based mobile code: Implementation and empirical evaluation. Technical Report CS-TR-2006-005, Computer Science, The University of Texas at San Antonio, 2006.

[5] A. W. Appel. Foundational proof-carrying code. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS'01)*, pages 247–256, Boston, MA, USA, June 2001. IEEE Computer Society Press.

[6] E. Armstrong. Cover story: HotSpot: A new breed of virtual machine. *JavaWorld: IDG's magazine for the Java community*, 3(3), Mar. 1998.

[7] M. Arnold, S. Fink, D. Grove, M. Hind, and P. F. Sweeney. Adaptive optimization in the Jalapeño JVM. In *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages and Application (OOPSLA'2000)*, volume 35 of *ACM SIGPLAN Notices*, pages 47–65, New York, Oct. 2000. ACM Press.

[8] F. Broustaut, C. Fabre, F. de Ferrière, É. Ivanov, and M. Fiorentini. Verification of ANDF components. *ACM SIGPLAN Notices*, 30(3):103–110, Mar. 1995.

[9] J. Bundgaard. An andf based ada 95 compiler system. In *TRI-Ada '95: Proceedings of the conference on TRI-Ada '95*, pages 436–445, New York, NY, USA, 1995. ACM Press.

[10] C. Colby, P. Lee, G. C. Necula, F. Blau, M. Plesko, and K. Cline. A certifying compiler for Java. In *Proceedings of the Conference on Programming Language Design and Implementation (PLDI'2000)*, volume 35 of *ACM SIGPLAN Notices*, pages 95–107, New York, June 2000. ACM Press.

[11] C. Colby, G. C. Necula, and P. Lee. A proof-carrying code architecture for Java. In *Proceedings of the International Conference on Computer Aided Verification (CAV'2000)*, June 2000.

[12] R. Crelier. OP2: A Portable Oberon Compiler. Technical Report 1990TR-125, Swiss Federal Institute of Technology, Zürich, Feb. 1990.

[13] I. F. Currie. *TDF Specification, Issue 4.0*. Defence Research Agency, England, June 1995.

[14] R. Cytron, J. Ferrante, B. K. Rosen, M. N. Wegman, and F. K. Zadeck. Efficiently computing static single assignment form and the control dependence graph. *ACM Transactions on Programming Languages and Systems*, 13(4):451–490, Oct. 1991.

[15] B. Davis, A. Beatty, K. Casey, D. Gregg, and J. Waldron. The case for virtual register machines. In *IVME '03: Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators*, pages 41–49, New York, NY, USA, 2003. ACM Press.

[16] M. de Icaza and B. Jepson. Mono and the .Net framework. *Dr. Dobb's Journal of Software Tools*, 27(1):21–24, 26, Jan. 2002.

[17] J. des Rivières and J. Wiegand. Eclipse: A platform for integrating development tools. *IBM Systems Journal*, 43(2):371–383, 2004.

[18] D. R. Engler. VCODE : A retargetable, extensible, very fast dynamic code generation system. In *Proceedings of the Conference on Programming Language Design and Implementation (PLDI'1996)*, volume 31 of *ACM SIGPLAN Notices*, pages 160–170, New York, May 1996. ACM Press.

[19] R. W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science*, volume 19 of *Proceedings of Symposia in Applied Mathematics*, pages 19–32, Providence, Rhode Island, Apr. 1967. American Mathematical Society.

[20] B. Folliot, I. Piumarta, L. Seinturier, C. Baillarguet, C. Khoury, A. Leger, and F. Ogel. Beyond flexibility and reflection: The virtual virtual machine approach. In D. Grigoras, A. Nicolau, B. Toursel, and B. Folliot, editors, *IWCC*, volume 2326 of *Lecture Notes in Computer Science*, pages 16–25. Springer, 2001.

[21] M. Franz. Emulating an operating system on top of another. *Software – Practice and Experience*, 23(6):677–692, June 1993.

[22] M. Franz. *Code-Generation On-the-Fly: A Key for Portable Software*. PhD thesis, Institute for Computer Systems, ETH Zürich, 1994.

[23] M. Franz. The Java Virtual Machine: A passing fad? *IEEE Software*, 15(6):26–29, Nov. / Dec. 1998.

[24] M. Franz. Open standards beyond java: On the future of mobile code for the internet. *J. UCS*, 4(5):522–533, 1998.

[25] M. Franz, W. Amme, M. Beers, N. Dalton, P. H. Frohlich, V. Haldar, A. Hartmann, P. S. Housel, F. Reig, J. von Ronne, C. H. Stork, and S. Zhenochin. *Making mobile code both safe and efficient. In Foundations of Intrusion Tolerant Systems*, pages 337–356. IEEE Computer Society Press, 2003.

[26] M. Franz, D. Chandra, A. Gal, V. Haldar, C. W. Probst, F. Reig, and N. Wang. A portable virtual machine target for proof-carrying code. *Journal of Science of Computer Programming*, 57(3):275–294, Sept. 2005.

[27] M. Franz and T. Kistler. Introducing juice. Published in Internet, 1996.

[28] M. Franz and T. Kistler. Slim Binaries. *Communications of the ACM*, 40(12):87–94, Dec. 1997.

[29] M. Franz, C. Krintz, V. Haldar, and C. H. Stork. Tamper proof annotations. Technical Report 02-10, Department of Information and Computer Science, University of California, Irvine, Mar. 2002.

[30] M. Franz and S. Ludwig. Portability redefined. In *Proceedings of the 2nd International Modula-2 Conference*, Loughborough, England, Sept. 1991.

[31] A. Gampe. An interpreter for safetsa. Master's thesis, 2006. Masters thesis, Friedrich-Schiller-University, Jena, Germany.

[32] F. Gasperoni and G. Dismukes. Multilanguage programming on the JVM: The Ada 95 benefits. *ACM SIGADA Ada Letters*, 20(4):3–28, Dec. 2000. Special Issue: Presentations from SIGAda 2000.

[33] J. Gosling. Java intermediate bytecodes. In *Proceedings of the Workshop on Intermediate Representations (IR'1995)*, volume 30 of *ACM SIGPLAN Notices*, pages 111–118, San Francisco, CA, Jan. 1995. ACM Press.

[34] J. Gosling, B. Joy, G. L. Steele, and G. Bracha. *The Java Language Specification Second Edition*. The Java Series. Addison-Wesley, Reading, MA, USA, second edition, 2000.

[35] K. J. Gough. Stacking them up: a comparison of virtual machines. In *Proceedings of the 6th Australasian conference on Computer systems architecture*, pages 55–61. IEEE Computer Society Press, Feb. 2001.

[36] D. Grossman and G. Morrisett. Scalable certification of native code: Experience from compiling to TALx86. Technical Report TR2000-1783, Cornell University, Computer Science, Feb. 2000.

[37] D. Grossman and J. G. Morrisett. Scalable certification for typed assembly language. In R. Harper, editor, *TIC*, volume 2071 of *Lecture Notes in Computer Science*, pages 117–146. Springer, 2000.

[38] D. Grossman, J. G. Morrisett, and S. Zdancewic. Syntactic type abstraction. *ACM Trans. Program. Lang. Syst*, 22(6):1037–1080, 2000.

[39] V. Haldar, C. H. Stork, and M. Franz. The source is the proof. In *The 2002 New Security Paradigms Workshop*, pages 69–74, Virginia Beach, VA, USA, Sept. 2002. ACM SIGSAC, ACM Press.

[40] B. S. Hansen and J. U. Toft. The formal specification of ANDF, an application of action semantics. In *Proceedings of the 1st International Workshop on Action Semantics, Edinburgh, 1994*, number NS-94-1 in BRICS Notes Series, pages 34–42. BRICS, Dept. of Computer Science, Univ. of Aarhus, 1994. BRICSreportNS941.

[41] K. Hansson. Java: Trees versus bytes. Master's thesis, a BComp Honours thesis, 2004.

[42] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Journal of the ACM*, 40(1):143–184, Jan. 1993.

[43] A. Hartmann, W. Amme, J. von Ronne, and M. Franz. Code annotation for safe and efficient dynamic object resolution. In J. Knoop and W. Zimmermann, editors, *Proceedings of Compiler Optimization Meets Compiler Verification (COCV'2003)*, pages 18–32, Warsaw, Poland, Apr. 2003.

[44] IBM Research. *Jikes RVM User's Manual*, v2.0.3 edition, Mar. 2002.

[45] R. Keskar and R. Venugopal. *Compiling safe mobile code. In Compiler Design Handbook: Optimzations and machine code generation*, pages 763–800. CRC Press, 2003.

[46] T. Kistler and M. Franz. Slim binaries. techreport 96-24, Department of Information and Computer Science, University of California, Irvine, June 1996.

[47] T. Kistler and M. Franz. A Tree-Based alternative to Java byte-codes. *International Journal of Parallel Programming*, 27(1):21–34, Feb. 1999.

[48] T. P. Kistler. Continuous program optimization. *PhD Dissertation, University of California, Irvine*, 1999.

[49] T. Lindholm and F. Yellin. *The Java Virtual Machine Specification*. The Java Series. Addison-Wesley, second edition, 1999.

[50] E. Meijer, R. Wa, and J. Gough. Technical overview of the common language runtime. Microsoft, Oct. 2000.

[51] G. Morrisett, K. Crary, N. Glew, D. Grossman, R. Samuels, F. Smith, D. Walker, S. Weirich, and S. Zdancewic. Talx86: a realistic typed assembly language. In *2nd ACM SIGPLAN Workshop on Compiler Support for System Software (WCSSS'99)*, pages 25–35, Atlanta, GA, USA, May 1999.

[52] G. Morrisett, D. Walker, K. Crary, and N. Glew. From System F to typed assembly language. *ACM Transactions on Programming Languages and Systems*, 21(3):527–568, May 1999.

[53] R. Nagy. Menu in activeX controls, Jan. 08 2004.

[54] G. C. Necula. Proof-carrying code. In *Proceedings of the Symposium on Principles of Programming Languages (POPL'1997)*, ACM SIGPLAN Notices, pages 106–119, New York, NY, USA, Jan. 1997. ACM Press.

[55] G. C. Necula. *Compiling with Proofs*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, Sept. 1998. Technical report CMU-CS-98-154.

[56] G. C. Necula and P. Lee. Research on proof-carrying code for untrusted-code security. In *Proceedings of the Conference on Security and Privacy (S&P'1997)*, pages 204–204, Los Alamitos, May 1997. IEEE Computer Society Press.

[57] G. C. Necula and P. Lee. Efficient representation and validation of logical proofs. In *Proceedings of the Annual Symposium on Logic in Computer Science (LICS'1998)*, pages 93–104, Indianapolis, Indiana, June 1998. IEEE Computer Society Press.

[58] G. C. Necula and P. Lee. The design and implementation of a certifying compiler. *SIGPLAN Not.*, 39(4):612–625, 2004.

[59] G. C. Necula and S. P. Rahul. Oracle-based checking of untrusted software. In *POPL '01: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 142–154, New York, NY, USA, 2001. ACM Press.

[60] K. V. Nori, U. Ammann, K. Jensen, N. Nageli, and C. Jacobi. Pascal-P implementation notes. In D. W. Barron, editor, *Pascal – The Language and its Implementation*, pages 125–170. John Wiley & Sons, Ltd., 1981.

[61] OpenGroup. Architecture Neutral Distribution Format (XANDF) Specification. *Open Group Specification P527*, page 206, Jan. 1996.

[62] M. Paleczny, C. A. Vick, and C. Click. The java hotspot^{TM} server compiler. In *Java^{TM} Virtual Machine Research and Technology Symposium*. USENIX, 2001.

[63] E. Schanzer. Performance considerations for run-time technologies in the .net framework. Microsoft technical report, Microsoft Corporation, Aug. 2001.

[64] Z. Shao. An overview of the FLINT/ML compiler. In *Proceeding of the Workshop on Types in Compilation (TIC'1997)*, ACM SIGPLAN Notices, Amsterdam, The Netherlands, June 1997. ACM Press.

[65] Y. Shi, D. Gregg, A. Beatty, and M. A. Ertl. Virtual machine showdown: stack versus registers. In *VEE '05: Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, pages 153–163, New York, NY, USA, 2005. ACM Press.

[66] K. Sohr. *Die Sicherheitsaspekte von mobilem Code*. PhD thesis, Universität Marburg, 2001.

[67] R. F. Stärk, J. Schmid, and E. Börger. *Java and the Java Virtual Machine: Definition, Verification and Validation*. Springer, 2001.

[68] A. Stump and D. L. Dill. Faster proof checking in the Edinburgh Logical Framework. In *Automated Deduction – CADE-18*, volume 2392 of *Lecture Notes in Computer Science*, pages 392–407. Springer-Verlag, July 2002.

[69] T. Thorn. Programming languages for mobile code. *ACM Computing Surveys*, 29(3):213–239, Sept. 1997.

[70] J. von Ronne, W. Amme, and M. Franz. Safetsa: An inherently type-safe ssa-based code format. Technical Report CS-TR-2006-004, Department of Computer Science, The University of Texas at San Antonio, 2006.

[71] T. A. Welch. A technique for high performance data compression. *IEEE Computer Magazine*, 17(6):8–19, June 1984.

[72] M. Wildmoser, A. Chaieb, and T. Nipkow. Bytecode analysis for proof carrying code. In *Proceedings of the Workshop on Bytecode Semantics, Verification, Analysis and Transformation (Bytecode'2005)*, pages 16–30, 2005.

Recent Developments in the Evaluation of Information Retrieval Systems: Moving Towards Diversity and Practical Relevance

Thomas Mandl
Information Science
University of Hildesheim
Marienburger Platz 22, D-31141 Hildesheim, Germany
E-mail: mandl@uni-hildesheim.de

Overview Paper

Keywords: information retrieval, evaluation

Received: July 25, 2007

The evaluation of information retrieval systems has gained considerable momentum in the last few years. Several evaluation initiatives are concerned with diverse retrieval applications, innovative usage scenarios and different aspects of system performance. These evaluation initiatives have led to a considerable increase in system performance. Data for evaluation efforts include multilingual corpora, structured data, scientific documents, Web pages as well as multimedia objects. This paper gives an overview of the current activities of the major evaluation initiatives. Special attention is given to the current tracks and developments within TREC, CLEF and NTCIR. The evaluation tasks and issues, as well as some results, will be presented.

Povzetek: Pregledni članek opisuje usmeritve v informacijskih povpraševalnih sistemih.

1 Information retrieval and its evaluation

Information retrieval is the key technology for knowledge management which guarantees access to large corpora of unstructured data. Very often, text collections need to be processed by retrieval systems. Information retrieval is the basic technology behind Web search engines and an everyday technology for many Web users.

Information retrieval deals with the storage and representation of knowledge and the retrieval of information relevant to a specific user problem. Information retrieval systems respond to queries which are typically composed of a few words taken from a natural language. The query is compared to document representations which were extracted during the indexing phase. The most similar documents are presented to the users who can evaluate the relevance with respect to their information needs and problems.

In the 1960s, automatic indexing methods for texts were developed. They implemented the bag-of-words approach at an early stage, and this still prevails today. Although automatic indexing is widely used today, many information providers and even internet services still rely on human information work.

In the 1970s, research shifted its interest to partial match retrieval models and proved superior compared to Boolean retrieval models. Vector space and later probabilistic retrieval models were developed. However, it took until the 1990s for partial match models to succeed in the market. The Internet accelerated this development. All Web search engines were based on partial match models and provided results as ranked lists rather than unordered sets of documents. Consumers got used to this kind of search system and eventually all big search engines included partial match functionality. However, there are many niches in which Boolean methods still dominate, e.g. patent retrieval. The growing amount of machine-readable documents available requires more powerful information retrieval systems for diverse applications and user needs.

The evaluation of information retrieval systems is a tedious task. Obviously, a good system should satisfy the needs of a user. However, the users' satisfaction requires good performance in several dimensions. The quality of the results with respect to the information need, system speed and the user interface are major dimensions. To make things more difficult, the most important dimension, the level to which the search result documents help the user to solve the information need, is very difficult to evaluate. User-oriented evaluation is extremely difficult and requires many resources. In order to evaluate the individual aspects of searches and the subjectivity of user judgments regarding the usefulness of searches, an impracticable effort would be necessary. As a consequence, information retrieval evaluation experiments try to evaluate only the system. The user is an abstraction and not a real user. In order to achieve that, the users are replaced by objective experts who judge the relevance of a document to one information need. This evaluation methodology is called the Cranfield paradigm, based on the first information retrieval system evaluation in the 1960's (Cleverdon 1997). This is still the evaluation model for modern evaluation initiatives. The first main modern evaluation initiative was the Text Retrieval Conference (TREC). TREC had a huge impact on the field. The emphasis on evaluation in information retrieval research was strengthened. System development and the exchange of ideas was fostered by TREC and systems greatly improved in the first few years.

Recent evaluation efforts try to keep their work relevant for the real world and make their results interesting for practical applications. Yet, in order to cope with these new heterogeneous requirements and to account for the changing necessities of different domains and information needs, new approaches and tasks need to be established.

The remainder of the paper is organized as follows. The next section provides an introduction to the measures commonly used in information retrieval evaluation. Section 3 introduces the basic activities and the history of the three major evaluation initiatives. The following sections present challenges recently taken up in the scientific evaluation of information retrieval systems. They discuss how different document types, multimedia elements and large corpora are introduced. Section 5 points to new developments regarding evaluation methods.

2 Evaluation of information retrieval systems

The information retrieval process is inherently vague. In most systems, documents and queries consist of natural language. The content of the documents needs to be analyzed, which is a hard task for computers. Robust semantic analysis for large text collections or even multimedia objects has yet to be developed. Therefore, text documents are represented by natural language words mostly without syntactic or semantic context. This is often referred to as the bag-of-words approach. These keywords or terms can only imperfectly represent an object because their context and relations to other terms are lost in the indexing process.

Information retrieval systems can be implemented in many ways by selecting a model and specific language processing tools. They interact in a complex system and their performance for a specific data collection cannot be predicted. As a consequence, the empirical evaluation of performance is a central concern in information retrieval research (Baeza-Yates & Ribeiro-Neto 1999). Researchers are faced with the challenge to find measures which can be used to determine whether a system is better than another one (Bollmann 1984).

The most important basic measures are recall und precision. Recall indicates the ability of a system to find relevant documents, whereas precision measures show how good a system is in finding only relevant documents without ballast. Recall is calculated as the fraction of relevant documents found among all relevant documents, whereas precision is the fraction of relevant documents in the result set. The recall requires knowledge of all the relevant documents in a collection that could never be put together in any real world collection. The number of known relevant documents is usually used to calculate the value. Both measures are set oriented, However, most current systems present ranked results. In this case, a recall and precision value pair can be obtained for each position on the ranked list taking into account all documents from the top of the list down to that position. Plotting these values leads to the recall-precision graph. The average of precision values at certain levels of recall is calculated as the mean average precision (MAP), which expresses the quality of a system in one number.

Evaluation initiatives compare the quality of systems by determining the mean average precision for standardized collections and topics like descriptions of information needs. The relevant documents for the topics are assessed by humans who work through all the documents in a pool. The pool is constructed from the results of several systems and ultimately limits the number of relevant documents which can be encountered. Research on the pooling technique has shown that the results are reliable (Buckley & Voorhees 2005).

3 Major evaluation initiatives

The three major evaluation initiatives are historically connected. TREC was the first large effort, which started in 1992. Subsequently, CLEF and NTCIR adopted the TREC methodology and developed specific tasks for multilingual corpora, cross-lingual searches as well as for specific application scenarios.

3.1 Text Retrieval Conference (TREC)

TREC[1] was the first large-scale evaluation initiative and is now in 2008 in its 17th year. TREC is sponsored by the National Institute for Standards and Technology (NIST) in Gaithersburg, Maryland, USA where the annual TREC conference is held. TREC may be considered as the start of a new era in information retrieval research (Voorhees & Buckland 2006). For the first time in information science, TREC achieved a high level of comparability of system evaluations. In the first few years, the effectiveness of the systems approximately doubled. The initial TREC collections for ad-hoc retrieval, which were based on some statement expressing an information need, were newspaper and newswire articles. These test data and collections have stimulated considerable research and are still a valuable resource for development. The model of the user for the ad-hoc evaluation is that of a "dedicated searcher" who is willing to read through hundreds of documents. In the first few years, the topics were very elaborate and long. Starting with TREC 3, the topics became shorter.

TREC has organized the evaluation in 26 tracks which started and ended in different years (Voorhees 2007). Important tracks, apart from the ad-hoc track,

[1] http://trec.nist.gov

were Filtering, Question Answering, Web and Terabyte Track. Other tracks which ran over the last years were the following ones:

- The Question Answering (QA) track requires systems to find short answers to mostly fact-based questions from various domains. In addition to the identification of a relevant document, question-answering systems need to extract the snippet which serves as an answer to the question. In recent years, the QA track is also moving towards more difficult questions like list and definition questions.
- The track with the most participants in 2005 was the Genomics track which combines scientific text documents with factual data on gene sequences (Hersh et al. 2004, Hersh et al. 2006). These tasks attract researchers from the bio-informatics community as well as text retrieval specialists (see also section 3 for domain specific data). The Genomics track ran three times and ended in 2007.
- In the HARD track (High Accuracy Retrieval from Documents), the systems are provided with information on the user and the context of the search. This meta-data needs to be exploited during the retrieval (Allen 2004).
- The Robust Retrieval track applied new evaluation measures which focus on a stable performance over all topics instead of just rewarding systems with good mean average precision (see section 5).
- In the Spam track, which started in 2005, the documents are e-mail messages and the task is to identify spam and non-spam mail. One English and one Chinese corpus need to be filtered. In the Immediate Feedback Track, the system is given the correct class after each message classification and in the Delayed Feedback after a batch of mails. Both tasks simulate a user who gives feedback to a spam filter (Cormack 2006).
- The Blog track, which started in 2006, explores information behavior in large social computing data sets (see section 5.2).
- The Terabyte track can be seen as a continuation of the ad-hoc track and its successor, the Web track. The data collection of almost one terabyte comprises a large and recent crawl of the GOV domain containing information provided by US government agencies. Here, the participants need to scale information retrieval algorithms to large data sets.

```
   <top> <head> Tipster Topic Description
   <num> Number:  051
   <dom> Domain:  International Economics
   <title> Topic:  Airbus Subsidies
   <desc> Description: Document will
discuss government assistance to Airbus
Industrie, or mention a trade dispute
between Airbus and a U.S. aircraft
producer over the issue of subsidies.
   <smry> Summary: Document will discuss
government assistance to Airbus Industrie,
or mention a trade dispute between Airbus
and a U.S. aircraft producer over the
issue of subsidies.
   <narr> Narrative: A relevant document
will cite or discuss assistance to Airbus
Industrie by the French, German, British
or Spanish government(s), or will discuss
a trade dispute between Airbus or the
European governments and a U.S. aircraft
producer, most likely Boeing Co. or
McDonnell Douglas Corp., or the U.S.
government, over federal subsidies to
Airbus.
   <con> Concept(s):
   … </top>

   <top> <num> Number: 400
   <title> Amazon rain forest
   <desc> Description: What measures are
being taken by local South American
authorities to preserve the Amazon
tropical rain forest?
   <narr> Narrative: Relevant documents
may identify: the official organizations,
institutions, and individuals of the
countries included in the Amazon rain
forest; the measures being taken by them
to preserve the rain forest; and
indications of degrees of success in these
endeavors.
   </top>
```

Figure 1: Example Topics from TREC 1 (51) and TREC 7 (400)[2]

TREC continuously responded to ideas from the community and created new tracks. In 2008, the following five tracks are organized at TREC:

- In the Enterprise Track, the participants have to search through the data of one enterprise. The model for this track is intranet search, which is becoming increasingly important. This track started in 2005.
- The Legal track intends to develop effective techniques for legal experts. It was organized for the first time in 2006.
- The large amount of results and submission data has been analysed in many studies. The Million Query Track is a consequence of such evaluation research stimulated by TREC. It was organized for the first time in 2007. Some 10,000 queries from a search engine log were tested against the GOV Web collection (see section 6).[3]
- In 2008, a new Relevance Feedback Track was established.

TREC has greatly contributed to empirically-driven system development and it has improved retrieval systems considerably.

[2] http://trec.nist.gov/data/topics_eng

[3] http://ciir.cs.umass.edu/research/million/

3.2 Cross-language evaluation forum (CLEF)

CLEF[4] is based on the Cross-Language Track at TREC which was organized three years ago (Peters et al. 2005). In 2000, the evaluation of multilingual information retrieval systems moved to Europe and the first CLEF workshop took place. Since then, the ever-growing number of participants has proved that this was the right step. Different languages require other optimization methods in information retrieval. Each language has its own morphological rules for word creation and its words with specific meanings and synonyms. As a consequence, linguistic resources and retrieval algorithms need to be developed for each language. CLEF intends to foster this development.

CLEF closely followed the TREC model for the creation of an infrastructure for research and development. The infrastructure consisted of multilingual document collections comprised of national newspapers from the years 1994, 1995 and 2002. CLEF has been dedicated to include further languages. Document collections for the following languages have been offered over the years: English, French, Spanish, Italian, German, Dutch, Czech, Swedish, Russian, Finnish, Portuguese, Bulgarian and Hungarian.

All topics developed in one year are translated into all potential topic languages. Participants may start with the topics in one language and retrieve documents in another language. CLEF offers more topic languages than document languages. Some languages which attract less research from computational linguistics can be used as topic languages as well. These have included Amharic, Bengali, Oromo and Indonesian over the years.

The participating systems return their results, which are then intellectually assessed. These relevance assessments are always done by native speakers of the document languages (Braschler & Peters 2004). The results from CLEF have led to scientific progress as well as significant system improvement. For example, it could be shown that character n-grams can be used for representing text instead of stemmed terms (McNamee & Mayfield 2004).

Similar to TREC, a question-answering (QA) track has been established, which has attracted many participants. In addition to finding a short answer to a question, the system needs to cross a language border. The language of the query and the document collection are not identical in most cases. Like in the ad-hoc tasks, languages are continuously added. Furthermore, the types of questions are modified. Questions for which no answer can be found in the collection need to be handled properly as well. Temporally restricted questions have also been added (for example, “Where did a meteorite fall in 1908”). A document collection of eight languages has been established as the standard collection.

The number of questions answered correctly is the main evaluation measure. Over the last years, systems have considerably improved. In 2005, six systems reached an accuracy of 40% and two were even able to achieve 60% accuracy. Most experiments submitted are monolingual (61%), bi-lingual experiments reach an accuracy of 10% less than the monolingual. There is a tendency toward applying more elaborated linguistic technologies like deep parsing (Vallin et al. 2005, Leveling 2006).

This performance gap between mono- and cross-lingual retrieval is mainly due to translation errors which lead to non-relevant documents. On the other hand, there are some topics which benefit from the translation. In the target language there might be no synonyms for a topic word leading to a performance decrease in the initial language. Overall, it needs to be said that the variance between topics is typically much larger than the performance difference between systems (Mandl, Womser-Hacker et al. 2008).

In the ImageCLEF track, combined access to textual and graphic data is evaluated (see section 3). The Interactive task (iCLEF) is focused on the user interaction and the user interface. Participants need to explore the differences between a baseline and a contrastive system in a user test setting. The comparison is done only within the runs of one group. The heterogeneity of approaches does not allow for a comparison between groups. In 2004, the interactive track included question answering and in 2005, systems for image retrieval were evaluated in user tests. In the target search task, the user is presented with one image and needs to find it through a keyword search (Gonzalo et al. 2005). In the interactive setting, systems for question answering and image retrieval proved that they are mature enough to support real users in their information needs.

A Web track was installed in 2005 (see section 4) as well as the GeoCLEF track focusing on geographic retrieval (see section 5.1). The tracks Spoken Document Retrieval and Domain Specific are also mentioned in sections below.

For CLEF 2008, library catalogue records from The European Library (TEL) will form a new collection for ad-hoc retrieval. A new filtering task will be established. After a pre-test in 2007, the role of disambiguation in retrieval will be investigated in cooperation with the SemEval Workshop. Participants will receive disambiguated collections and topics and can experiment on ways to use that additional information successfully.

3.3 Asian language initiative NTCIR

NTCIR[5] is dedicated to the specific language technologies necessary for Asian languages and cross-lingual searches among these languages and English (Oyama et al. 2003, Kando & Evans 2007). The institution organizing the NTCIR evaluation is the National Institute for Informatics (NII) in Tokyo where the workshops have been held since the first campaign in 1997. NTCIR takes one and a half years to run an evaluation campaign. In December 2005, the fifth

[4] http://www.clef-campaign.org

[5] http://research.nii.ac.jp/ntcir/

workshop was organized with 102 participating groups from 15 countries. NTCIR is attracting more and more European research groups. NTCIR established a raw data archive which contains the submissions of participants. This will allow long-term research on the results.

The cross-lingual ad-hoc tasks include the three Asian languages Chinese, Japanese and Korean (CJK). Similar to TREC and CLEF, the basic document collections are newspaper corpora. Meanwhile the newspaper collection contains some 1.6 million documents. Overall, the results of the systems are satisfactory and comparable to the performance levels reached at TREC and CLEF; however, the performance between language pairs differs greatly. The fifth workshop emphasized named entity recognition, which is of special importance for Asian language retrieval. In Asian languages, word borders are not marked by blanks like in Western languages. Consequently, word segmentation is not trivial and the identification of named entities is more complicated than in Western languages.

Apart from the ad-hoc retrieval tasks, NTCIR has a patent, a Web and a question-answering track. The general model for the cross-language question-answering task from newspaper data is report writing. It requires processing series of questions belonging together. Patent retrieval focuses on invalidity search and text categorization. The collection has been extended from 3.5 to 7 million documents. Rejected claims from patent offices are used as topics for invalidity search. A set of 1200 such queries has been assembled. Patent search by non-experts based on newspaper articles is also required. A sub-task for passage retrieval aims at more precise retrieval within a document. The number of passages which need to be read until the relevant passage is encountered is the evaluation measure (Oyama et al. 2003).

The Web track comprises a collection of approximately one Terabyte of page data collected from the JP domain. The search task challenges developers to find named pages. This is called a navigational task because users often search for homepages or named pages in order to browse to other pages from there.

4 Document types

TREC started to develop collections for retrieval evaluation based on newspaper and news agency documents. This approach has been adopted by CLEF and NTCIR because newspaper articles are easily available in electronic formats, they are homogeneous, no domain experts are necessary for relevance assessments and parallel corpora from different newspapers, dealing with the same stories, can be assembled. Nevertheless, this approach has often been criticized because it was not clear how the results gained from newspaper data would generalize to other kinds of data. Especially domain-specific texts have other features than newspaper data and the vocabularies used are quite different across domains. The focus on newspapers seemed to make evaluation results less reliable and relevant for other realistic settings.

As a consequence, many other collections and document types have been integrated into evaluation collections throughout recent years. These include structured documents and multimedia data which are discussed in the following sections.

An important step was the establishment of the domain specific track at CLEF where systems can be evaluated for domain specific data in mono- and multi-lingual settings for German, English and Russian. The collection is based on the GIRT (German Indexing and Retrieval Test Database) corpus from the social sciences, containing English and German abstracts of scientific papers (Kluck 2004). At TREC, the demand for bio-informatics led to the integration of the Genomics track where genome sequences and text data are combined. The new legal track at TREC is also dedicated to domain experts. The patent retrieval task at NTCIR requires the optimization for the text type patent. For all these domain specific tasks, the special vocabularies and other characteristics need to be considered in order to achieve good results.

4.1 Structured documents

Newspaper stories have a rather simple structure. They contain a headline, an abstract and the text. In many applications, far more numerous and complex document structures need to be processed by information retrieval systems.

The inclusion of Web documents into evaluation campaigns has been a first step to integrate structure. Web documents written in HTML have very heterogeneous structures and only a small portion is typically exploited by retrieval systems. The HTML tag title is most often used for specific indexing, but headlines and links texts are used, too.

One initiative is specifically dedicated to the retrieval from documents structured with XML. INEX[6] (Initiative for the Evaluation of XML Retrieval) started in 2002 and is annually organized by the University of Duisburg-Essen in Germany. The topics are based on information needs and as such, cannot be solved merely by XML database retrieval. The challenge for the participants lies in tuning their systems such that they do not only retrieve relevant document parts, but the smallest XML element which fully satisfies the information need (Fuhr 2003). The need to exploit structure has attracted many database research groups to INEX. The test collection within INEX includes several computer science bibliography and paper collections as well as the Lonely Planet travel guide books which exhibit a rich structure and even contain pictures. Based on these pictures, a multimedia track has been established at INEX.

[6] http://inex.is.informatik.uni-duisburg.de/

4.2 Multimedia data

Multimedia data is becoming very important and most search engines already provide some preliminary form of image retrieval. Research has been exploring the algorithms for content based multimedia retrieval but is still struggling with the so-called "semantic gap" (Mittal 2006). Systems cannot yet make the step from atomic features of an image, like the color of a pixel, to the level of an object which a human would recognize. Evaluation campaigns are integrating multimedia data in various forms into their efforts.

The track ImageCLEF began in 2003 and explores the combination of visual and textual features in cross-lingual settings. Images seem to be language independent, but they often have associated text (e.g. captions, annotations). ImageCLEF assembled collections of historic photographs and medical images (radiographs, photographs, power-point slides). For the historic photographs, ad-hoc retrieval is performed and the topics are motivated by a log-file analysis from an actual image retrieval system (for example "waves breaking on the beach", "a sitting dog"). Visual as well as textual topics were developed and some topics contain both text and images. In contrast to other tasks at CLEF, where usually binary assessments are required, ternary relevance assessment is carried out by three assessors at ImageCLEF. The best systems reach some 0.4 mean average precision; however, performance varies greatly among languages (Clough et al. 2005).

For the medical images, retrieval and annotation is required. Medical doctors judged the relevance of the images for the information need. For the automatic annotation task, images needed to be classified into 57 classes identifying, for example, the body region and image type.

In addition, ImageCLEF introduced an interactive image retrieval task in cooperation with the Interactive track to investigate the interaction issues of image retrieval. It could be shown that relevance feedback improved results similarly to ad-hoc retrieval.

In 2001, a video track started up and ran again in 2002. Starting in 2003, the evaluation for video retrieval established an independent evaluation campaign called TRECVid[7] (TREC Video Retrieval Evaluation). In 2005, TRECVid concentrated on four tasks:

- Shot boundary determination: systems need to detect meaningful parts within video data.
- Low-level feature extraction: systems need to recognize whether camera movement appears in a scene (pan, tilt or zoom)
- High-level feature extraction: ten features from a Large Scale Concept Ontology for Multimedia (LSCOM) were selected and systems need to identify their presence in video scenes. The ontology includes cars, explosions and sports.
- Search tasks include interactive, manual, and automatic retrieval. Examples of topics are: "Find shots of fingers striking the keys on a keyboard which is at least partially visible" and "Find shots of Boris Yeltsin".

The data collection includes 170 hours of television news in three languages (English, Arabic and Chinese) from November 2004 collected by the Linguistic Data Consortium[8] (LDC), some hours of NASA educational programs and 50 hours of BBC rushes on vacation spots (Smeaton 2005). Considerable success has been achieved by applying speech recognition to the audio track of a video and by running standard text retrieval techniques to the result. On the other hand, content-based techniques for the visual data still require much research to bridge the semantic gap.

Apart from visual data, retrieval of audio data has also attracted considerable research. At CLEF, a Cross-Language Spoken Document Retrieval (CL-SR) track has been running since 2003. In 2005, the experiments were based on the recordings of interviews with Holocaust survivors (Malach collection). The interviews last 750 hours and are provided as audio files and as transcripts of an automatic speech recognition (ASR) system. Participants may base the retrieval on their own ASR or use the transcript provided. The data was tagged by humans who added geographical and other terms. For the retrieval test, interviews in Czech and English are provided. The retrieval systems need to be optimized for the partially incorrect output of the ASR (Oard et al. 2006).

Even for music retrieval, an evaluation campaign has been established. The Music Information Retrieval Evaluation eXchange (MIREX[9]) focuses on content-based music data processing. The tasks include query by humming, melody extraction and music similarity (Downie 2003, Downie et al. 2005).

5 Specific user needs

Focusing on very specific user needs makes evaluation more real-world-oriented and increases its value for that specific application area. Each application has its own particular character. While some users work on a recall oriented basis (patent attorneys), others focus on precision (web users). Many users want all aspects of a topic to be represented in the result set independent of the number of retrieved relevant documents. This aspect has been evaluated in the Genomics Track (Hersh et al. 2006) and has previously been researched in the Novelty Track.

5.1 Geographic information retrieval

In GeoCLEF[10], systems need to retrieve news stories with a geographical focus. GeoCLEF is a modified ad hoc retrieval task, involving both spatial and multilingual aspects based on newspaper collections previously offered at CLEF (Gey et al. 2007, Mandl, Gey, et al.

[7] http://www.itl.nist.gov/iaui/894.02/projects/trecvid/

[8] http://www.ldc.upenn.edu/

[9] http://www.music-ir.org/mirex2006/

[10] http://www.uni-hildesheim.de/geoclef/

2008). Examples of topics are “shark attacks off California and Australia” or “wind power on Scottish Islands”. In order to master the last topic, the systems need knowledge of what the Scottish Islands are. For other topics, it is necessary to include symbolic knowledge about the inclusion of one geographical region within another. Participants applied named entity identification for geographical names and used geographical knowledge sources like ontologies and gazetteers. However, standard approaches outperformed specific geographical tools in the first two editions 2005 and 2006 and still perform similarly in 2007. This might be due to the fact that standard approaches like blind relevance feedback lead to results similar to geographical reasoning systems.

For 2007, the topics were developed to include more challenging aspects. Ambiguity, vaguely defined geographic regions (Near East) and more complex geographical relations were emphasized (Mandl, Gey, et al. 2008).

5.2 Opinion retrieval

Social Software applications allow users to create and modify Web pages very easily. Such systems enable users to quickly publish content and share it with other users. The success of social systems encouraged millions of users to become members of social networks and led to the creation of a large amount of user-generated content.

Users create huge amounts of text in blogs which can be simplistically described as online diaries with comments and discussions. Many blogs contain personal information; others are dedicated to specific topics. The huge interest in blogs has also led to blog spam. In order to explore searching in blogs, TREC initiated a blog track in 2006. A collection was created by crawling well-known blog locations on the Web. More than 3.2 million documents, in this case blog entries from more than 100,000 blogs, were collected (Ounis et al. 2006).

One of the most interesting and blog-specific issues is the subjective nature of the content. It is very likely to find opinions on topics. Companies e.g. are beginning to exploit blogs by looking for opinions on their products. Consequently, a very natural retrieval task regarding blogs is the retrieval of opinions on a given topic.

Typical approaches for opinion retrieval include list-based and machine learning approaches. List-based methods rely on large lists of words of a subjective nature. Their occurrence in a text is seen as an indicator of opinionated writing. Machine learning methods are trained on typically objective texts like online lexical documents and on subjective texts like product review sites. Systems learn to identify texts with opinions based on features like individual words, the number of pronouns or adjectives.

The opinion retrieval task in the blog track at TREC was based on relevance assessment at several levels. The typical relevant and non-relevant judgments were supplemented by explicitly negative, explicitly positive and both positive and negative judgments. The subjective documents were well balanced in the pool. The document pool contained 2% spam blog posts, showing that spam is a problem. The variance among topics is very large. However, systems managed to retrieve spam documents more likely at later ranks rather than on earlier ranks. Interestingly, opinion finding and relevance scores of the systems correlate substantially. The opinion finding scores are higher than the topic relevance scores overall (Ounis et al. 2006).

The idea of opinion analysis is considered at NTCIR as well. For NTCIR-7, a track for Cross-Language Information Retrieval for Blogs (CLIRB) and a track for Multilingual Opinion Analysis Task (MOAT) are envisioned.

User-created content is also becoming a subject for CLEF. The interactive track at CLEF 2008 intends to investigate how users use the picture-sharing platform FlickR to search for images in a multilingual way.

6 Large corpora

Information Retrieval is faced with new challenges on the Web. The mere size of the Web forces search engines to apply heuristics, in order to find a balance between efficiency and effectiveness. One example for a heuristic would be to only index significant parts of each document. The dynamic nature of the Web makes frequent crawling necessary and creates the need for efficient index update procedures. One of the most significant challenges of the Web is the heterogeneity of the documents in several respects. Web pages vary greatly in length, document format, intention, design and language. These issues have been dealt with in evaluation initiatives.

The Web Track at TREC ran from 1999 until 2004. In its last edition, it attracted 74 runs (Craswell & Hawking 2004). The Web corpus used at TREC had a size of 18 GB and was created by a crawl of the GOV domain, containing US government information. This track is focused on retrieval of Web pages in English. Similarly, a Chinese Web Evaluation Initiative organized by Beijing University is focused on the Chinese Web . The document collection crawled from the CN domain contains some 100 Gigabyte . The tasks for the systems are named page finding, home page finding and an information ad-hoc task based on topics selected from a search engine log. NTCIR also includes a Web collection for retrieval evaluation, based on a collection of one Terabyte of document data from the Japanese Web.

The task design for Web retrieval evaluation in evaluation initiatives is oriented towards navigational information needs (Broder 2002) and known item finding tasks. As such, these evaluations differ from ad-hoc retrieval, where an informational need is the model for the topics developed. The main difference between the two search types is that the navigational information needs aims for one specific Web page (homepage or another page) which the user might even have visited before. In contrast, the informational task aims at finding pages on a certain topic to satisfy a certain information need. In these cases, it is not known how many potential target pages exist. The pooling technique is not necessary for navigational information needs. On the contrary, for informational search tasks, the quality of the pooling technique needs to be re-evaluated. The quality and depth of the pool from which the relevant documents will be extracted by human assessors cannot be judged. The effect of this fact on the evaluation results needs to be assessed. Consequently, most evaluation tracks for Web retrieval remain restricted to navigational information needs.

Most Web retrieval tracks include mainly navigational information needs. This may be partially due to the need for many resources to create relevance judgments for informational Web search tasks.

The results of the TREC Web track indicate that the use of Web-specific knowledge of document structure and anchor text positively affects retrieval quality. The contribution of link structure and URL length is less obvious. Typical information retrieval techniques like stemming do not seem to be necessary for Web retrieval (Craswell & Hawking 2004).

At TREC 2002, a navigational task as well as a topic distillation task were offered. Both led to different results. For navigational tasks, link analysis led to better results whereas link analysis could not improve topic distillation (Craswell & Hawking 2003:6).

```
   <title> highway safety
   <desc> Description:
   Find documents related to improving
highway safety in the U.S.
   <narr> Narrative:
Relevant documents include those related
to the improvement of safety of all
vehicles driven on highways, including
cars, trucks, vans, and tractor trailers.
Ways to reduce accidents through
legislation, vehicle checks, and drivers'
education programs are all relevant.
```

Figure 2: Example for a Topic of the TREC Web Track 2002 (Craswell & Hawking 2003)

A new Terabyte Track at TREC is based on a crawl of the domain GOV and contains more than 400 Gigabyte of document data. The topics are developed from ad-hoc type information needs. The goal of this track is scaling the systems and evaluation methodology to a large-size collection. It is expected that the pool and the relevance assessments will be dramatically less complete than for newspaper collections for ad-hoc retrieval. The effects of this problem for evaluation methods are being investigated (Clarke et al. 2004).

Another solution to this problem lies in the development of more topics. A statistical analysis of TREC results modified the number of topics and used different amounts of the relevance assessment available (Sanderson & Zobel 2005). It revealed that more topics and shallow pools led to more reliable results than deep relevance assessments for fewer topics. Fewer relevance judgments could diminish the cost of evaluation campaigns drastically. A new step in this direction is the so-called Million Query for TREC 2007 where this finding will be exploited. Some 10,000 queries from a search engine log were tested against the GOV Web collection. Relevance assessment will focus on a subset of a few hundred queries and it will consider 40 or more documents per topic.

At NTCIR-4, an informational retrieval task was organized which has been dropped at NTCIR 5. A navigational task was part of NTCIR 3 through NTCIR-5 (Eguchi et al. 2004). For the informational retrieval task, the pooling problem was addressed at NTCIR. Shallow vs. deep pooling was compared. For all topics, pooling with the top ten documents was carried out and for a subset, the top 100 documents of pooled runs were used and additional techniques were used to extend the pool (Eguchi et al. 2004). The pooling levels were mapped to different user models in Web search. The results varied between the two methods to a considerable extent. Another Web specific parameter in the evaluation was the document model behind the relevance assessment. The information unit can be the page itself or pages to which it directly links. This document model considers the hub function of pages, which is often highly valuable for informational search tasks.

The Web is obviously a very natural environment for multilingual retrieval. Users have many different native languages and for each user, most of the information on the Web is not in his or her native language. In 2005, a new Web track was established at CLEF focusing on the challenges of multi-linguality. Similar to the corpus used at TREC, European government sites were crawled and included in the collection. Unlike the TREC GOV collection, which is mainly English, and the NTCIR collection, which is English and Japanese, the EuroGOV collection contains pages in more than 25 languages (Sigurbjörnsson et al. 2005)[11]. Many pages are even multilingual (Artemenko et al. 2006). The multilingual corpus of Internet pages was engineered by the University of Amsterdam. The web crawl collected pages of official institutions (mainly ministries) in European countries. It covers 27 domains and contains 3.6 million Web pages. The documents are written in some 25 languages. The size of the corpus is some 100 Gigabyte. Together with the participants, the track organizers were able to create 575 topics for homepage and named page finding in the first year. The tasks offered were mixed-mono-lingual (many queries of different languages being submitted to one search engine), bi-lingual (retrieve English documents based on Spanish queries) and truly multi-lingual where the language of the target was not specified (Sigurbjörnsson et al. 2006).

The performance for the mixed-mono task is comparable to mono-lingual ad-hoc results; however, the performance for both cross-lingual tasks lacks far behind. There is a great need for further research. The first year of work on the Web task led to surprising results. Whereas the automatic translation of topics is the main approach to bridge the language gap in ad-hoc retrieval, translation harmed performance for the Web topics. This

[11] http://ilps.science.uva.nl/WebCLEF/

may be due to the reason that the Web task is focused on homepage and named page finding.

7 Evaluation measures

The initiatives adhered to the traditional evaluation measures mentioned in the second section of this paper. They assumed that there is a valid concept of the quality of a system, which can be assessed by several strongly correlating measures. However, the large-scale evaluations themselves have stirred interest in these basic questions of evaluation.

It has often been pointed out that the variance between queries is larger than the variance between systems. There are often very difficult queries. Few systems solve these well and they lead to very bad results for most systems (Harman & Buckley 2004). Thorough failure analysis can lead to substantial improvement. For example, the absence of named entities is a factor which can generally make queries more difficult (Mandl & Womser-Hacker 2004). It is also understood that the requests which are answered poorly will strongly contribute toward any negative feelings of the user.

As a consequence, a new evaluation track for robust retrieval has been established by the Text Retrieval Conference (TREC). Robustness can be seen as the capacity of a system to perform well under heterogeneous conditions. The robust track not only measures the average precision over all queries, but also emphasizes the performance of the systems for difficult queries. In order to perform well in this track, it is more important for the systems to retrieve at least a few documents for difficult queries than to improve the performance on average (Voorhees 2005). To allow for a system evaluation based on robustness, more queries are necessary than for a normal ad-hoc track. The score per system is not calculated by the arithmetic mean of all topics, but by the geometric mean. The geometric mean reduces the influence of topics which were solved with very good results. The concept of robustness is extended in TREC 2005. Systems need to perform well over different tracks and tasks (Voorhees 2005). For multilingual retrieval, robustness is also an interesting evaluation concept because the performance between queries differs greatly. The issue of stable performance over all topics instead of high average performance has been explored at CLEF 2006 for six languages (Di Nunzio et al. 2007). For the top systems, a high correlation between standard and robust measures was found. However, further analysis revealed that the robust measures lead to very different results with a growing number of topics, especially if the percentage of low performing topics is high. Because this is the case in multi-lingual retrieval settings, robust evaluation is of high importance for multi-lingual technology (Mandl, Womser-Hacker et al. 2008).

For several other tasks, the traditional measures have been considered to be inadequate. For the Web tasks, for example, Web-user-oriented measures were sought. For the navigational tasks, the mean reciprocal rank of the target item was established. For informational tasks, early precision measures were used. Often, the precision at ten documents is used. The recall power of a system can be neglected when taking into consideration an underlying user model with the average Web user who is seeking only a few hits.

One concern about evaluations of large collections is the percentage of judged documents. The effort spent on relevance assessment remains constant. As collections grow, only a small fraction of documents is actually being assessed by humans. Many of the documents retrieved by systems are not judged. These documents are considered as not relevant. Results might be unreliable because most documents in the result lists are not judged. A new measure has been proposed and meanwhile adopted for most experiments with large collections. The binary preference (Bpref) metric takes only documents into account which were judged by a human juror. They are disregarded and the new measure checks how many times a system retrieves a relevant document before a document is judged as not relevant (Buckley & Voorhees 2004).

Still, it remains unclear how evaluation results relate to user satisfaction. For a small experiment with simple search tasks, no correlation between evaluation measures and user satisfaction was found (Turpin & Scholer 2006). The relation between system performance and the perception of the user needs to be the focus of more research.

Many novel retrieval measures have been developed in the past years. Nevertheless, the classic measures are still being widely used. Overall, there is a consensus that these new measures might reveal something important that is not covered by recall and precision. However, it is not yet well understood what this “something” is (Robertson 2006).

8 Summary

As this overview shows, the evaluation of information retrieval systems has greatly diversified in recent years. Research has recognized that evaluation results from one domain and one application cannot be transferred to other domains. Evaluation campaigns need to continuously re-consider their tasks, topics and evaluation measures, in order to make them as similar to real-world tasks and information needs as possible.

In the future, the diversification will continue as further tasks are being explored. The evaluations of multimedia data and of Web resources are likely to converge because more and more multimedia data is available on the Web. Further evaluation initiatives are being established. In 2008, the new Indian evaluation campaign FIRE (Forum for Information Retrieval Evaluation) will run for the first time and provide test environments for the major languages spoken in India .

9 References

[1] Allen, James (2004). HARD Track Overview in TREC 2004 High Accuracy Retrieval from Documents. In: Buckland, Lori; Voorhees, Ellen

(Eds.). *The Thirteenth Text Retrieval Conference (TREC 2004)* NIST Special Publication: SP 500-261. http://trec.nist.gov/pubs/trec13/t13_proceedings.html

[2] Artemenko, Olga; Mandl, Thomas; Shramko, Margaryta; Womser-Hacker, Christa (2006). Evaluation of a Language Identification System for Mono- and Multi-lingual Text Documents. In: *Proceedings of 2006 ACM SAC Symposium on Applied Computing (SAC)* April, 23-27, 2006, Dijon, France. pp. 859-860.

[3] Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier (1999). Retrieval Evaluation. In: Baeza-Yates, R.; Ribeiro-Neto, B. (eds.): *Modern Information Retrieval*. Addison-Wesley. pp. 73-97.

[4] Bollmann, Peter (1984). Two Axioms for Evaluation Measures in Information Retrieval. In: *Proceedings of the Third Joint BCS/ACM Symposium on Research and Development in Information Retrieval (SIGIR 1984)* Cambridge, 2-6 July 1984. pp. 233-245

[5] Braschler, Martin; Peters, Carol (2004). Cross-Language Evaluation Forum: Objectives, Results, Achievements. In: *Information Retrieval* no. 7. pp. 7-31.

[6] Broder, Andrei (2002). A taxonomy of web search. In: *ACM SIGIR Forum* vol. 36(2) pp. 3–10.

[7] Buckley, Chris; Voorhees, Ellen (2005): Retrieval System Evaluation. In: *TREC: Experiment and Evaluation in Information Retrieval*. Cambridge & London: MIT Press. pp. 53-75.

[8] Clarke, Charles; Craswell, Nick; Soboroff, Ian (2004). Overview of the TREC 2004 Terabyte Track. In: Buckland, Lori; Voorhees, Ellen (Eds.). *The Thirteenth Text Retrieval Conference (TREC 2004)* NIST Special Publication: SP 500-261. http://trec.nist.gov/pubs/trec13/t13_proceedings.html

[9] Cleverdon, Cyril (1997). The Cranfield Tests on Index Language Devices. In: Sparck-Jones, Karen; Willett, Peter (Eds.): *Readings in Information Retrieval*. Morgan Kaufman. pp. 47-59.

[10] Clough, Paul; Müller, Henning; Deselaers, Thomas; Grubinger, Michael; Lehmann, Thomas; Jensen, Jeffery; Hersh, William (2005). The CLEF 2005 Cross-Language Image Retrieval Track. In: *Working Notes of the 6th Workshop of the Cross-Language Evaluation Forum, CLEF*. Sep. 2005, Vienna, Austria. http://www.clef-campaign.org/

[11] Cormack, Gordon (2006). TREC 2006 Spam Track Overview. In: Voorhees & Buckland (2006)

[12] Craswell, Nick; Hawking, David; Wilkinson, Ross; Wu, Mingfang (2004). Overview of the TREC 2003 Web Track. In: *Proceedings Text Retrieval Conference (TREC).* http://trec.nist.gov/pubs/trec12/t12_proceedings.html

[13] Craswell, Nick; Hawking, David (2004). Overview of the TREC-2004 Web Track. In: Voorhees & Buckland 2004.

[14] Downie, Stephan (2003). Toward the Scientific Evaluation of Music Information Retrieval Systems. In: *Intl Symposium on Music Information Retrieval (ISMIR)* Washington, D.C., & Baltimore, USA. http://ismir2003.ismir.net/papers/Downie.PDF

[15] Downie, Stephen; West, Kris; Ehmann, Andreas; Vincent, Emmanuel (2005). The 2005 Music Information retrieval Evaluation Exchange (MIREX 2005): Preliminary Overview. In: *6th International Conference on Music Information Retrieval (ISMIR)* London, UK, 11-15 Sept. pp. 320-323.

[16] Eguchi, Koji; Oyama, Keizo; Aizawa, Akiko; Ishikawa, Haruko (2004). Overview of the Informational Retrieval Task at NTCIR-4 WEB. In: *NTCIR Workshop 4 Meeting Working Notes.* http://research.nii.ac.jp/ntcir-ws4/NTCIR4-WN/index.html

[17] Fuhr, Norbert (2003). Initiative for the Evaluation of XML Retrieval (INEX): *INEX 2003 Workshop Proceedings,* Dagstuhl, Germany, December 15-17. http://purl.oclc.org/NET/duett-07012004-093151

[18] Gey, Fredric; Larson, Ray; Sanderson, Mark; Bischoff, Kerstin; Mandl, Thomas; Womser-Hacker, Christa; Santos, Diana; Rocha, Paulo; Di Nunzio, Giorgio; Ferro, Nicola (2007). GeoCLEF 2006: the CLEF 2006 Cross-Language Geographic Information Retrieval Track Overview. In: Peters, Carol et al. (Eds.). *Evaluation of Multilingual and Multi-modal Information Retrieval. 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006*, Alicante, Spain, Revised Selected Papers. Berlin et al.: Springer [LNCS 4730] pp. 852-876.

[19] Gonzalo, Julio; Clough, Paul; Vallin, Alessandro (2006). Overview of the CLEF 2005 Interactive Track In: Peters, Carol; Gey, Fredric C.; Gonzalo, Julio; Jones, Gareth J.F.; Kluck, Michael; Magnini, Bernardo; Müller, Henning; Rijke, Maarten de (Eds.). *Accessing Multilingual Information Repositories: 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, Vienna, Austria, Revised Selected Papers. Berlin et al.: Springer [LNCS 4022] pp. 251-262.

[20] Harman, Donna; Buckley, Chris (2004). The NRRC reliable information access (RIA) workshop. In: *Proceedings of the 27th annual international conference on Research and development in information retrieval (SIGIR)*. pp. 528-529.

[21] Hersh, William; Bhuptiraju, Ravi; Ross, Laura; Johnson, Phoebe; Cohen, Aaron; Kraemer, Dale (2004). TREC 2004 Genomics Track Overview. In: Buckland, Lori; Voorhees, Ellen (Eds.). *The Thirteenth Text Retrieval Conference (TREC 2004)* NIST Special Publication: SP 500-261. http://trec.nist.gov/pubs/trec13/t13_proceedings.html

[22] Hersh, William; Cohen, Aaron; Roberts, Phoebe; Rekapalli, Hari Krishna (2006). TREC 2006 Genomics Track Overview. In: Voorhees & Buckland (2006)

[23] Kando, Noriko and Evans, David (2007). *Proceedings of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*. National Institute of Informatics, Tokyo, Japan. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings6/NTCIR/index.html

[24] Kluck, Michael (2004). The GIRT Data in the Evaluation of CLIR Systems - from 1997 until 2003. In: *Comparative Evaluation of Multilingual Information Access Systems: 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003,* Trondheim, Norway. Revised Selected Papers. Springer: LNCS 3237. pp. 376-390

[25] Leveling, Johannes (2006). A baseline for NLP in domain-specific information retrieval. In: Peters, Carol et al. (eds): *Accessing Multilingual Information Repositories: 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, Vienna, Austria, Revised Selected Papers Berlin: Springer [LNCS 4022] pp. 222-225.

[26] Mandl, Thomas; Womser-Hacker, Christa (2005). The Effect of Named Entities on Effectiveness in Cross-Language Information Retrieval Evaluation. In: *Proceedings ACM SAC Symposium on Applied Computing (SAC)*. Santa Fe, New Mexico, USA. March 13.-17. pp. 1059-1064.

[27] Mandl, Thomas; Gey, Fredric; Di Nunzio, Giorgio; Ferro, Nicola; Larson, Ray; Sanderson, Mark; Santos, Diana; Womser-Hacker, Christa; Xing, Xie (2008). GeoCLEF 2007: the CLEF 2007 Cross-Language Geographic Information Retrieval Track Overview. In: Peters, Carol et al. (Eds.). *8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007,* Budapest, Hungary, Revised Selected Papers. Berlin et al.: Springer [LNCS] to appear. Preprint at: http://www.clef-campaign.org

[28] Mandl, Thomas; Womser-Hacker, Christa; Ferro, Nicola; Di Nunzio, Giorgio (2008). How Robust are Multilingual Information Retrieval Systems? In: *Proceedings ACM Symposium on Applied Computing (SAC)* Fortaleza, Brazil. pp. 1132-1136.

[29] McNamee, Paul; Mayfield, James (2004). Character N-Gram Tokenization for European Language Text Retrieval. In: *Information Retrieval*, vol. 7 (1/2). pp. 73-98.

[30] Mittal, Ankush (2006). An Overview of Multimedia Content-Based Retrieval Strategies. In: *Informatica* 30. pp. 347-356.

[31] Di Nunzio, Giorgio; Ferro, Nicola; Mandl, Thomas; Peters, Carol (2007). CLEF 2006: Ad Hoc Track Overview. In: Peters, Carol et al. (Eds.). *Evaluation of Multilingual and Multi-modal Information Retrieval. 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006*, Alicante, Spain, Revised Selected Papers. Berlin et al.: Springer [LNCS 4730] pp. 21-34.

[32] Di Nunzio, Giorgio; Ferro, Nicola; Mandl, Thomas; Peters, Carol (2008). CLEF 2007: Ad Hoc Track Overview. In: Peters, Carol et al. (Eds.). *8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007*, Budapest, Hungary, Revised Selected Papers. Berlin et al.: Springer [LNCS] to appear. Preprint: http://www.clef-campaign.org

[33] Robertson, Stephan (2006). On GMAP: and other transformations. In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)* Arlington, Virginia, USA. pp. 872-877

[34] Sanderson, Mark; Zobel, Justin (2005). Information retrieval system evaluation: effort, sensitivity, and reliability. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2005)* Salvador, Brazil. ACM Press. pp. 162 - 169.

[35] Sigurbjörnsson, Börkur; Kamps, Jaap; Rijke, Maarten de (2006). Overview of WebCLEF 2005. In: Peters, Carol; Gey, Fredric C.; Gonzalo, Julio; Jones, Gareth J.F.; Kluck, Michael; Magnini, Bernardo; Müller, Henning; Rijke, Maarten de (Eds.). *Accessing Multilingual Information Repositories: 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, Vienna, Austria, Revised Selected Papers. Berlin et al.: Springer [LNCS 4022] pp. 810-824.

[36] Sigurbjörnsson, Börkur; Kamps, Jaap; de Rijke, Maarten (2005). Blueprint of a Cross-Lingual Web Retrieval Collection. In: Journal of Digital Information Management, vol. 3 (1) pp. 9-13.

[37] Smeaton, Alan (2005). Large Scale Evaluations of Multimedia Information Retrieval: The TRECVid Experience. In: CIVR 2005 – International Conference on Image and Video Retrieval, Springer: LNCS 3569, pp 11-17.

[38] Turpin, Andrew; Scholer, Falk (2006). User performance versus precision measures for simple search tasks. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, Seattle, Washington, USA, August 6-11. ACM Press. pp. 11-18.

[39] Oard, Douglas W.; Wang, Jianqiang; Jones, Gareth; White, Ryen; Pecina, Pavel; Soergel, Dagobert; Huang, Xiaoli; Shafran, Izhak (2006). Overview of the CLEF-2006 Cross-Language Speech Retrieval Track. In: Nardi, Alessandro; Peters, Carol; Vicedo, José Luis (Eds.): *CLEF 2006 Working Notes*. http://www.clef-campaign.org/2006/working_notes

[40] Ounis, Iadh; Rijke, Maarten; Macdonald, Craig; Mishne, Gilad; Soboroff, Ian (2006). Overview of the TREC-2006 Blog Track. In: Voorhees & Buckland (2006)

[41] Oyama, Keizo; Ishida, Emi; Kando, Noriko (2002) (eds.). *NTCIR Workshop 3: Proceedings of the Third NTCIR Workshop on research in Information Retrieval, Automatic Text Summarization and Question Answering* (Sept 2001-Oct 2002) http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/index.html

[42] Vallin, Alessandro; Giampiccolo, Danilo; Aunimo, Lili; Ayache, Christelle; Osenova, Petya; Peñas,

Anselmo; de Rijke, Maarten; Sacaleanu, Bogdan; Santos, Diana; Sutcliffe, Richard (2006). Overview of the CLEF 2005 Multilingual Question Answering Track. In: Peters, Carol; Gey, Fredric C.; Gonzalo, Julio; Jones, Gareth J.F.; Kluck, Michael; Magnini, Bernardo; Müller, Henning; Rijke, Maarten de (Eds.). *Accessing Multilingual Information Repositories: 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005,* Vienna, Austria, Revised Selected Papers. Berlin et al.: Springer [LNCS 4022] pp. 307-331.

[43] Voorhees, Ellen; Buckland, Lori (2006) (eds.). *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)* NIST Special Publication SP 500-272. National Institute of Standards and Technology. Gaithersburg, Maryland. Nov. 2006. http://trec.nist.gov/pubs/trec15/

[44] Voorhees, Ellen (2005). The TREC robust retrieval track. In: *ACM SIGIR Forum* 39 (1) pp. 11-20.

[45] Voorhees, Ellen (2006). Overview of TREC 2006. In: Voorhees & Buckland (2006)

Semantic Grid Platform in Support of Engineering Virtual Organisations

Matevž Dolenc, Robert Klinc and Žiga Turk
University of Ljubljana, Faculty of Civil and Geodetic Engineering,
Jamova 2, SI-1000 Ljubljana, Slovenia
E-mail: {mdolenc, zturk, rklinc}@itc.fgg.uni-lj.si

Peter Katranuschkov
Technical University of Dresden, Mommsenstr. 13, 01062 Dresden, Germany
E-mail: peter.katranuschkov@cib.bau.tu-dresden.de

Krzysztof Kurowski
Poznan Supercomputing and Networking Center, Noskowskiego 10 Street, 61-704 Poznan, Poland
E-mail: kikas@man.poznan.pl

Keywords: virtual organisation, interoperability, service oriented architecture, ontologies, semantic grid, engineering software, InteliGrid

Received: January 15, 2008

The EU project InteliGrid (2004-2007) combined and extended the state-of-the-art research and technologies in the areas of semantic interoperability, virtual organisations and grid technology to provide diverse engineering industries with a collaboration platform for flexible, secure, robust, interoperable, pay-per-demand access to information, communication and processing infrastructure. This paper describes the system architecture and the technical aspects of the developed platform as well as the key components it offers, including services for document management, access to product model servers and utilisation of high-performance computing infrastructure.

Povzetek: Predstavljena je semantična grid platforma za podporo inženirskim virtualnim organizacijam.

1 Introduction

Grids are generally known as infrastructures for high performance computing. However, the original idea behind grid computing was to support collaborative problem solving in virtual organizations (VO). This coincides with the EU project InteliGrid (2004-2007) vision: to provide complex industries with challenging integration and interoperability needs (such as automotive, aerospace and construction) a flexible, secure, robust, ambient accessible, interoperable, pay-per-demand access to (1) information, (2) communication and (3) processing infrastructure.

The isolation and lack of interoperability of software applications - identified in the late 1980s as the *islands of automation* problem [1] - is well known in various industries. The term was largely used during the 1980s to describe how rapidly developing automation systems were at first unable to communicate easily with each other. Industrial communication protocols, network technologies, and system integration helped to improve this situation. A number of European projects such as ATLAS [2], COMBI [3], COMBINE [4], ToCEE [5], ISTforCE [6], OSMOS [7] and others have proven theoretically and by developed prototypes that interoperability based on product data technology is achievable and can provide many benefits to the industry. Nevertheless, solutions for the practice require comprehensive environments that must incorporate coherently all aspects of interoperability. This is the reason for rare use of such solutions in the industry despite all research and development efforts. However, most of the necessary technology for solving this problem, particularly the standards and tools for interoperability, is either already existing or emerging in ongoing grid and web services developments. An overview of past computer integrated construction research is presented by Boddy et al. [8].

The industry still communicates mostly by using drawings, files, project web sites and related ASP services. Semantic Web services built around a standardised product model have been demonstrated partially in research projects (e.g. ISTforCE, OSMOS) but their scalability in large complex environments has not been tested. Semantic interoperability of software and information systems belonging to members of the virtual organisation is essential for their efficient collaboration. Grids provide the robustness but need to be made aware of the business concepts that the VO is addressing. The grid environment itself needs to commit to ontology of the products and processes, thereby evolving into an ontology committed semantic grid environment; to do so there is a need for the generic business-object-aware extensions to grid middleware, implemented in a way that allows grids to commit to an arbitrary ontology; these extensions need to be propagated to toolkits that allow hardware and software to be integrated into the grid. These were the challenges in the

InteliGrid project.

Key requirements for the InteliGrid platform were gathered through an extensive requirements elicitation and analysis process and were used as a baseline in the design of the high-level InteliGrid architecture [9]. Based on the work done by the OSMOS project, InteliGrid internal requirements analysis and feedback from various public demonstrations, as well as various formal and informal discussions with different members of the engineering community, the top requirements can be summarised with the term *5S Grid*:

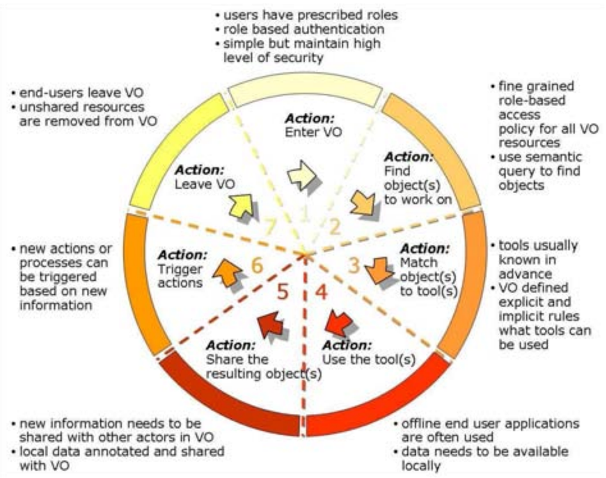

• users have prescribed roles
• role based authentication
• simple but maintain high level of security
• end-users leave VO
• unshared resources are removed from VO
Action: Enter VO
• fine grained role-based access policy for all VO resources
• use semantic query to find objects
Action: Leave VO
Action: Find object(s) to work on
1
2
7
• new actions or processes can be triggered based on new information
Action: Trigger actions
6
5
4
3
Action: Match object(s) to tool(s)
• tools usually known in advance
• VO defined explicit and implicit rules what tools can be used
Action: Share the resulting object(s)
Action: Use the tool(s)
• new information needs to be shared with other actors in VO
• local data annotated and shared with VO
• offline end user applications are often used
• data needs to be available locally

Figure 1: Generic virtual organisation end-user scenario actions.

- *Security.* Industry is eager to move to a ground-up secure environment [10]. InteliGrid addressed this by adopting Grid Security Infrastructure (GSI) and integrating the Role Based Access Control model (RBAC) [11] into the platform authorisation processes [12].

- *Simplicity.* The platform must work seamlessly with current client applications and operating systems and should not require end users to redefine their usual work processes.

- *Stability & standards.* The need for stable long-term specifications and (open) standards is well known [13]. The developed platform complies with such open standards, including WS-Resource Framework (WSRF) [14] and WS+Interoperability (WS+I) [15] for grid technology related developments, RBAC model for VO security, etc.

- *Scalable service oriented architecture (SOA).* The service-oriented architecture [16] is a well-accepted and known system architecture. The InteliGrid project adopted the Open Grid Service Architecture OGSA [17] as a baseline, and developed the platform using an OGSA compliant grid middleware.

- *Semantics.* The platform must support rich, domain specific semantics [18]. The InteliGrid project addressed this issue by developing a set of domain specific ontologies [19].

A number of different use cases were considered while designing the InteliGrid platform, ranging from basic ones, such as joining a virtual organisation, to more advanced cases involving the use of semantic information [9]. The developed use cases were abstracted into the generic virtual organisation end user scenario presented in Fig. 1. Starting from available technologies, industry practices and trends, the project aim was to create knowledge, infrastructure and toolkits that allow a broad transition of the industry towards semantic, model based and ontology committed collaboration based on the grid technology (rather than the web, which is the infracturcture technology of the SWOP project [20]).

The rest of the paper is organised as follows: Section 2 presents the background information on three key technologies (grid technology, semantic interoperability and virtual organisations), Section 3 describes the high-level system architecture of the platform as well as the developed tools and services, and finally Section 4 presents our conclusions and outlines proposed future research and development work.

2 Technology

The InteliGrid project addressed the challenge by successfully combining and extending the state-of-the-art research and technologies in three key areas: (a) semantic interoperability, (b) virtual organisations, and (c) grid technology (see Fig. 2) to provide standards-based collection of ontology based services and grid middleware in support of dynamic virtual organisations as well as grid enabled engineering applications. It was recognized that if a grid technology is to ensure the underlying engineering interoperability and collaboration infrastructure for a complex engineering virtual organisation, the grid technology needs to support shared semantics.

2.1 Grid technology

At its core, grid technology can be viewed as a generic enabling technology for distributed computing, based on an open set of standards and protocols that enable communication across heterogeneous, geographically dispersed environments. With grid computing, organizations can optimize computing and data resources, pool them for large capacity workloads, share them across networks and enable collaboration [21].

Foster [22] notes that the grid must be evaluated in terms of the applications, business value, and scientific results that it delivers, and not its architecture. It is based on hardware and software infrastructures which provide a dependable, consistent, pervasive and inexpensive access to com-

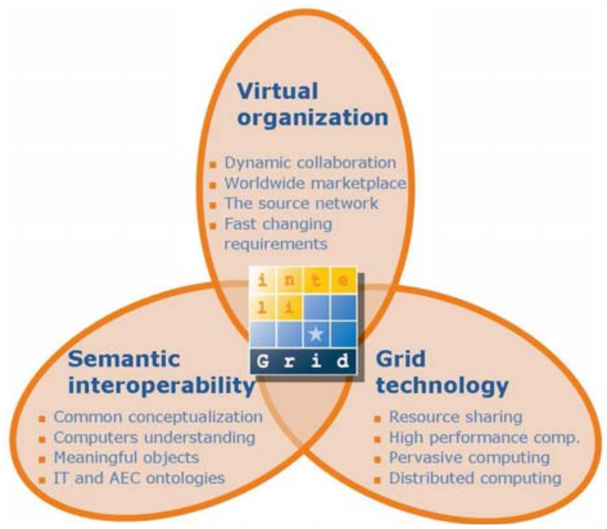


Figure 2: The InteliGrid project addressed three key technology areas: grid technology, semantic interoperability and virtual organisations.

puting resources anywhere and anytime. The term *resource* has evolved from covering only computing power and storage to covering a wide spectrum of concepts, including: physical resources (computation, communication, storage), informational resources (databases, archives, instruments), individuals (people and the expertise they represent), capabilities (software packages, brokering and scheduling services) and frameworks for access and control of these resources [23]. By using a grid for sharing resources, researchers and small enterprises can gain access to resources they cannot afford otherwise. Research institutes, on the other hand, can leverage their investment in research facilities by making them available to many more scientists. An overview of grid technology in civil engineering (including different grid technology standards, middleware and specific challenges related to technology addoption within engineering industries) has been published by Dolenc et al. [24].

2.2 Semantic interoperability

The development of grids and the increased use of agent and service based technologies have a profound impact on the way of data being exchanged on the Internet. The important feature of new semantic based approaches is the separation of content from presentation, which makes the use and reuse of data easier.

To achieve that kind of (semantic) interoperability, systems must be able to exchange data and information in a way that the precise meaning of the data is not lost and is readily accessible, and that the data itself can be translated into a form that is understandable by almost any system. It is important that the meaning of the exchanged information is interpreted accurately. The benefits of semantic interoperability are numerous, but the most notable one is that it assures the processing and reasoning of data by computers. The state-of-the-art developments and corresponding standards that are developed and used for semantic web applications can be reused in the grid computing environments with some modifications.

Semantic interoperability and its content description standards are in particular about ontologies and their inherent rules. Whilst the content description standard addresses the general applicability in distributed environments (as in InteliGrid), an important aspect in ensuring semantic interoperability is extensibility, i.e. the content description standard is required to make an 'open world assumption'. That is, semantic concepts are not confined to a single file or scope. While a concept may be defined originally in a basic ontology, it can be extended and instantiated in another definition or exchange file. The ontologies for semantic interoperability are therefore designed mostly in a layered approach, allowing for vertical and horizontal extensions. This means that the ontology has to support abstraction layers (from high-level concepts, such as 'resource', to specific concepts, such as 'construction-site-meeting-memo'), as well as the possibility for horizontal extensions targeting different domain/application areas. These requirements are especially addressed by ontology standards for the semantic web. Therefore they have been gaining more importance.

Semantic interoperability issues in the context of information and communication technology as well as recent semantic web developments have been addressed by Veltman [25].

2.3 Virtual organisations

Client demands for one-of-a-kind-products and services demand a one-time collaboration of different organisations, which have to consolidate and synergise their dispersed competencies in order to deliver the desired product or service. Each organisation is usually involved in the delivery of one or more components of the requested product or service. To deliver the complete product or service, organisations need to rely on each other for information completeness, as all product components are inter-related. Consequently, this has an implication not only on the way information (related to the to-be-delivered product or service) is exchanged and shared, but also on the way in which secure, quick to set-up, transparent (to the end-user) and non-intrusive (to the normal ways of work of an individual/organisation) information and communication technology is used for this purpose.

Virtual organisation is quickly becoming the preferred organisational form for one-of-a-kind settings to deliver one-of-a-kind product and typically goes through four distinct lifecycle stages (Fig. 3) [26]:

1. *Identification/conception* typically begins upon a specific (unique) client need for a product or service that a single organisation cannot deliver and serves as a business opportunity for a set of organisations which will

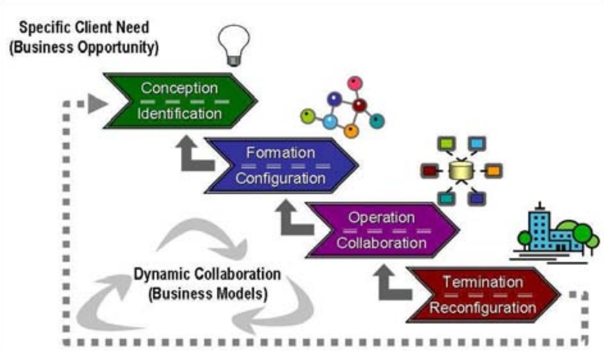


Figure 3: Typical virtual organisation lifecycle [26].

combine competencies to deliver the product and/or service that the client needs.

2. *Formation/configuration* focuses on the establishment of the VO in terms of role definitions, definition of information flow mechanisms, identification of information exchange formats and modalities, interoperability of inter- organisational tools, shared resource and services definition and configuration, etc. According to Kazi and Hannus [27] one of the key ICT requirements in a VO environment is the capability of a quick set-up and configuration.

3. *Operation/collaboration* is the main stage of a typical VO where different VO tasks are carried out in parallel and/or in series based on task needs. Within this stage there is a significant degree of work taking place within a distributed (engineering) setting with the possibility of some partners leaving and others joining according to the need of the VO.

4. *Termination/reconfiguration.* When a VO consortium completes the delivery of the required product/service, it is terminated or reconfigured to form another VO (e.g. from a VO that develops a product to a VO that provides maintenance or service for that product). During this stage, it is very important to have proper mechanism in place for archiving the data/information used and produced during the operation and collaboration stage.

3 InteliGrid platform

Grids were expected to be the solution to the "islands of computation" problem, but they were also expected to ensure the interoperability and collaboration platform providing that they include the key ingredient required for a complex engineering virtual organization - the support for the shared semantics. Scientific research and technical development in the project have advanced the state-of-the-art in the field of semantic grids and in the field of virtual organization interoperability; while the architecture, engineering and construction sector (including facility management) has provided the testing environment for the project, all technologies developed are generic and applicable in any kind of virtual organisation environment.

A grid environment, in the context of the InteliGrid project, is an infrastructure for secure and coordinated resource-sharing among individuals and institutions with the aim to create dynamic virtual organizations. InteliGrid's hypothesis was that the meaning of the resources should be explicit which leads us to the issue of semantics and ontologies. The vision of the project was to create virtual dynamic organizations through secure and coordinated resource-sharing among individuals, institutions, and resources. Grid computing is an approach to distributed computing that spans not only locations but also organizations, machine architectures and software boundaries to provide unlimited power, collaboration and information access to everyone connected to a grid.

3.1 Architecture

The InteliGrid architecture is based on the SOA concept as well as on lessons learned in earlier related projects [5, 6]. It is a high-level architecture, conforming to the key requirement of a generic approach which can be proven by trying to fit the existing architectures of systems developed over the last decade into it. The architecture is used also to identify the components that exist and the components that need to be developed. It includes (see Fig. 4): (1) the layer representing the conceptually modelled real world domain that is being addressed (e.g. buildings, aeroplanes, organizations, engineers, processes etc.), (2) the conceptual layer containing things that exist in the form of standards, ideas, graphs, schemas, ontologies, notions etc., (3) the software layer comprised of software that can be compiled, installed, executed, and runs and communicates with other software, and (4) the basic resource layer that include IT resources which are needed to run the applications and services defined in the layer above, e.g. hardware, firmware, software, etc.

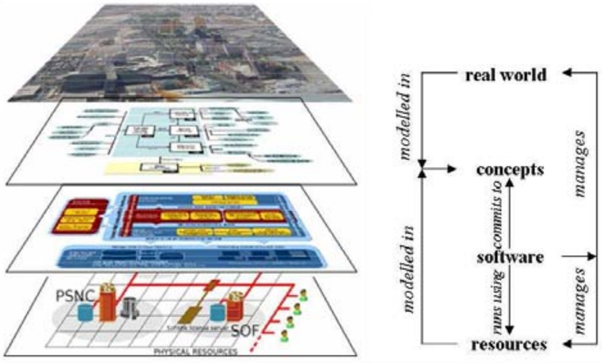


Figure 4: Four main layers of the InteliGrid conceptual architecture and their relationships - it is important that all architectural layers commit to common ontologies.

InteliGrid is delivering a generic grid-based integra-

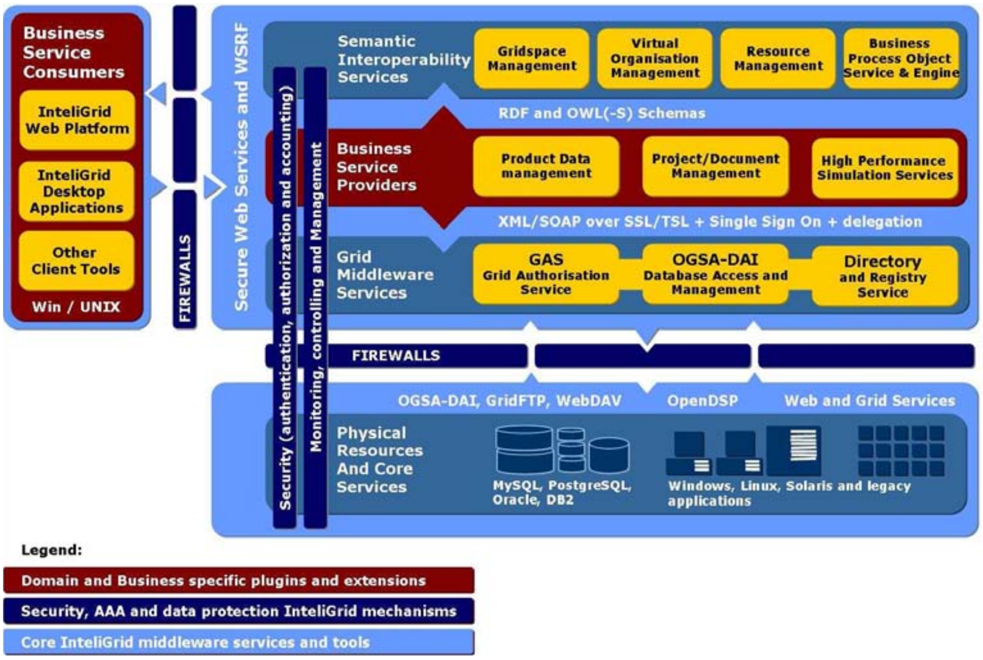

Business Service Consumers
InteliGrid Web Platform
InteliGrid Desktop Applications
Other Client Tools
Win / UNIX
FIREWALLS
Secure Web Services and WSRF
Security (authentication, authorization and accounting)
Monitoring, controlling and Management
Semantic Interoperability Services
Gridspace Management
Virtual Organisation Management
Resource Management
Business Process Object Service & Engine
RDF and OWL(-S) Schemas
Business Service Providers
Product Data management
Project/Document Management
High Performance Simulation Services
XML/SOAP over SSL/TSL + Single Sign On + delegation
Grid Middleware Services
GAS Grid Authorisation Service
OGSA-DAI Database Access and Management
Directory and Registry Service
FIREWALLS
OGSA-DAI, GridFTP, WebDAV
OpenDSP
Web and Grid Services
Physical Resources And Core Services
MySQL, PostgreSQL, Oracle, DB2
Windows, Linux, Solaris and legacy applications
Legend:
Domain and Business specific plugins and extensions
Security, AAA and data protection InteliGrid mechanisms
Core InteliGrid middleware services and tools

Figure 5: InteliGrid high-level platform architecture.

tion and a semantic-web based interoperability platform for creating and managing networked virtual organisations. The developed service-oriented architecture is presented in Fig. 5, together with all its principal components and their interfaces. From the security perspective, a virtual organisation is a collection of individuals and institutions, represented by various services and service consumers that are defined according to a set of resource and data sharing security policies and rules. Those resource sharing rules must be dynamically controlled and then enforced into the whole virtual organisation environment. Thus, one of the most challenging tasks in the project was to create an appropriate security infrastructure covering all aspects of operating within a dynamically established virtual organisation. The InteliGrid platform enables both service consumers and service providers to manage and share their resources securely with any of the individual organizations participating in the virtual organisation.

Technically speaking, components are deployed either at some workstation or at a remote node on the grid. If on the grid, it is not important where they are deployed physically, the resource where they run will be very likely allocated dynamically. The grouping of the various services in Fig. 5 is presented according to the logic of the service and does not necessarily imply who uses which service. There are four main types of components in the InteliGrid platform:

- *Business specific applications.* These applications are the consumers of the business service providers and are usually accessed through a web based portal interface, although desktop applications can also make use of different available services.

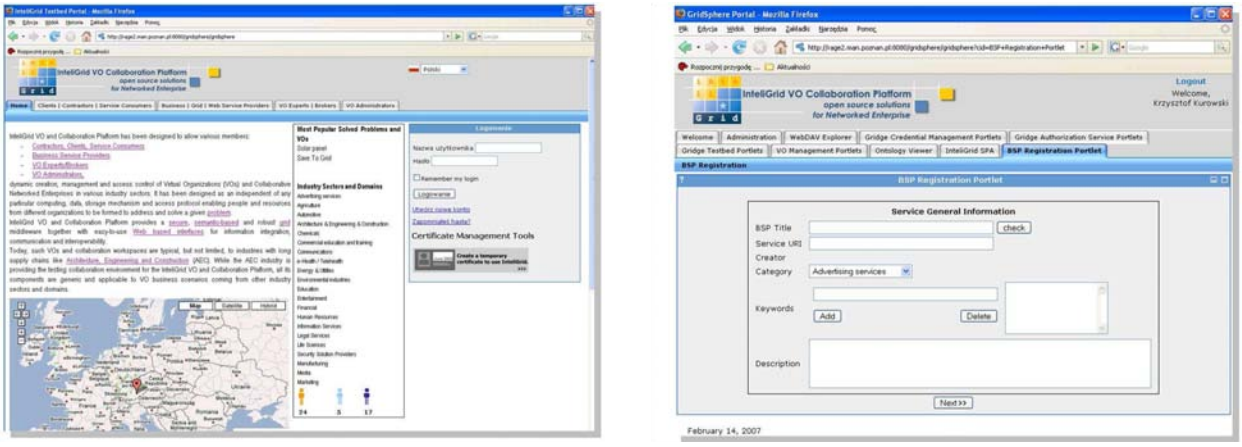
- *Secure Web Services and WSRF compliant services.* They can be further divided into: (1) inter-operability services (top tier) that simplify the interoperability among all services, and (2) domain and business specific services that perform some value added work. There are two kinds of business services: (a) collaboration services that provide file and structured data sharing and collaboration infrastructure, and (b) vertical business services that create new design or plan information.

- *Middleware services.* These services offer traditional grid middleware functionality extended with particular needs of the InteliGrid platform. The services are based on mature grid technologies and their open source reference implementations.

- *Other resources.* The bottom layer consists of various physical infrastructure resources that suppliers offered to the platform. All these resources are available and can be accessed remotely through well-defined interfaces and secure communication protocols.

3.2 Tools and services

The developed InteliGrid platform includes different client side applications and tools as well as many server side components enabling potential end users to securely execute high-performance calculations, access heterogeneous data resources, and generally work in established virtual organisations. The description of all available applications and services is available on-line at http://www.InteliGrid.com/products. The following sections provide an overview of the main InteliGrid products:



InteliGrid VO and Collaboration Platform has been designed to allow various members:

- Contractors, Clients, Service Consumers
- Business Service Providers
- VO Experts/Brokers
- VO Administrators,

dynamic creation, management and access control of Virtual Organizations (VOs) and Collaborative Networked Enterprises in various industry sectors. It has been designed as an independent of any particular computing, data, storage mechanism and access protocol enabling people and resources from different organizations to be formed to address and solve a given problem. InteliGrid VO and Collaboration Platform provides a secure, semantic-based and robust grid middleware together with easy-to-use Web based interfaces for information integration, communication and interoperability. Today, such VOs and collaboration workspaces are typical, but not limited, to industries with long supply chains like Architecture, Engineering and Construction (AEC). While the AEC industry is providing the testing collaboration environment for the InteliGrid VO and Collaboration Platform, all its components are generic and applicable to VO business scenarios coming from other industry sectors and domains.

Most Popular Solved Problems and VOs
Solar panel
Save To Grid

Industry Sectors and Domains
Advertising services
Agriculture
Automotive
Architecture & Engineering & Construction
Chemicals
Commercial education and training
Communications
e-Health / Telehealth
Energy & Utilities
Environmental industries
Education
Entertainment
Financial
Human Resources
Information Services
Legal Services
Life Sciences
Security Solution Providers
Manufacturing
Media
Marketing

Nazwa użytkownika
Hasło
□ Remember my login
Utwórz nowe konto
Zapomniałeś hasła?
Certificate Management Tools
Create a temporary certificate to use InteliGrid

(a) A single sign on entry point to all InteliGrid available online services - the authentication process is based on a defined RBAC model.

Service General Information
BSP Title
Service URI
Creator
Category: Advertising services
Keywords
Description

(b) The platform requires that all business services and resources are registered - a portlet enables registration of several different types of services and resources.

Figure 6: InteliGrid testbed portal implementation is based on the GridSphere portal framework.

- *Collaboration platform* that provides a working testbed environment, including online access to available resources;
- *Ontology services* that together with the developed ontologies, establish the conceptual and architectural backbone of a semantic grid infrastructure;
- *Semantic document management service and tools* that provide a major testing application for the ontology services;
- *High-performace services* that provide easy integration of existing engineering software (for example finite element codes, etc.);
- *Product model services* that provide access and itegration of engineering product models.

3.2.1 Collaboration platform

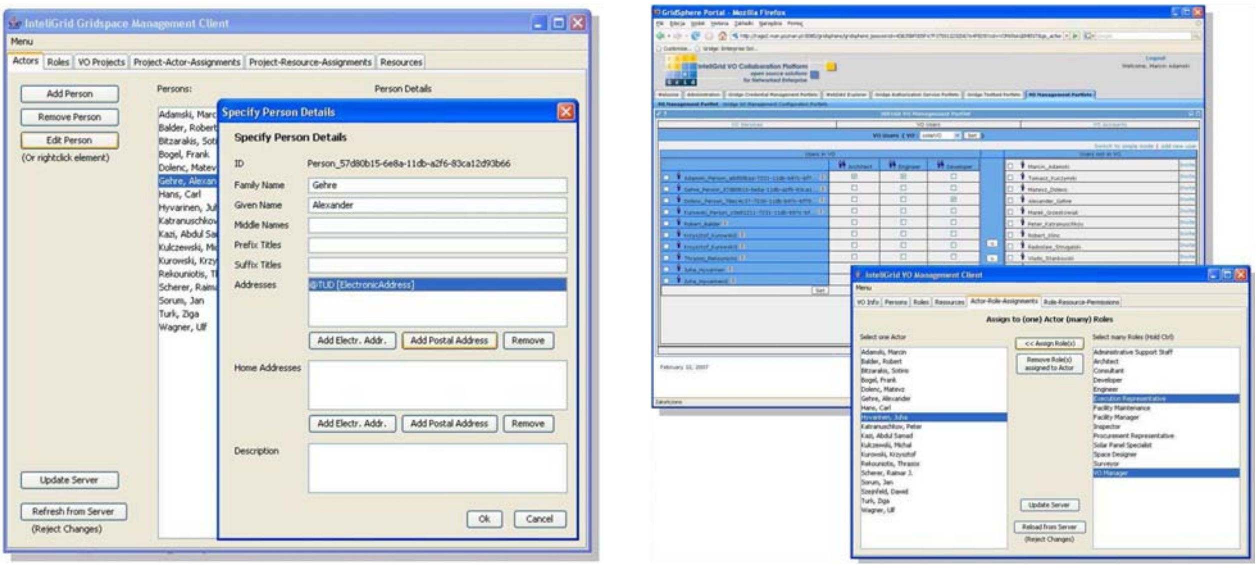
The InteliGrid collaboration platform for virtual organisations allows dynamic creation and management of virtual organizations in various engineering industry sectors. The platform is independed of the underlying computing technologies, data storage mechanisms or access protocols. The platform enables secure sharing and control of resources across dynamic and geographically dispersed organizations. It features a secure, semantic-based and robust grid middleware together with easy-to-use web based interfaces for information integration, communication and interoperability.

The web interface is built on the GridSphere portal framework [28] which provides an open source portlet-based web portal. Built-in single sign on (Fig. 6a), authentication, authorization and control mechanisms allow end users such as engineers, designers, architects, etc. to create their own space within a virtual organization to securely share relevant information and resources with other business partners and groups. The platform enables local administrators and IT staff to monitor the status and conditions of all provided services (Fig. 7a). It also allows virtual organisation managers to orchestrate and control access to different business service providers (Fig. 6b). Other actors such as virtual organisation project managers and grid administrators are able to establish and dynamically modify virtual organisations and their resources including users, services, databases and computation resources (Fig. 7b).

3.2.2 Ontology services

To fully utilise the advantages of the ontology-based approach, ontology services - providing convenient methods for management of ontology instances, i.e. semantic metadata about entities in the IT environment - need to be developed and made available through the platform service framework [29]. These services constitute the interoperability layer and make use of the grid middleware services that provide basic authorisation management and generic access to all grid resources. The ontology services provide generic and specific convenience methods to create, manipulate and manage the ontology instances of classes defined in the ontology framework.

The developed ontologies and ontology services establish the conceptual and architectural backbone of the semantic grid infrastructure. They facilitate information management, improve the consistency of the distributed environment and make it less prone to errors. End-user applications can also strongly benefit from the added semantic value. The technology is well suited to support human-computer interactions while semantic models are more related to end user perceptions than the usually applied IT based schemas. All InteliGrid developed business services and end-user client applications use ontology services actively to enhance the end user experience [30].



Persons:
Adamski, Marc
Balder, Robert
Bitzarakis, Soti
Bogel, Frank
Dolenc, Matev
Gehre, Alexan
Hans, Carl
Hyvarinen, Jul
Katranuschkov
Kazi, Abdul Sa
Kulczewski, Mi
Kurowski, Krzy
Rekouniotis, T
Scherer, Raim
Sorum, Jan
Turk, Ziga
Wagner, Ulf

Specify Person Details

ID: Person_57d80b15-6e8a-11db-a2f6-83ca12d93b66
Family Name: Gehre
Given Name: Alexander
Middle Names:
Prefix Titles:
Suffix Titles:
Addresses: @TUD [ElectronicAddress]
Home Addresses:
Description:

(a) A desktop application for platform management - service and resource monitoring, searching, etc.

Select one Actor:
Adamski, Marcin
Balder, Robert
Bitzarakis, Sotiris
Bogel, Frank
Dolenc, Matevz
Gehre, Alexander
Hans, Carl
Hyvarinen, Juha
Katranuschkov, Peter
Kazi, Abdul Samad
Kulczewski, Michal
Kurowski, Krzysztof
Rekouniotis, Thrasos
Scherer, Raimar J.
Sorum, Jan
Szeinfeld, Dawid
Turk, Ziga
Wagner, Ulf

Select many Roles (Hold Ctrl):
Administrative Support Staff
Architect
Consultant
Developer
Engineer
Execution Representative
Facility Maintenance
Facility Manager
Inspector
Procurement Representative
Solar Panel Specialist
Space Designer
Surveyor
VO Manager

(b) A portlet enabling dynamic user roles and resource access management - a single user can have different roles depending on the projects state.

Figure 7: The InteliGrid ontology based virtual organisation management clients.

3.2.3 Semantic document management service

As the majority of the communication in a typical engineering project is still document based it is essential for collaboration environments to provide tools and services that enable end users as well as other services to access document based information in a secure, location independent, personalised and on-demand way.

To address these requirements as well as the problem of information overload [31], a semantic document management system has been developed based on the InteliGrid semantic grid architecture. The design goal of the system was that the right document should be delivered to the right place at the right time to support the dynamic decision-making process at any level of a virtual organization. The document management system offers a generic, grid based ontology enabled document management solution that provides client (Fig. 8a) as well as server side components with a well-defined web services interface which enables a remote access to the underlying document management services. Some of the main features of the developed system are: security based on RBAC model, use of domain specific ontologies and taxonomies (Fig. 8b) for document annotations, semantic search based document retrival using SPARQL query language [32], etc.

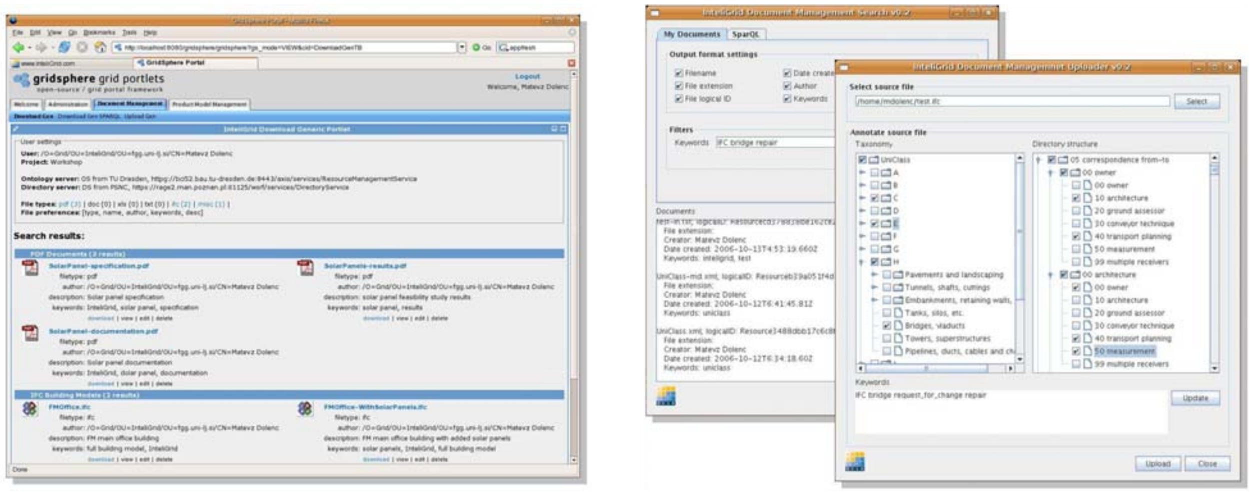
3.2.4 High-performace service

Engineering end users and application developers are interested in running complex computing experiments consisting of thousands of jobs or scientific services which have to be dynamically managed over distributed grid environments. The workflow is a widely accepted approach to compose grid experiments defining thousands of different parallel or sequential tasks together with various dependencies among them in advance. Consequently, people are interested in a high-level intuitive description of scientific workflows as well as a grid workflow management system providing a support for remote workflow execution and runtime control. If the complexity and security constraints of the distributed infrastructures on which workflow experiments have to be performed efficiently are realized, some additional mechanisms to the grid workflow management system are required, for instance secure data transfer mechanisms, data replica and management services, etc. All these components together are parts of InteliGrid High Performance Services. In addition to aforementioned functionalities the graphic tools, portals and GUIs are also desired to enable end users a visual workflow composition and animation (Fig. 9).

InteliGrid High Performance Service is in fact a resource management system [33] with a workflow engine that executes and manages jobs on remote grid resources. It is possible to submit to this service workflow experiments based on an XML workflow schema, defining flexible and mechanisms for dynamic workflow control, including various types of precedence constraints, different locations of the final data products, executables, etc. All these features allow end users to speed up remote workflow calculations and improve data management mechanisms.

3.2.5 Product model services

The product data model is a shared resource for data interoperability between heterogeneous software applications in the manufacturing virtual enterprise [34]. Multiple client applications need simultaneous and consistent access to



gridsphere grid portlets
open-source / grid portal framework

Welcome, Matevz Dolenc

InteliGrid Download Generic Portlet

User settings

User: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
Project: Workshop

Ontology server: OS from TU Dresden, httpg://bci52.bau.tu-dresden.de:8443/axis/services/ResourceManagementService
Directory server: DS from PSNC, https://rage2.man.poznan.pl:81125/wsrf/services/DirectoryService

File types: pdf (3) | doc (0) | xls (0) | txt (0) | ifc (2) | misc (1) |
File preferences: [type, name, author, keywords, desc]

Search results:

PDF Documents (3 results)

SolarPanel-specification.pdf
filetype: pdf
author: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
description: Solar panel specification
keywords: InteliGrid, solar panel, specification
download | view | edit | delete

SolarPanels-results.pdf
filetype: pdf
author: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
description: solar panel feasibility study results
keywords: solar panel, results
download | view | edit | delete

SolarPanel-documentation.pdf
filetype: pdf
author: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
description: Solar panel documentation
keywords: InteliGrid, dolar panel, documentation
download | view | edit | delete

IFC Building Models (2 results)

FMOffice.ifc
filetype: ifc
author: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
description: FM main office building
keywords: full building model, InteliGrid
download | view | edit | delete

FMOffice-WithSolarPanels.ifc
filetype: ifc
author: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
description: FM main office building with added solar panels
keywords: solar panels, InteliGrid, full building model
download | view | edit | delete

(a) Users can browse or search (using SPARQL query language) for documents independet of their location.

InteliGrid Document Management Search v0.2

My Documents | SparQL

Output format settings
☑ Filename ☑ Date create
☑ File extension ☑ Author
☑ File logical ID ☑ Keywords

Filters
Keywords: IFC bridge repair

Documents
test-in.txt; logicalID: Resourcecd3788...
File extension:
Creator: Matevz Dolenc
Date created: 2006-10-13T4:53:19.660Z
Keywords: inteligrid, test

UniClass-md.xml; logicalID: Resourceb39a051f4d...
File extension:
Creator: Matevz Dolenc
Date created: 2006-10-12T6:41:45.81Z
Keywords: uniclass

UniClass.xml; logicalID: Resource3488dbb17c6c8...
File extension:
Creator: Matevz Dolenc
Date created: 2006-10-12T6:34:18.60Z
Keywords: uniclass

InteliGrid Document Managemnet Uploader v0.2

Select source file
/home/mdolenc/test.ifc [Select]

Annotate source file

Taxonomy
☑ UniClass
☐ A
☐ B
☑ C
☐ D
☑ E
☐ F
☐ G
☑ H
☐ Pavements and landscaping
☐ Tunnels, shafts, cuttings
☐ Embankments, retaining walls,
☐ Tanks, silos, etc.
☑ Bridges, viaducts
☐ Towers, superstructures
☐ Pipelines, ducts, cables and ch

Directory structure
☑ 05 correspondence from-to
☑ 00 owner
☐ 00 owner
☑ 10 architecture
☐ 20 ground assessor
☐ 30 conveyor technique
☑ 40 transport planning
☐ 50 measurement
☐ 99 multiple receivers
☑ 00 architecture
☑ 00 owner
☐ 10 architecture
☐ 20 ground assessor
☐ 30 conveyor technique
☑ 40 transport planning
☑ 50 measurement
☐ 99 multiple receivers

Keywords
IFC bridge request_for_change repair [Update]

[Upload] [Close]

(b) Documents can be annotated using domain specific ontologies and taxonomies as well as free text.

Figure 8: InteliGrid semantic based document management system.

partial model data and must be able to immediately, incrementally and transparently update the persistent storage. The challenge of the grid enabled product model servers (to be available anywhere, any time and without regard to where the model data actually resides) is that the grid environment must be designed to make a single source of a product data model available for collaborative work sessions and sharing between many different applications, spread over multiple coupled organizations in virtual enterprises. Besides, it must be able to create a single, virtual visualization of a collection of product model data sources, federated, for sharing by a single enterprise in a workgroup, multi-project environment.

The InteliGrid Product Data Management Service, as a part of the InteliGrid Platform, addresses the above stated challenges by providing interfaces and different services for product model access and integration (Fig. 10):

- *Product model server* provides a back-end product model storage and management including various programming possibilities enabling development of additional server side operations or stored procedures.

- *Product data management service* is the main server side component of the product data management system and provides an easy to use programming interface for the development of client and server side applications that exploit product model server resources.

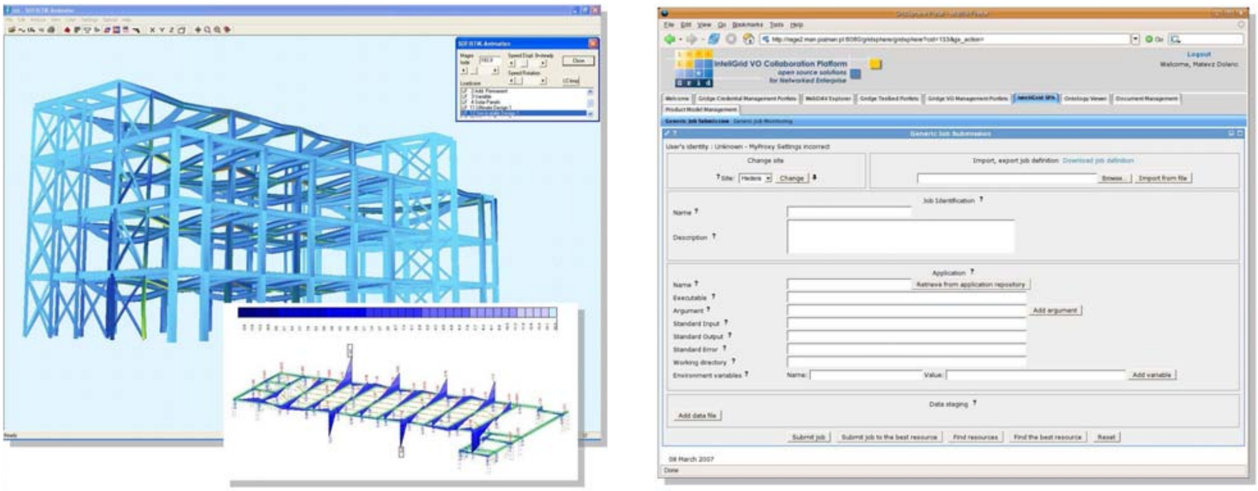
4 Conclusions and future work

The paper describes the overview of a collaboration platform developed within the InteliGrid project which combines and extends the state-of-the-art in technology areas of grid technology, semantic interoperability, and virtual organisations. The underlying system architecture is largely technology independent and can support different engineering domains. Although the main integrated demonstrator was from the architecture, engineering and construction sector, it has been shown in a number of partial demonstrations that the developed platform can be adapted for other engineering sectors as well. The demonstrations presented several advantages over other collaboration platforms for the support of virtual organisations. Nevertheless it is recognised that several questions about the effective use of the technologies remain unanswered:

- *Grid services (technology)* standards are converging with already widely accepted web service standards and container implementations. It is therefore relevant to question whether grid technology was the right choice for the underlying communication technology or not. We argue that it was as it does provide one of the core added values; namely secure communication environment required by the industry.

- *Dynamic virtual organisations* providing one-of-a-kind products require dynamic configuration of the information and communication infrastructure. This can be achieved by adopting certain design principles and the use of advanced information technologies. But there are many legal issues that need to be addressed before dynamic virtual organisations can became primary organisational form in the global economy, e.g. data management after the project is finished, the use of different national regulations, etc.

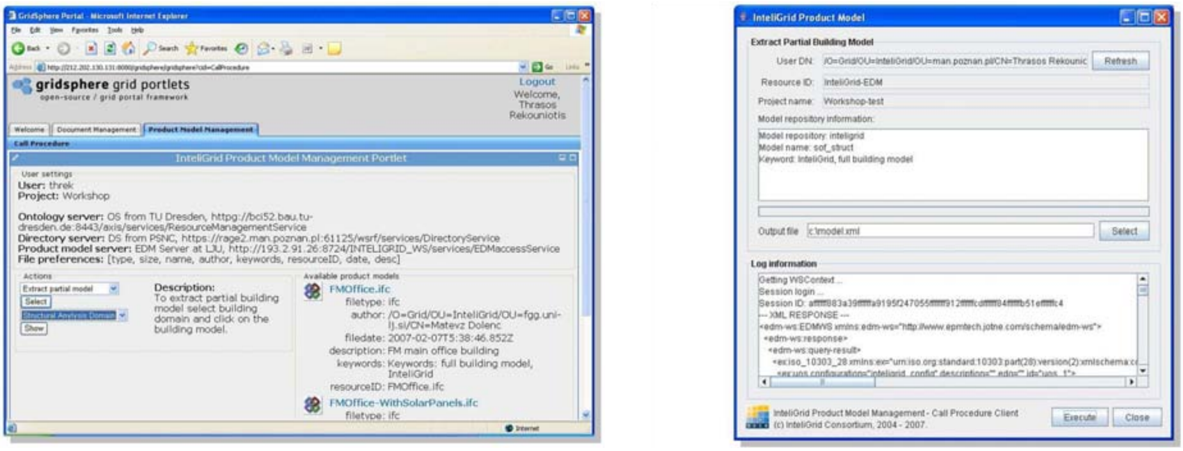
- The InteliGrid project focused its efforts on providing virtual organisation related communication and *semantic interoperability* regarding services and resources - it was assumed that the data level interoperability is provided. The data level interoperability is of course the baseline for the general integration within



(a) The InteliGrid platform integrates various client side structural analysis software - client side libraries are available to ease the development of integrated applications.

(b) A portlet based application for online job submittion - any commandline client side application can be used as a part of an analysis workflow.

Figure 9: InteliGrid high-performance services enable integration of existing engineering applications.



User settings
User: threk
Project: Workshop

Ontology server: OS from TU Dresden, httpg://bci52.bau.tu-dresden.de:8443/axis/services/ResourceManagementService
Directory server: DS from PSNC, https://rage2.man.poznan.pl:61125/wsrf/services/DirectoryService
Product model server: EDM Server at LJU, http://193.2.91.26:8724/INTELIGRID_WS/services/EDMaccessService
File preferences: [type, size, name, author, keywords, resourceID, date, desc]

Description: To extract partial building model select building domain and click on the building model.

FMOffice.ifc
filetype: ifc
author: /O=Grid/OU=InteliGrid/OU=fgg.uni-lj.si/CN=Matevz Dolenc
filedate: 2007-02-07T5:38:46.852Z
description: FM main office building
keywords: Keywords: full building model, InteliGrid
resourceID: FMOffice.ifc

FMOffice-WithSolarPanels.ifc
filetype: ifc

Extract Partial Building Model

User DN: /O=Grid/OU=InteliGrid/OU=man.poznan.pl/CN=Thrasos Rekounic
Resource ID: InteliGrid-EDM
Project name: Workshop-test
Model repository information:

Model repository: inteligrid
Model name: sof_struct
Keyword: InteliGrid, full building model

Output file: c:\model.xml

Log information

```
Getting WSContext ...
Session login ...
Session ID: a####883a39####a9195f247055####912####cd####84####b51e####c4
--- XML RESPONSE ---
<edm-ws:EDMWS xmlns:edm-ws="http://www.epmtech.jotne.com/schema/edm-ws">
 <edm-ws:response>
  <edm-ws:query-result>
   <ex:iso_10303_28 xmlns:ex="urn:iso.org:standard:10303:part(28):version(2):xmlschema:c
    <ex:uos configuration="inteligrid_config" description="" edo="" id="uos_1">
```

InteliGrid Product Model Management - Call Procedure Client
(c) InteliGrid Consortium, 2004 - 2007.

(a) A portled based application provides location independent access to various product models.

(b) An end-user application for extracting partial information models described by an Information Delivery Manual (IDM) description.

Figure 10: InteliGrid services enable access and integration of product information models.

specific industry sectors. Currently, there are several on-going initiatives addressing these issues [35].

– Relatively straight forward *business model* adopted for the InteliGrid platform requires that all available resources must be registered (including providing semantic annotation) in the platform. But it should be investigated whether a peer-to-peer system architecture supporting dynamic resource discovery and integration would be more appropriate for supporting dynamic virtual organisations providing one-of-a-kind products.

The future work following the project ending is focused on maintaining the established testbed as well as on addressing some of the above mentioned issues. One of the immediate research efforts is addressing issues of alternatives to the use of ontologies for semantic annotation of services and resources. In addition, it is expected that developed system architecture, concepts and platform will be tested in real world scenario.

Acknowledgement

The presented research has been carried out in the context of the 6th Framework IST project InteliGrid (IST4664), founded by the European Commission as well as industrial partners. The contribution of the founding agency and all project partners University of Ljubljana (Slovenia), Technical University of Dresden (Germany), VTT (Finland), ESoCE Net (Italy), Poznan Supercomputing and Networking Center (Poland), OBERMEYER Planen + Beraten (Germany), Sofistik (Greece), EPM Technology (Norway) and

Conject AG (Germany) is gratefully acknowledged.

References

[1] M. Hannus, H. Penttila, P. Silen, *"Islands of Automation"*, (1987 - updated by Hannus, M. 2002) http://cic.vtt.fi/hannus/islands.html

[2] R. Greening and M. Edwards, *"ATLAS implementation scenario"*. Scherer R. J. (ed.) Proc. ECPPM 1995 Product and Process Modelling in the Building Industry, Balkema, Rotterdam, 1995.

[3] R.J. Scherer, *"EU-project COMBI - Objectives and overview"*, Scherer R. J. (ed.) Proc. ECPPM 1995 Product and Process Modelling in the Building Industry, Balkema, Rotterdam, 1995.

[4] G. Augenbroe, *"An overview of the COMBINE project"*, Scherer R. J. (ed.) Proc. ECPPM 1995 Product and Process Modelling in the Building Industry, Balkema, Rotterdam, 1995.

[5] R.J. Scherer, R. Wasserfuhr, P. Katranuschkov, D. Hamann, R. Amor, M. Hannus, Z. Turk, *"A Concurrent Engineering IT Environment for the Building Construction Industry"*, In: D. Fichtner D. and R. MacKay (eds.) Facilitating Deployment of Information and Communication Technologies for Competitive Manufacturing, Proc. of the European Conf. on Integration in Manufacturing, IiM97, ISBN 3-86005-192-X, pp. 31-40, 1997.

[6] P. Katranuschkov, R.J. Scherer, Z. Turk, *"Intelligent services and tools for concurrent engineering: An approach towards the next generation of collaboration platforms"*, ITcon Vol. 6, Special Issue Information and Communication Technology Advances in the European Construction Industry, pp. 111-128, 2001, http://www.itcon.org/2001/9

[7] I. Wilson, S. Harvey, R. Vankeisbelck and A.S. Kazi, *"Enabling the construction virtual enterprise: the OSMOS approach"*, ITcon Vol. 6, Special Issue Information and Communication Technology Advances in the European Construction Industry, pp. 83-110, 2001, http://www.itcon.org/2001/8.

[8] S. Boddy, Y. Rezgui, G. Cooper and M. Wetherill, *"Computer integrated construction: A review and proposals for future direction"*, Advances in Engineering Software, Vol. 38, Issue 10, p. 677-687, October 2007.

[9] P. Katranuschkov, A. Gehre, E. Balaton, R. Balder, S. Bitzarakis, M. Dolenc, C. Ebert, C. Hans, J. Hyvarinen, K. Kurowski, T. Pappou, E. Petrinja, V. Stankovski, Z. Turk, U. Wagner, *"InteliGrid Deliverable D12 - Requirements Analysis, Rev. 1.4"*, The InteliGrid Consortium c/o University of Ljubljana, www.InteliGrid.com, 2006, http://www.inteligrid.com/data/works/att/d12.content.03122.pdf

[10] Deloitte, *"Global Security Survey"*, 2005, http://www.deloitte.com/dtt/cda/doc/content/dtt_financialservices_2005/GlobalSecuritySurvey_2005-07-21.pdf

[11] D.F. Ferraiolo, D.R. Kuhn, R. Chandramouli, *"Role Based Access Control"*, Artech House, 2003.

[12] M. Adamski, M. Kulczewski, K. Kurowski, J. Nabrzyski, A. Hume, *"Security and Performance Enhancements to OGSA-DAI for Grid Data Virtualization"*, 2nd VLDB Workshop on Data Management in Grids, Seoul, Korea, 11 September 2006.

[13] R.S. Sutor, *"Open Standards vs. Open Source: How to think about software, standards, and Service Oriented Architecture at the beginning of the 21st century"*, 2006, http://www.sutor.com/newsite/essays/Sutor-OpenStdsVsOpenSrc-20060527.pdf

[14] Web Services Resource Framework (WSRF), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

[15] WS-I Basic Profile 1.1 Specification, http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[16] T. Erl, *"Service-Oriented Architecture (SOA): Concepts, Technology, and Design"*, Prentice Hall PTR, 2005.

[17] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. von Reich, *"The Open Grid Services Architecture, Version 1.0"*, 2005, http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf

[18] Z. Turk, *"Understanding Grid semantics for virtual collaboration"*, 2005, http://istresults.cordis.lu/index.cfm/section/news/tpl/article/BrowsingType/Features/ID/79773

[19] A. Gehre and P. Katranuschkov, *"InteliGrid Deliverable D32.2 - Ontology Services"*, The InteliGrid Consortium c/o University of Ljubljana, www.inteliGrid.com, 2007.

[20] Semantic Web-based Open engineering Platform (SWOP), http://www.swop-project.eu

[21] J. Hyvarinen, Z. Turk, E. Balaton, C. Ebert, A. Gehre, P. Katranuschkov, A.S. Kazi, K. Kurowski,

E. Petrinja, V. Stankovski, *"D11.1 State of the art and market watch report"*, The InteliGrid Consortium c/o University of Ljubljana, www.InteliGrid.com, 2006, http://www.inteligrid.com/data/works/att/d11_1.content.05914.pdf

[22] I. Foster, *"What is the Grid: A Three Point Checklist"*, Grid Today, July 20. 2002.

[23] I. Foster, C. Kesselman, J.M. Nick & S. Tuecke, *"The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration"*, 2002, http://www.gridforum.org/ogsi-wg/drafts/ogsa_draft 2.9_2002-06-22.pdf

[24] M. Dolenc, V. Stankovski and Z. Turk, *"Grid Technology in Civil Engineering"*, Topping B.H.V. (ed.) Innovation in Civil and Structural Engineering Computing, Saxe-Coburg publications on computational engineering. Stirling: Saxe-Coburg, pp. 75-96, 2005.

[25] K.H. Veltman, *"Syntactic and Semantic Interoperability: New Approaches to Knowledge and the Semantic Web"*, The New Review of Information Networking, vol. 7, 2001,

[26] M. Hannus, *"Guidelines for Virtual Organisations"*, VOSTER Project Consortium, http://cic.vtt.fi/projects/voster, 2004.

[27] A.S. Kazi and M. Hannus, *"Interaction Mechanisms and Functional Needs for One-of-a-kind Production in Inter-enterprise Settings"*, Global Engineering and Manufacturing in Enterprise Networks (Karvonen I., et al., editors), VTT Symposium Series, pp.301-312, 2003.

[28] J. Novotny, M. Russell and O. Wehrens, *"GridSphere: A Portal Framework For Building Collaborations"*, http://www.gridlab.org/Resources/Papers/gridsphere_mgc_2003.pdf

[29] A. Gehre, P. Katranuschkov and R.J. Scherer, *"Managing Virtual Organization Processes by Semantic Web Ontologies"*, Rebolj D. (Ed.) Proc. CIB 24th W78 Conference Maribor Bringing ITC knowledge to work, University Library Maribor, ISBN 978-961-248-033-2, pp. 177-182, 2007.

[30] A. Gehre and P. Katranuschkov, *"InteliGrid Deliverable D32.2 - Ontology Services"*, The InteliGrid Consortium c/o University of Ljubljana, 2007, http://www.inteliGrid.com.

[31] K.A. Miller, *"Surviving Information Overload: The Clear, Practical Guide to Help You Stay on Top of What You Need to Know"*, Zondervan, 2004.

[32] E. Prud'hommeaux and A. Seaborne, *"SPARQL Query Language for RDF"*, W3C Working Draft, 26 March 2007, http://www.w3.org/TR/2007/WD-rdf-sparql-query-20070326/

[33] OpenDSP, http://sourceforge.net/projects/opendsp

[34] Z. Turk, *"Constraints of Product Modelling Approach in Building"*, 8th International Conference on Durability of Building, Vancouver, Canada, May 30 - June 3, 1999.

[35] buildingSMART, http://www.iai-na.org/bsmart/

A System for Speaker Detection and Tracking in Audio Broadcast News

Janez Žibert, Boštjan Vesnicer and France Mihelič
University of Ljubljana, Faculty of Electrical Engineering
Tržaška 25, SI-1000, Ljubljana, Slovenia
E-mail: janez.zibert@fe.uni-lj.si

Keywords: speaker diarization, speech detection, speaker clustering, audio indexing, speaker recognition, speaker tracking

Received: May 22, 2007

A system for speaker-based audio-indexing and an application for speaker-tracking in broadcast news audio are presented. The process of producing an indexing information in continuous audio streams based on detected speakers is composed of several tasks and is therefore treated as a multistage process. The main building blocks of such an indexing system include components for an audio segmentation, a speech detection, a speaker clustering and a speaker identification. We give an overview of each component of the system with emphasis to the approaches that are followed in each stage of building of our speaker-diarization and tracking system. The proposed system is evaluated on the audio data from the broadcast news domain, whereas we test each of the system's component and measure their impacts to the overall system's performance. The evaluation results indicate the importance of an audio segmentation and a speech detection module to the reliable performance of the whole system. Based on an indexing information produced by our system we also developed an application for searching target speakers in broadcast news. The application is designed in a way to be user-friendly and can be easily integrated in various computer environments.

Povzetek: Predstavljen je sistem za indeksacijo zvočnih posnetkov glede na govorce in aplikacija tega sistema za iskanje govorcev v zvočnih posnetkih informativnih oddaj.

1 Introduction

With the increasing availability of audio data derived from various multimedia sources comes an increasing need for efficient and effective means for searching and indexing through this type of information. Searching or tagging speech based on who is speaking is one of the more basic components required for dealing with spoken documents collected in large audio data archives, such as recordings of broadcast news or recorded meetings. In this paper, we focus on the indexing and searching of speakers in audio broadcast news (BN).

Audio data of BN shows present a typical multispeaker environment. The goal of searching and indexing of target speakers in such an environment is to find and identify the regions in the audio streams that belong to target speakers and produce an efficient way for accessing this regions from the audio data archives. The task of finding such speaker-defined regions is known as a speaker diarization task and was first introduced in the NIST[1] project of *Rich Transcription* in *'Who spoke when'* evaluations, [7]. The task of identifying the regions according to given speakers is known as a speaker tracking task and was first defined in 1999 NIST Speaker Recognition evaluation, [14]. While diarization and tracking procedures serve for a detection of speakers in audio data, is the purpose of speaker indexing an organization of audio data according to detected speakers for efficient speaker-based information audio-retrieval. In this paper, we present the approaches of speaker diarization and tracking in multispeaker audio BN data.

The paper is organized as follows. In the first sections, we describe in more detail a system for speaker diarization, which serves for speaker-indexing of BN shows. A system is composed of several components, which include procedures for an audio segmentation, a speech detection, a speaker clustering and a speaker identification. The first two procedures aim in detecting speaker and acoustic changes in speech portions of audio streams and thus correspond to partitioning of audio data to the homogeneous segments. The procedures for speaker clustering and identification are employed to group together segments of the same speaker and to provide speaker names to each such portion of speech data. Hence, they are used for tagging target speakers in the audio data. In Section 2, we give an overview of all of the above procedures, which were implemented to build a system for speaker tracking in BN shows. In the following section we present experiments and the evaluation results on the Slovenian audio BN database, where we explore the impact of each of the procedure on the overall speaker-tracking results. At the end, an application for speaker detection and tracking, based on the proposed methods, is described.

[1] National Institute of Standards and Technology, http://www.nist.gov/speech/

2 Speaker diarization in continuous audio streams

Speaker diarization is the process of partitioning input audio data into homogeneous segments according to the speaker's identities. The aim of speaker diarization is to improve the readability of an automatic transcription by structuring the audio stream into speaker turns, and in cases when used together with speaker-identification systems by providing the speaker's true identity. Such information is of interest to several speech- and audio-processing applications. For example, in automatic speech-recognition systems the information can be used for unsupervised speaker adaptation [1, 15], which can significantly improve the performance of speech recognition in large vocabulary continuous speech recognition (LVCSR) systems [10, 28, 4]. This information can also be applied for the indexing of multimedia documents, where homogeneous speaker or acoustic segments usually represent the basic units for indexing and searching in large archives of spoken audio documents, [13]. The outputs of a speaker diarization system could also be used in speaker-identification and in speaker-tracking systems, [6, 20], which was also the case in our presented application.

Most speaker diarization systems for a detection of speakers in continuous audio streams have a similar general architecture, [3, 26]. First, the signal is chopped into homogeneous segments. The segment boundaries are located by finding acoustic changes in the signal and each segment is expected to contain speech from only one speaker. Those segments, which do not represent speech data, are additionally detected and discarded from a further processing. The resulting segments are then clustered so that each cluster corresponds to only one speaker. At the final stage, each cluster is labeled by a corresponding speaker identification name or is left unlabeled, if the speech data in cluster do not correspond to any of the previously enrolled target speakers. As such, speaker diarization in continuous audio streams is a multistage process comprised by four main modules: an audio segmentation, a speech detection, a speaker clustering and a speaker identification.

A baseline speaker-indexing system architecture, which was followed in this work, is shown in Figure 1. First, the audio signal is processed in an *audio segmentation* module, where time-stamps are produced at the locations of detected acoustic changes. Audio data are thus partitioned into small homogeneous segments labeled by starting and ending time of each segment (segments: $[st_i, et_i]$ in Figure 1). It is expected that each such segment should contain data from just one acoustic source, i.e. speech from one speaker or non-speech data corresponding to music, silence or other non-speech source. Therefore, the obtained segments should be additionally divided to those, which contain speech or non-speech data. This is done in a *speech detection* module. Non-speech segments are marked as $[NS, st_i, et_i]$ in Figure 1 and are discarded from further processing. Only speech segments are then passed through a *speaker clustering* module. The aim of a speaker clustering is to merge speech segments from each speaker together, a major issue being that the information of speakers and the actual number of speakers are unknown *a priori* and need to be automatically determined. At this stage, just *relative* speaker labels are produced and segments are marked with automatically derived cluster names (segments $[C_i, st_i, et_i]$ in Figure 1). The true identities of the speakers are obtained in a *speaker identification* module in the next stage. Here, a multiple speaker verification of each cluster is performed. Speaker identification module is capable to recognize just those speakers, who are presented in the repository of the target speakers and are previously enrolled into the system. Speech data from clusters, which do not correspond to any of the speakers from target group, should be marked as *unknown* speaker data and are discarded from further processing.

Our speaker-indexing system [35] was designed in such a way, that all the modules include the standard approaches from similar state-of-the-art systems. In the following subsections each of the integrated module is described in more details.

2.1 Audio segmentation module

In general, spoken audio documents derived from BN shows include data from multiple audio sources, which may contain speech of different speakers as well as music segments, commercials and various types of noises, that are present in the background of BN reports. Another characteristic of BN audio documents is, that the data are delivered in the form of continuous audio streams. In order to efficiently process and extract the required information from such documents the continuously derived audio data should be adequately chopped into smaller portions of data, which are suitable for further processing. In the case of speaker-tracking applications the process of breaking the continuous audio streams into the homogeneous regions based on speaker turns is done in an audio segmentation module.

The segmentation of the audio data was made using the acoustic-change detection procedure based on the Bayesian Information Criterion (BIC), which was first proposed for the audio segmentation in [5] and improved by Tritschler and Gopinath in [27]. The applied procedure processed the audio data in a single pass while searching for change points within a window using a penalized likelihood ratio test (BIC score) of whether the data in the window is better modeled by a single probability distribution or two different distributions. If the estimated BIC score was under the given threshold (meaning that the data from the current window are better modeled by two probability distributions), a change point was detected and searching was restarted in the next window. In the opposite case, the analyzed window was extended and searching was redone. The threshold, which was implicitly included in the penalty term of the BIC score, has to be given in advance and was in our case estimated from the training data. The output

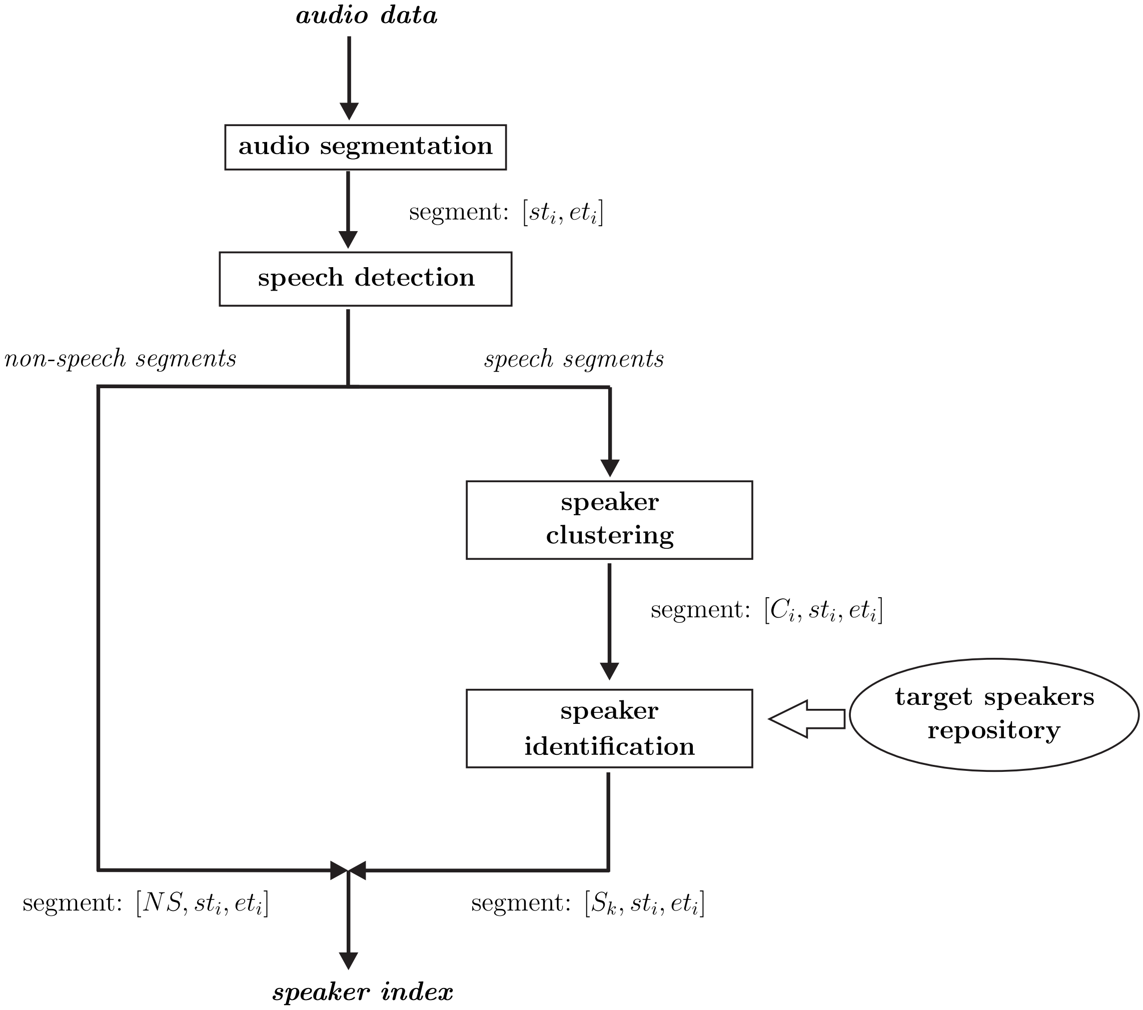


Figure 1: Main building blocks of a typical speaker-indexing system. Most systems have modules to perform speech detection, audio segmentation, speaker clustering and speaker identification, which may include component for gender detection.

of the audio segmentation module were acoustic change detection points, which defined basic audio segments for further processing.

This procedure is widely used in most of the current audio-segmentation systems [26, 7, 23, 30, 12, 33], and performed the best in comparison to alternative audio-segmentation approaches [26].

2.2 Speech detection module

The aim of this module in a speaker diarization system is to find the regions of speech in an audio stream. Since the audio stream was in our case already segmented into homogeneous regions of audio data based on acoustic changes, a speech detection module had to distinguish, which regions correspond to speech and non-speech data. The problem here represent non-speech data, which may consist of many acoustic phenomena such as silence, music, background noise or cross-talk.

The general approach used is a maximum likelihood classification with Gaussian Mixture Models (GMMs) estimated from acoustic representations of audio signals and trained on manually labeled training data [29, 19, 9, 23, 11, 24]. A speech detection based on such GMMs is performed either on pre-determined audio segments or by applying segmentation and detection together by using Viterbi decoding in the classification-network composed from trained GMMs. In both cases speech and non-speech data are usually modeled by several GMMs to cover various acoustic phenomena, which are expected in the processing audio data. To overcome this problem we proposed a new high-level representation of audio signals based on the phoneme recognition features, that are more suitable for speech/non-speech classification, [34, 16]. We developed four different measures based on consonant-vowel pairs and voiced-unvoiced regions obtained from phoneme speech recognizers and tested them in different segmentation-classification frameworks. The evaluation experiments on the BN audio data, [34], proved that a combination of acoustic features – modeled by mel-frequency cepstral coefficients (MFCCs) – and our proposed phoneme-recognition features constituted the most powerful representation of audio data, which were robust enough and relatively unsensitive to different training and unforseen conditions. Hence, we also implemented this kind of fusion of acoustic and phoneme-recognition representations in our speech detection module. The speech detection was performed by using a standard maximum likelihood classification with just two GMMs (one model for speech and the other for non-speech data) on already segmented audio streams, which were obtained from the previously described audio segmentation module.

Detected speech segments were further passed to a speaker clustering module, while non-speech segments were discarded from further processing.

2.3 Speaker clustering module

The purpose of this stage is to associate or cluster segments from the same speaker together. The ideal clustering should produce one cluster for each speaker, which should include all segments of a given speaker.

The general clustering method, which was also followed in our speaker-indexing system, is to perform agglomerative clustering using bottom-up approach, [25]. The basic steps of the speaker-clustering algorithm based on this approach can be described in the following steps [35]:

1. *Initialization step:* Model each segment by a single Gaussian distribution.

2. *Merging step:* Use a BIC measure to estimate whether to join two clusters or not. The candidates for merging are those clusters, where the lowest BIC score is achieved.

3. *Stopping step:* Repeat the second step until some stopping criterion is not satisfied.

Since in our speaker-clustering approach a BIC measure was used for merging, clusters should be modeled by Gaussian distributions. In the initialization step each segment represents one cluster. In the merging step joining of clusters (segments) is performed by searching a minimum (or maximum, depending on BIC measure) BIC score among all possible pair-wise combinations of clusters. A BIC measure is usually the same as one used for audio segmentation and also possesses the same philosophy. It measures the difference when modeling the data from two separate clusters with two normal distributions and when modeling with just one. The low differences speak in favor of modeling the data with just one distribution, meaning that the data the most likely belong to just one audio source, i.e. one speaker in our case, while higher differences support hypothesis that the data from separate clusters correspond to different speakers. The merging process is generally stopped when the lowest BIC score is greater than a specified threshold, but there can be also applied other stopping criteria, [35]. The stopping criterion is critical for a good performance and depends on how the output to be used [26]. In our speaker-tracking system a stopping threshold was used, which was estimated from the development data to optimize the speaker clustering performance.

The output of the speaker clustering module contains segments with relative labels, which join speech segments of the same speaker together. Non-speech segments are treated in this stage as separate cluster. The task of such labeling of continuous audio streams is known as a speaker diarization task and can be used in various audio processing applications.

In this stage, several improvements can be made to increase a speaker diarization performance, like joint segmentation and clustering [17] and/or cluster recombination [31], but in the case of indexing information by speakers in our speaker-tracking system we found no additional gain in the performance when applying some of these methods.

2.4 Speaker identification module

Since speaker diarization systems only produce relative speaker labels (such as 'spk1'), additional modules for speaker identification has to be included into the system, when the true identities of the speakers are needed. This can be achieved in various ways. We decided to follow the standard approach of building speaker models for people who are likely to be in the news broadcasts (such as prominent politicians or main news anchors and reporters) and including these models in the last stage of the speaker-indexing system.

A speaker identification component was adopted from a speaker verification system, which was based on the state-of-the-art Gaussian Mixture Model Ű Universal Background model (GMM-UBM) approach, [22]. Such systems are in generally composed of an enrolment phase and a test phase. In the enrollment phase, a model of the client (target) speaker is built based on a client's speech data, while in the test phase, another speech data, which are in our case collected from speaker clusters, are tested against a hypothesized client model. As a result, a matching score is generated based on the likelihood ratio (LR) between the likelihood that the speech was produced by the claimed speaker and the likelihood that the speech was not provided by the claimed speaker. If the score is greater than a given threshold, the speaker is accepted (client trial), otherwise it is rejected (impostor trial). There have been many solutions proposed how to efficiently calculate the denominator of the LR, i.e. the likelihood that the given speech data were not uttered by the claimed speaker. The best results up to now are achieved when likelihoods are calculated by using UBMs, which are usually trained from pooled speech of a large number of different speakers [22]. These models also serve as a prior for deriving client speaker models by Bayesian technique called maximum aposteriori (MAP) adaptation [8, 22], which was also applied in our speaker identification module.

In addition to that, we computed a new set of MFCC features, which were subjected to feature warping [21] to compensate different channel effects, and the log-likelihood scores normalization was performed at the end by applying ZT-normalization technique [2].

At the output of this stage the audio streams are equipped with the segment-time boundaries together with true speaker identification labels. Those clusters of segments, of which data do not belong to any of the enrolled speakers, get empty labels correspond to 'unknown' speakers. The output from this module also present the final results of the speaker-based audio-indexing and can be used for a detecting speakers in speaker-tracking applications. An application for speaker-tracking in BN shows, which was based on speaker's information obtained from our speaker-indexing system, is described in the last section.

3 Evaluation experiments

Evaluation of our speaker-based audio-indexing system was performed on the SiBN database [32], which consisted of 32 hours of BN shows in Slovenian language. 20 hours were used for an estimation of all the open parameters in all of the components of our indexing system, and the rest 12 hours served for the assessment of the system's performance.

The open parameters in the audio segmentation, the speech detection and the speaker clustering module corresponded to setting the thresholds to optimize the overall speaker diarization performance on the training data. In the audio segmentation module a threshold had to be estimated in the penalty factor of the BIC measure. It was set so to detect as many true change-detection points in the audio streams, while in the same time preserve low rate of miss-detected segment boundaries. The emphasis was put more on a detection of true segment boundaries, even if additional segment boundaries were falsely detected. In that case the over-segmented audio streams were produced, but they had almost no influence on the overall speaker diarization results while using them as inputs in speaker clustering module. In the case of under-segmented audio data it was found, that they could heavily degrade speaker-diarization and tracking performance. The same phenomena was explored in our speaker clustering module. Here, a threshold for stopping criteria of a merging process in a bottom-up clustering procedure had to be estimated. By setting a proper threshold we could optimize the speaker-diarization performance on the training data, but it was found out that this did not necessary reflect in the overall best performance of the speaker tracking system. The optimal performance was achieved in the case, when clusters did not contain speech from several speakers, i.e. a better performance was achieved in the under-clustering case, where speaker data were distributed over several clusters, rather than in the over-clustering case where too many contaminated clusters were produced containing speech from different speakers, which degraded a speaker-detection performance.

Another important issue was concerning a speech detection module. As was shown in [36] the impact of a speech detection in speaker diarization and tracking systems is direct and indirect. Since, non-speech data are treated as data from one of the speakers in the speaker-tracking system, a speech detection has a direct influence on the speaker-tracking results. On the other hand, an erroneous speech/non-speech classification of audio segments in the speaker-indexing system influences a speaker clustering and identification performance. Therefore, a good speech detection in continuous audio streams is a necessary pre-processing step for achieving a good speaker-diarization and tracking results. Since we decided to use a fusion of acoustic and phoneme-recognition features in a speech detection module, we had to apply a simplified version of a phoneme recognizer for deriving phoneme-recognition features. The recognizer was built on a standard way, using Hidden Markov models (HMMs) trained on Slovenian data. In addition, we had to estimate two GMMs for detecting speech and non-speech data, which were estimated from the training part of the SiBN database.

Since in a speaker identification module a true detection of speakers was carried out, GMM of each target speaker had to be provided. They were built from UBMs, which were trained on the speech data of the training part from the SiBN database. We were designed two UBMs corresponding to female and male speech data. All the models were constituted from 1024 Gaussian mixtures, which were estimated using Baum-Welch Expectation-Maximization (EM) algorithm. The GMM model for each target speaker was derived from corresponding UBM using MAP adaptation technique in a standard way, [22]. The evaluated system was capable to detect 41 target speakers, which were extracted from the training data in the enrollment phase. In the test phase, data from each cluster were compared against all of the models from target-speakers repository and LR score were produced. In the evaluation phase no additional score threshold was proposed, since we tried to evaluate the system in the whole range of all the possible operating points.

Note that gender-dependent UBMs were used for deriving speaker-dependent GMMs, meaning that in the test phase a gender classification was performed at first by using the same gender UBMs, which were also applied in the estimation of the target speaker models.

All modules in the tested system were built by using our own tools. The procedures for audio segmentation and speaker clustering were implemented in C/C++ programming environment, whereas the same component for the computation of the BIC measure was integrated in both modules. The fusion of acoustic and phoneme-recognition features was in the speech-detection module applied by performing a Viterbi decoding on a classification network of speech and non-speech GMMs. The training of GMMs and decoding through the network was done by using *HTK Toolkit*, [37], while the acoustic and phoneme-recognition features were produced by our own tools. The same set of acoustic features was then used in the speaker identification module, where all the training and testing procedures were also implemented by our own tools.

3.1 Evaluation results

Since several modules were included in the speaker-based audio-indexing system of BN shows, series of experiments were performed to measure the impact of each module to the overall speaker-tracking results.

Overall results of the evaluated speaker-tracking systems are depicted in Figure 2. The results are presented in terms of false acceptance (FA) and false rejection (FR) rates (false alarm and miss probabilities in Figure 2), measured at different operating points in the form of Detection Error Trade-off (DET) curves, [14]. In our case, the evaluated speaker tracking systems were capable to detect 41 target speakers from the audio data, which include 551 different speakers. The performances of the evaluated systems were therefore assessed by including all 41 target speakers with the addition of non-speech segments and the results were produced by FA and FR rates measured at the time (frame) level.

Figure 2 presents the evaluation results from six tested speaker-tracking systems, whereas different versions of system's components were combined. Only the speaker identification module was the same in all the evaluations, while other components (audio segmentation, speech detection and speaker clustering module) were combined by applying manual or automatic version of each procedure. In addition to that, two versions of speaker-tracking system without speaker clustering were also tested. In this way we tried to estimate the gain of each component to the overall speaker-tracking results. In Figure 2, a procedure for speech detection is marked as *S/N* (referring to speech/non-speech detection), procedures for audio segmentation are marked as *S* and for speaker clustering *C*. Manual versions of each procedure are abbreviated as *man*, automatic versions as *aut*, and in systems, where one of the procedure was missing, an abbreviation *w/o* is used for that procedure. For example, a system where manual audio segmentation was used prior to automatic speech/non-speech detection and automatic speaker clustering is in Figure 2 marked as *(aut S/N man S aut C)*, a system, where everything was performed automatically, is marked as *(aut S/N aut S aut C)*, etc.

The evaluation results in Figure 2 are displayed in terms of DET curves. They are ranging from the best performance of a system, where all procedures, except speaker identification, were preformed manually, to a system, where all the procedures were performed automatically.

The impact of speaker clustering were explored in series of experiments with systems *(man S/N man S man C)*, *(man S/N man S aut C)*, and *(man S/N man S w/o C)*, where audio segmentation and speech detection were performed manually, and with systems *(aut S/N aut S aut C)* and *(aut S/N aut S w/o C)*, where *S* and *S/N* procedures were performed automatically. Results from Figure 2 reveal expected performances of these systems. The best results were obtained in a system where everything was carried out manually, next to them are results from the system, where just speaker clustering procedure was applied automatically, and at the end are systems where all three procedures were done automatically. A comparison of the system *(man S/N man S man C)* and the system *(man S/N man S aut C)* indicates that a proper speaker clustering can significantlly improve the overall performance of a speaker-tracking system. The same can be concluded for speech detection and audio segmentation tasks by expecting the performances of the systems *(man S/N man S aut C)* and *(aut S/N aut S aut C)*. When applying automatic versions of audio segmentation and speech detection into the speaker-tracking system (by using the same speaker clustering procedure), the overall results of the system dropped for around 3% in the whole range of the operating points. Another important issue can be observed by expecting the systems *(man S/N man S aut C)* and *(man S/N man S w/o C)*, and the systems *(aut S/N aut S aut C)* and *(aut S/N aut S w/o C)*. In this evaluations we investigated whether it is better to use a speaker clustering procedure in speaker-tracking systems or not. As can be seen from Figure 2 there is no so much difference in the performances of the systems, where clustering was applied, to those without clustering. Our tracking results with automatic clusterings show that just a marginal gain could be obtained. This indicates that in our case the speaker-tracking system could not benefit from speaker clustering. The same was shown in the study of speaker tracking of radio broadcast news in [18], where it was concluded that a speaker identification can even help to improve a speaker clustering performance and not a vice versa.

The influence of an audio segmentation to the overall speaker-tracking results was explored by the evaluation of the systems *(aut S/N man S aut C)* and *(aut S/N aut S aut C)*. In the first case the audio segmentation was performed manually, while in the second the audio segmentation procedure, described in Section 2.1, was applied. As can be seen from the results in Figure 2 a manual segmentation outperforms an automatic version by approximately 3% in the whole range of operating points. This means that an audio segmentation plays an important role in our evaluated speaker-tracking system. Since segmentation procedure is usually applied in the first steps of speaker-tracking systems, the errors from segmentation have impact on all subsequent procedures. In our case, the errors in detecting change points in continuous audio streams produced non-homogeneous segments, which caused unreliable detection of speech/non-speech regions and unreliable detection of target speakers as well. Consequently, both types of er-

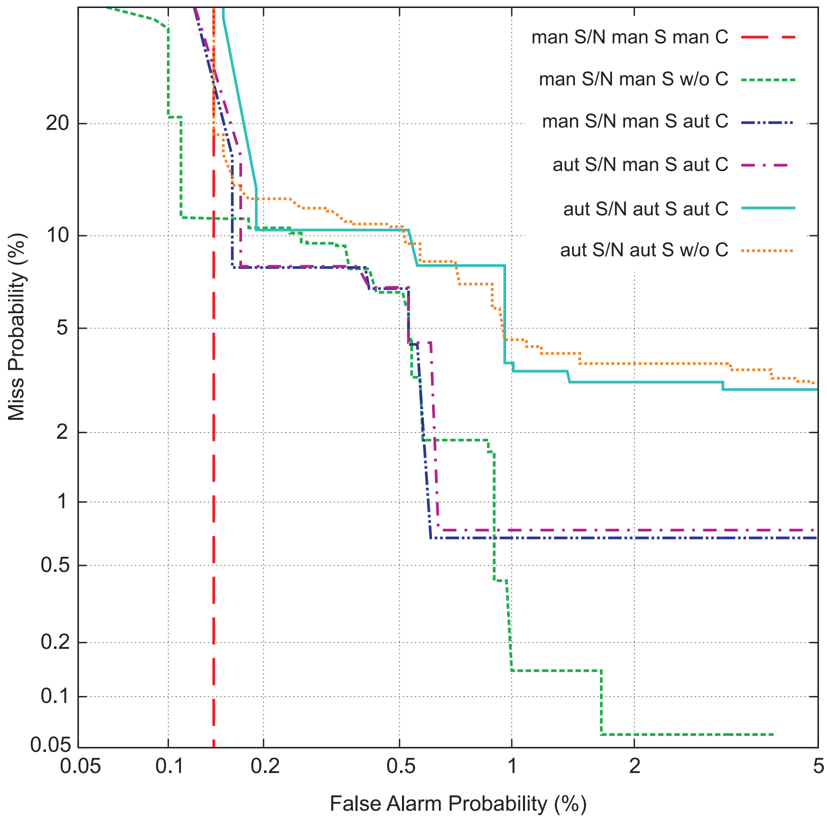
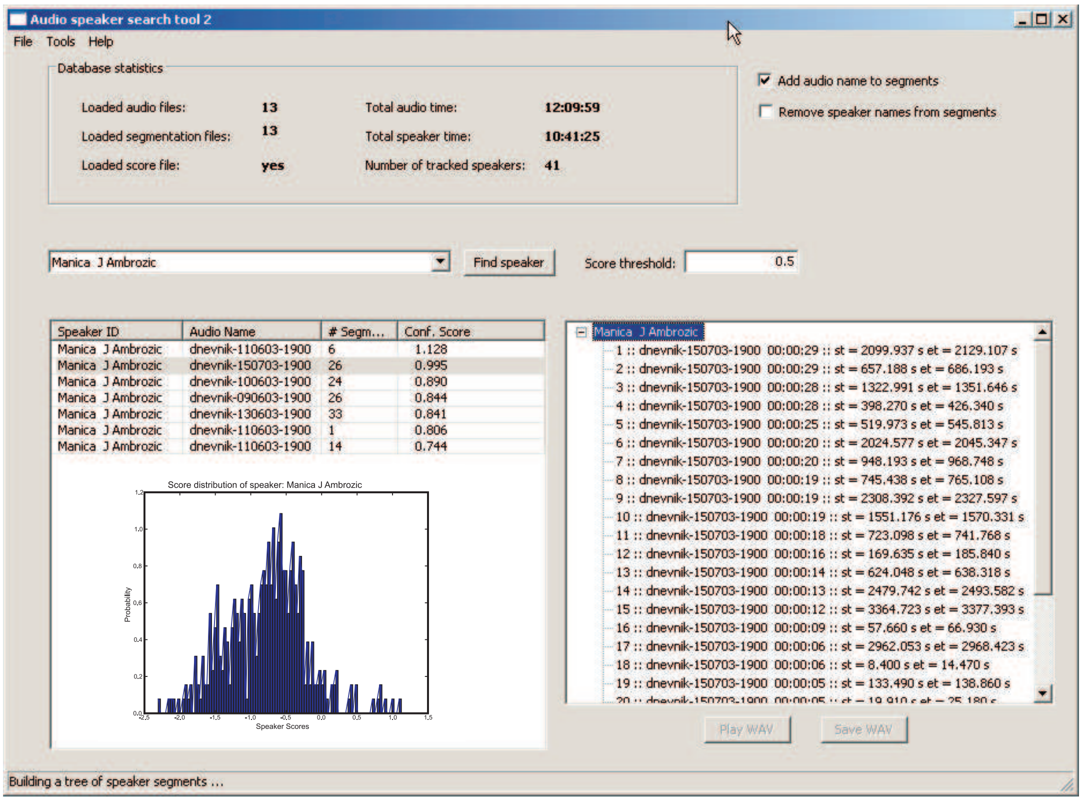


Figure 2: The overall speaker-tracking results of six evaluated systems. Lower DET values correspond to better performance.

rors were therefore integrated into the overall results of the evaluated system *(aut S/N aut S aut C)*.

Another evaluation perspective can be obtained by exploring systems *(man S/N man S aut C)* and *(aut S/N man S aut C)*. By comparing evaluation results of both systems we can estimate the gain of the speech detection procedure alone to the overall speaker-tracking results. As can be seen from the evaluation results in Figure 2, is the difference in the overall performances of systems, when using a manual and an automatic version of speech detection procedure, minimal. This marginal difference in the DET results was achieved due to the usage of the manual audio segmentation procedure in both systems. By applying a speech/non-speech detection procedure in a combination with manual segmentation (described in Section 2.2), a surprisingly high overall speech/non-speech accuracy of 99.38% was achieved, which resulted in the minimal difference of both evaluated systems. Note, that we used our own method for speech/non-speech detection, which proved to be a better choice for the speaker-diarisation and tracking tasks, as it was shown in a comparison study in [36].

To sum up, the comparison of the evaluation results of the different versions of the speaker-tracking system provides valuable insights of how the system works and which components of the system have greater impact on the overall performance. The overall results reveal an acceptable performance of the system, where all of the system's procedures were performed automatically. All other evaluated versions of the system serve for the estimation of the impact of each component to the overall speaker-tracking performance. It was found out that probably the most important component of the system is an audio segmentation module. If a segmentation procedure produces too many non-homogeneous segments due to improper detected change points in an audio stream, causes unreliable performances of a speech-detection and a speaker-identification module, and thus degrades the overall system's performance. As far as concerning speech detection module alone it was also shown, that we could gain some improvement in the overall system's performance by applying a good speech detection procedure. Since in our evaluated system a speech detection procedure, proposed in [34], was applied, almost no loss of the overall performance was obtained. Another important finding was concerning speaker clustering. The evaluation showed no important gain in the overall results, when speaker clustering was applied or not. This was in accordance with findings in [18].



Database statistics

Loaded audio files: 13
Loaded segmentation files: 13
Loaded score file: yes
Total audio time: 12:09:59
Total speaker time: 10:41:25
Number of tracked speakers: 41

☑ Add audio name to segments
☐ Remove speaker names from segments

Manica J Ambrozic

Score threshold: 0.5

Speaker ID	Audio Name	# Segm...	Conf. Score
Manica J Ambrozic	dnevnik-110603-1900	6	1.128
Manica J Ambrozic	dnevnik-150703-1900	26	0.995
Manica J Ambrozic	dnevnik-100603-1900	24	0.890
Manica J Ambrozic	dnevnik-090603-1900	26	0.844
Manica J Ambrozic	dnevnik-130603-1900	33	0.841
Manica J Ambrozic	dnevnik-110603-1900	1	0.806
Manica J Ambrozic	dnevnik-110603-1900	14	0.744

```
⊟ Manica J Ambrozic
   1 :: dnevnik-150703-1900  00:00:29 :: st = 2099.937 s et = 2129.107 s
   2 :: dnevnik-150703-1900  00:00:29 :: st = 657.188 s et = 686.193 s
   3 :: dnevnik-150703-1900  00:00:28 :: st = 1322.991 s et = 1351.646 s
   4 :: dnevnik-150703-1900  00:00:28 :: st = 398.270 s et = 426.340 s
   5 :: dnevnik-150703-1900  00:00:25 :: st = 519.973 s et = 545.813 s
   6 :: dnevnik-150703-1900  00:00:20 :: st = 2024.577 s et = 2045.347 s
   7 :: dnevnik-150703-1900  00:00:20 :: st = 948.193 s et = 968.748 s
   8 :: dnevnik-150703-1900  00:00:19 :: st = 745.438 s et = 765.108 s
   9 :: dnevnik-150703-1900  00:00:19 :: st = 2308.392 s et = 2327.597 s
   10 :: dnevnik-150703-1900  00:00:19 :: st = 1551.176 s et = 1570.331 s
   11 :: dnevnik-150703-1900  00:00:18 :: st = 723.098 s et = 741.768 s
   12 :: dnevnik-150703-1900  00:00:16 :: st = 169.635 s et = 185.840 s
   13 :: dnevnik-150703-1900  00:00:14 :: st = 624.048 s et = 638.318 s
   14 :: dnevnik-150703-1900  00:00:13 :: st = 2479.742 s et = 2493.582 s
   15 :: dnevnik-150703-1900  00:00:12 :: st = 3364.723 s et = 3377.393 s
   16 :: dnevnik-150703-1900  00:00:09 :: st = 57.660 s et = 66.930 s
   17 :: dnevnik-150703-1900  00:00:06 :: st = 2962.053 s et = 2968.423 s
   18 :: dnevnik-150703-1900  00:00:06 :: st = 8.400 s et = 14.470 s
   19 :: dnevnik-150703-1900  00:00:05 :: st = 133.490 s et = 138.860 s
   20 :: dnevnik-150703-1900  00:00:05 :: st = 19.910 s et = 25.180 s
```

Building a tree of speaker segments ...

Figure 3: A graphical user interface of a speaker-tracking system.

4 Speaker tracking system

A derivation of indexing information by speakers is an important step in applications, which are used for searching speakers in large archives of audio data. In this section we present one such application for a detection of speakers in continuous audio streams of BN shows, which are based on the system for a speaker-based audio-indexing presented in previous sections.

The application was designed in a way to separate processes of audio-indexing and searching of target speakers. This is also a standard approach in search engines based on text data. The indexing process is usually done once in a while, i.e. when new data arrive and index has to be updated, while searching of information is done all the time.

In our application the process of audio-indexing was performed once on the BN data from the SiBN database. The output of this process were time boundaries of speech segments with target-speaker's scores. And in the searching process the audio segments corresponding to a given speaker have to be provided and properly displayed to a user. A graphical user interface of our searching application is shown in Figure 3.

In the top pane of the application in Figure 3 are displayed some base properties of an audio database, which is currently loaded into the system, i.e. the information of total audio data time, of speech data time, how many speakers can the system detect, etc. In the middle pane is the search form, which includes a list of all possible target speakers, which the system is capable to find, and the score threshold, that can be optionally set to return just those speech segments, where speaker-detection scores are above the given threshold. The two bottom panes display information of the speaker, who has to be found by hitting the *Find-speaker* button. The bottom-left pane are filled with cluster information corresponding to a searched speaker, which are in the case of BN data divided to each BN show. The clusters are by default sorted by a confidence score, but the application also provides other sort possibilities, i.e. sorting by BN show name, number of segments in cluster, speaker name, etc. At the bottom of this pane it is also showed a histogram of the LR scores of a given speaker from all possible clusters in the database. A speaker score-distribution displayed in a histogram can serve for estimating the optimal threshold for obtaining just speech data of the current speaker . In this way a user can control the amount of data that are displayed and can inspect how likely the current data correspond to a searched speaker. The right-bottom pane in Figure 3 displays a list of all segments of a target speaker's cluster, which is marked in the left-bottom pane. A change in cluster (in the left-bottom pane) cause a fill of a new list with segments of that marked cluster. A user can listen or save the audio data by clicking on one of the displayed segments.

This application was developed by using Python programming language, while a graphical user interface of the application was designed by using *wxPython cross-platform GUI Toolkit*[2]. The application was implemented to operate as a stand-alone process and currently works just for searching speakers in BN shows, but it can be easily extended for other types of audio documents. Since the application expects that the audio-indexing is done beforehand, it is also independent of the methods used in the audio-indexing system. As such, it can be integrated in various types of computer applications and environments.

5 Conclusion

A system for speaker-based audio-indexing and an application for speaker-tracking in BN audio data based on this system were presented. We gave an overview of four main building blocks of such audio-indexing system and provide an extensive evaluation of all of the system's components. While they were modules for an audio segmentation, a speaker clustering and an identification implemented by using the latest state-of-the-art approaches, was in a module for speech detection followed our own approach of incorporating phoneme-recognition features in a classification process. In the evaluation experiments the impact of each module to the overall speaker-tracking performance was measured. It was found out that the most critical component of such a system is an audio segmentation module, since it is usually applied in the first processing stages of such system and its poor performance causes unreliable performances of all other components. Nevertheless, the evaluation results demonstrate an acceptable performance of the system, where all of the procedures were performed automatically. This system were later applied for an audio-indexing of BN shows in a speaker-tracking application. An application was designed to serve as a search tool for speakers, who are likely to be in the news broadcasts, but it could be easily extended for other types of audio documents.

Acknowledgement

This work was supported by the two Slovenian Research Agency (ARRS), development projects: L2-6277 (C) entitled “Broadcast news processing system based on speech technologies” and M2-0210 (C) entitled “AvID: Audiovisual speaker identification and emotion detection for secure communications.”

References

[1] T. Anastasakos, J. McDonough, R. Schwartz, J. Makhoul (1996) A Compact Model for Speaker-Adaptive Training, *Proceedings of International Conference on Spoken Language Processing (ICSLP1996)*, Philadelphia, PA, USA, 1996, pp. 1137-1140.

[2] R. Auckenthaler, M. Carey, & H. Lloyd-Thomas (2000) Score normalization for text-independent speaker verification system, *Digital Signal Processing*, Vol. 10, No. 1, January 2000, pp. 42-54.

[3] C. Barras, X. Zhu, S. Meignier, & J.-L. Gauvain (2006) Multistage Speaker Diarization of Broadcast News, *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, Vol. 14, No. 5, September 2006, pp. 1505-1512.

[4] P. Beyerlein, X. Aubert, R. Haeb-Umbach, M. Harris, D. Klakow, A. Wendemuth, S. Molau, H. Ney, M. Pitz & A. Sixtus (2002) Large vocabulary continuous speech recognition of Broadcast News Ű The Philips/RWTH approach, *Speech Communications*, Vol. 37, No. 1-2, May 2002, pp. 109-131.

[5] S. S. Chen & P. S. Gopalakrishnan (1998) Speaker, environment and channel change detection and clustering via the Bayesian information criterion, *Proceedings of the DARPA Speech Recognition Workshop*, Lansdowne, Virginia, USA, February 1998, pp. 127-132.

[6] P. Delacourt, J. Bonastre, C. Fredouille, T. Merlin & C. Wellekens (2000) A Speaker Tracking System Based on Speaker Turn Detection for NIST Evaluation, *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2000)*, Istanbul, Turkey, June, 2000.

[7] J. Fiscus, J. S. Garofolo, A. Le, A. F. Martin, D. S. Pallet, M. A. Przybocki & G. Sanders (2004) Results of the Fall 2004 STT and MDE Evaluation, *Proceeedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[8] J. L. Gauvain & C.-H. Lee (1994) Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Trans. Speech, and Audio Processing*, Vol. 2, No. 2, April 1994, pp. 291-298.

[9] J.-L. Gauvain, L. Lamel & G. Adda (1998) Partitioning and Transcription of Broadcast News Data, *Proceedings of the International Conference on Spoken Language Processing (ICSLP98)*, Sydney, Australia, December 1998, pp. 1335-1338.

[10] J. L. Gauvain, L. Lamel & G. Adda (2002) The LIMSI Broadcast News transcription system, *Speech Communications*, Vol. 37, No. 1-2, May 2002, pp. 89-108.

[11] T. Hain, S. E. Johnson, A. Tuerk, P. C. Woodland & S. J. Young (1998) Segment Generation and Clustering in the HTK Broadcast News Transcription System, *Proceeedings of the 1998 DARPA Broadcast News Transcription System*, Lansdowne, VA, USA, 1998, pp. 133-137.

[2] http://www.wxpython.org/

[12] D. Istrate, N. Scheffer, C. Fredouille & J.-F. Bonastre (2005) Broadcast News Speaker Tracking for ESTER 2005 Campaign, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 2445-2448.

[13] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz & A. Srivastava (2000) Speech and language technologies for audio indexing and retrieval, *Proceedings of the IEEE*, Vol. 88, No. 8, 2000, pp. 1338-1353.

[14] A. Martin, M. Przybocki, G. Doddington & D. Reynolds (2000) The NIST speaker recognition evaluation - overview, methodology, systems, results, perspectives, *Speech Communications*, Vol. 31, No. 2-3, June 2000, pp. 225-254.

[15] S. Matsoukas, R. Schwartz, H. Jin & L. Nguyen (1997) Practical Implementations of Speaker-Adaptive Training, *Proceedings of the 1997 DARPA Speech Recognition Workshop*, Chantilly VA, USA, February 1997, pp. 1137-1140.

[16] F. Mihelic & J. Zibert (2006) Robust speech detection based on phoneme recognition features, *Proceedings of Text, speech and dialogue (TSD 2006)*, Brno, Czech Republic, September 2006, pp. 455-462.

[17] S. Meignier, J.-F. Bonastre, C. Fredouille & T. Merlin (2000) Evolutive HMM for Multi-Speaker Tracking System, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, Istanbul, Turkey, June 2000.

[18] D. Moraru, M. Ben & G. Gravier (2005) Experiments on speaker tracking and segmentation in radio broadcast news, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005.

[19] P. Nguyen, L. Rigazio, Y. Moh & J. C. Junqua (2002) Rich Transcription 2002 Site Report. Panasonic Speech Technology Laboratory (PSTL), *Proceeedings of the 2002 Rich Transcription Workshop*, Vienna, VA, USA, May 2002.

[20] B.Nedic, G. Gravier, J. Kharroubi, G. Chollet, D. Petrovska, G. Durou, F. Bimbot, R. Blouet, M. Seck, J.-F. Bonastre, C. Fredouille, T. Merlin, I. Magrin-Chagnolleau, S. Pigeon, P. Verlinde, & J. Cernocky (1999) The Elisa'99 Speaker Recognition and Tracking Systems, *Proceedings of IEEE Workshop on Automatic Advanced Technologies*, 1999.

[21] J. Pelecanos & S. Sridharan (2001) Feature warping for robust speaker verification, *Proceedings of the Speaker Odyssey Workshop*, Crete, Greece, June 2001, pp. 213Ű218.

[22] D. A. Reynolds, T. F. Quatieri, & R. B. Dunn (2000) Speaker verification using adapted Gaussian mixture models, *Digital Signal Processing*, Vol. 10, No. 1, January 2000, pp. 19-41.

[23] D. A. Reynolds & P. Torres-Carrasquillo (2004) The MIT Lincoln Laboratory RT-04F Diarization Systems: Applications to Broadcast Audio and Telephone Conversations, *Proceeedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[24] R. Sinha, S. E. tranter, M. J. F. Gales & P. J. Woodland (2005) The Cambridge University March 2005 speaker diarisation system, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 2437-2440.

[25] S. Theodoridis & K. Koutroumbas (2003) *Pattern Recognition (2nd edition)*, Academic Press, 2003.

[26] S. Tranter & D. Reynolds (2006) An Overview of Automatic Speaker Diarisation Systems, *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, Vol. 14, No. 5, September 2006, pp. 1557-1565.

[27] A. Tritschler & R. Gopinath (1999) Improved speaker segmentation and segments clustering using the Bayesian information criterion, *Proceedings of EUROSPEECH 99*, Budapest, Hungary, September 1999, pp. 679-682.

[28] P. C. Woodland (2002) The development of the HTK Broadcast News transcription system: An overview, *Speech Communications*, Vol. 37, No. 1-2, May 2002, pp. 47-67.

[29] C. Wooters, J. Fung, B. Peskin & X. Anguera (2004) Towards Robust Speaker Segmentation: The ICSI-SRI Fall 2004 Diarization System, *Proceeedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[30] B. Zhou & J. Hansen, (2000) Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion, *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, October 2000, pp. 714-717.

[31] X. Zhu, C. Barras, S. Meignier & J.-L. Gauvain (2005) Combining Speaker Identification and BIC for Speaker Diarization, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 2437-2440.

[32] J. Žibert & F. Mihelič (2004) Development of Slovenian Broadcast News Speech Database, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, May 2004, pp. 2095-2098.

[33] J. Žibert, F. Mihelič, J.-P. Martens, H. Meinedo, J. Neto, L. Docio, C. Garcia-Mateo, P. David, J. Zdansky, M. Pleva, A. Cizmar, A. Žgank, Z. Kačič, C. Teleki & K. Vicsi (2005) The COST278 Broadcast News Segmentation and Speaker Clustering Evaluation - Overview, Methodology, Systems, Results, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 629-632.

[34] J. Žibert, N. Pavešić & F. Mihelič (2006) Speech/Non-Speech Segmentation Based on Phoneme Recognition Features, *EURASIP Journal on Applied Signal Processing*, Vol. 2006, No. 6, Article ID 90495, pp. 1-13.

[35] J. Žibert (2006) *Obdelava in analiza zvočnih posnetkov informativnih oddaj z uporabo govornih tehologij*, PhD thesis, Faculty of electrical engineering, University of Ljubljana, 2006.

[36] J. Žibert, B. Vesnicer & F. Mihelič (2007) Novel Approaches to Speech Detection in the Processing of Continuous Audio Streams, *Robust Speech Recognition and Understanding*, M. Grimm, K. Kroschel (Eds.), I-Tech Education and Publishing, 2007, pp. 23-48.

[37] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev & P. Woodland (2004) *The HTK Book (for HTK Version 3.2)*, Cambridge University Engineering Department, Cambridge, United Kingdom, 2004.

Study of Robust and Intelligent Surveillance in Visible and Multi-modal Framework

Praveen Kumar, Ankush Mittal and Padam Kumar
Department of Electronics and Computer Engineering,
Indian Institute of Technology, Roorkee,
India 247667
E-mail: praveen.kverma@gmail.com,{ankumfec,padamfec}@iitr.ernet.in

Keywords: **v**ideo surveillance, object detection and tracking, data fusion, event detection

Received: November 12, 2007

This paper gives a review of current state of the art in the development of robust and intelligent surveillance systems, going beyond traditional vision based framework to more advanced multi-modal framework. The goal of automated surveillance system is to assist the human operator in scene analysis and event classification by automatically detecting the objects and analyzing their behavior using computer vision, pattern recognition and signal processing techniques. This review addresses several advancements made in these fields while bringing out the fact that realizing a practical end to end surveillance system still remains a difficult task due to several challenges faced in a real world scenario. With the advancement in sensor and computing technology, it is now economically and technically feasible to adopt multi-camera and multi-modal framework to meet the need of efficient surveillance system in wide range of security applications like security guard for communities and important buildings, traffic surveillance in cities and military applications. Therefore our review includes significant discussion on multi-modal data fusion approach for robust operation. Finally we conclude with discussion on possible future research directions.

Povzetek: Podan je pregled metod inteligentnega video nadzora.

1 Introduction

Security of human lives and property has always been a major concern for civilization for several centuries. In modern civilization, the threats of theft, accidents, terrorists' attacks and riots are ever increasing. Due to the high amount of useful information that can be extracted from a video sequence, video surveillance has come up as an effective tool to forestall these security problems. The automated security market is growing at a constant and high rate that is expected to sustain for decades [1]. Video surveillance is one of the fastest growing sectors in the security market due to its wide range of potential applications, such as a intruder detection for shopping mall and important buildings [2], traffic surveillance in cities and detection of military targets[3], recognition of violent/dangerous behaviors (eg. in buildings, lifts) [4] etc. The projections of the compound annual growth rate of the video-surveillance market are about 23% over 2001-2011, to touch US$670.7 million and US$188.3 million in USA and Europe, respectively [5].

An automated surveillance system attempts to detect, recognize and track objects of interest from video obtained by cameras along with information from other sensors installed in the monitored area. The aim of an automated visual surveillance system is to obtain the description of what is happening in a monitored area and to automatically take appropriate action like alerting a human supervisor, based on the perceived description. Visual surveillance in dynamic scenes, especially for humans and vehicles, is currently one of the most active research topics in computer vision [6]. For at least two decades, the scientific community has been involved in experimenting with video surveillance data to improve image processing tasks by generating more accurate and robust algorithms in object detection and tracking [7,8], human activity recognition [9,10], database [11] and tracking performance evaluation tools [12].

The most desirable qualities of a video surveillance system are (a) *robust* operation in real world scenarios, characterized by sudden or gradual changes in the input statistics and (b) *intelligent* analysis of video to assist the operators in scene analysis and event classification. In the past several research works have been carried out in many fields of video surveillance using single vision camera and indeed significant results have been obtained. But mostly they are proven to work in a controlled environment and specific contexts. A typical example is of vehicle and traffic surveillance: systems for queue monitoring, accident detection, car plate recognition etc. In a recent survey on video surveillance and sensor networks research, Cucchiara [13] reports that there are still many unsolved problems in tracking in non ideal conditions, in cluttered and unknown environment, with variable and unfavorable luminance conditions, for surveillance in indoor and outdoor spaces. Traditional approaches in dealing with these problems have focused on improving the robustness of background model and object segmentation techniques by extracting additional content from data (color, texture etc). However they have used only single modality such as visible spectrum or thermal infrared video. Visible and thermal infrared spectrums are intuitively complementary, since they capture information in emitted and reflected radiations, respectively. Thus alternative approach of integrating information from multiple video modalities has the potential to deal with such dynamically changing environment by leveraging the combined benefits whilst compensating for failures in individual modalities [14].

In addition other media streams like audio can improve analysis of visual data. For example, visual and ambient media capture two different aspects - scene and sound, respectively. In many cases where visual information is not sufficient for reliably discriminating between activities, there is often audio stimulus that is extremely important for a particular classification or anomaly detection task [15].

Automatic intelligent analysis of incoming video data on-line is required because firstly it is practically infeasible to manually supervise huge amount of video data (especially with multiple cameras) and secondly, off-line analysis completely precludes any possibility of taking immediate action in the likely happening of an abnormal event, particularly in critical applications. Several intelligent activity/ event detection methods are being proposed as the behavior patterns of real life scenario still remain challenge for the research community.

Therefore our emphasis in this paper is to discuss the existing (and proposed) techniques and provide summary of progress achieved in the direction of building robust and intelligent surveillance system. The paper's scope goes beyond traditional vision based framework to multi-modal framework. In several places, we briefly review some related concepts in automated surveillance system to put everything in proper context. For detailed discussion on studies in those related areas, reviews are available as follows: Background subtraction techniques [16], tracking of people and body parts [17], face recognition [18], gesture recognition [19], issues in automated visual surveillance [20], multimedia and sensor networks [13], distributed surveillance systems [21] and a detailed review of techniques in all the stages in the general framework of visual surveillance [6].

The rest of the paper is organized as follows. Section 2 gives an overview of automated visual surveillance, its evolution and practical issues. Section 3 discusses computer vision techniques in and beyond visual spectrum that have been developed for object detection and tracking. Section 4 reviews the work related to data fusion in multi modal framework (including visible, infrared and audio). Section 5 covers the activity recognition and behavior understanding approaches for event detection. Finally section 6 concludes the paper by summarizing the discussion and analyzing some possible future research directions.

2 Overview of Automated Visual Surveillance System

The general framework of an automatic video surveillance system is shown in Figure1. Video cameras are connected to a video processing unit to extract high-level information identified with alert situation. This processing unit could be connected throughout a network to a control and visualization center that manages, for example, alerts. Another important component is a video database and retrieval tool where selected video segments, video objects, and related contents can be stored and inquired. In [6, 22], a good description of video object processing in surveillance framework is presented. The main video processing stages include background modeling, object segmentation, object tracking, behaviors and activity analysis. In multi camera scenario, fusion of information is needed, which can take place at any level of processing. Also these cameras may be of different modality like thermal infrared, near infrared, visible color camera etc so that multi spectral video of the same scene can be captured and the redundant information may be used to improve the robustness of the system against dynamic changes in environmental conditions.

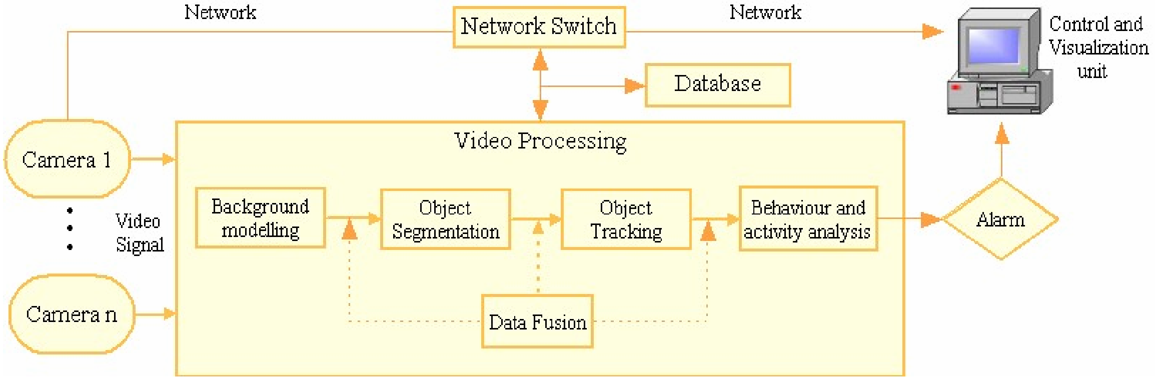


Figure1: General framework of automated visual surveillance system

2.1 Evolution of Surveillance systems

"First generation" video-based surveillance systems started with analog CCTV systems, which consisted of a number of cameras connected to a set of monitors through automated switches. In [23], for example, integration of different CCTV systems to monitor transport systems is discussed. But the human supervision being expensive and ineffective due to widespread deployment of such systems, they are more or less used as a forensic tool to do investigation after the event has taken place. By combining computer vision technology with CCTV systems for automatic processing of images and signals, it becomes possible to proactively detect alarming events rather than passive recording. This led to the development of semi-automatic systems called "second generation" surveillance systems, which require a robust detection and tracking algorithm for behavioral analysis. For example, the real-time visual surveillance system W4 [7] employs a combination of shape analysis and tracking, and constructs models of people's appearances in order to detect and track groups of people as well as monitor their behaviors even in the presence of occlusion and in outdoor environments. Current research issues in such systems are mainly real time robust computer vision algorithms and automatic learning of scene variability and patterns of behaviors.

Third generation surveillance system is aimed towards the design of large distributed and heterogeneous (with fixed, PTZ, and active cameras) surveillance systems for wide area surveillance like monitoring movement of military vehicles on borders, surveillance of public transport etc. For example the Defense Advanced Research Projection Agency (DARPA) supported the Visual Surveillance and Monitoring (VSAM) project [24] in 1997, whose purpose was to develop automatic video understanding technologies that enable a single human operator to monitor behaviors over complex areas such as battlefields and civilian scenes. The usual design approach of these vision systems is to build a wide network of cooperative multiple cameras and sensors to enlarge the field of view. From an image processing point of view, they are based on the distribution of processing capacities over the network and the use of embedded signal processing devices to give the advantages of scalability and robustness potential of distributed systems. The main research problems involved in such systems are: integration of information obtained from different sensors, establishing signal correspondence in space and time, coordination and distribution of processing task and video communication etc.

Recently, the rapid emergence of wireless networks and proliferation of networked digital video cameras have favorably increased the opportunity for deploying large scale Distributed Video Surveillance (DVS) systems on top of existing IP-network infrastructure. Many commercial companies now offer IP-based surveillance solutions. For example companies like Sony and Intel have designed equipments like smart cameras; Cisco provides many networking devices for video surveillance. All this has led to the latest step in the evolution of video-surveillance systems i.e migration to digital IP-based surveillance and recently to wireless interconnection network. Figure 2 shows a general DVS network architecture, where there are several video sensors/cameras distributed over a wide area, with smaller groups under a local base station called Processing proxy server (PPS). A PPS collects video streams from many such video cameras through a wireless (mostly) or wired LAN or mesh network. These servers are equipped with computational power to perform necessary machine vision processing and data filtering to analyze the video stream and identify alert situations. These servers then transmit the video data to different users the backbone internet network.

2.2 Practical issues in Real World Scenario

Despite much advancement in the field, realizing practical an end-to-end video surveillance system in a real world scenario remains a difficult task due to the following issues:

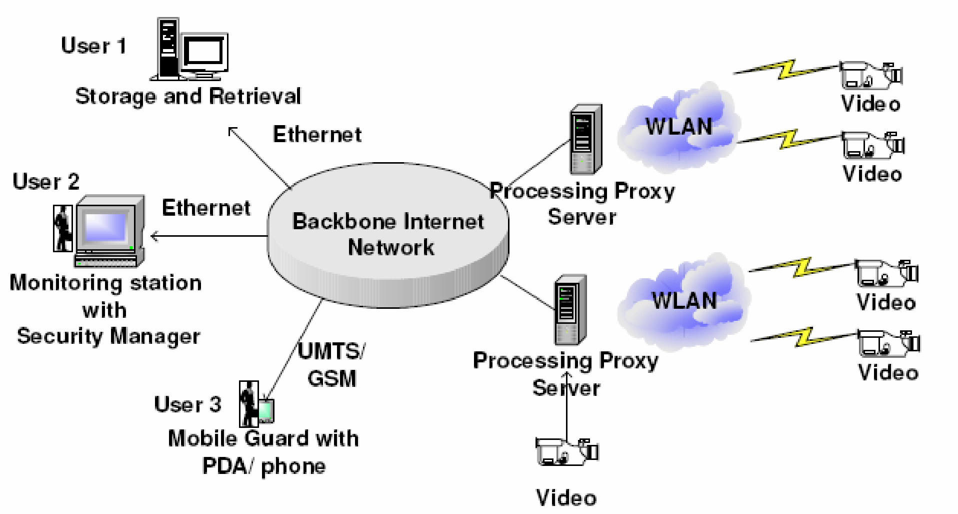


Figure 2: Distributed Video Surveillance Network Architecture

1. Robustness: Real world scenarios are characterized by sudden or gradual changes in the input statistics. A major challenge for real world object detection and tracking is the dynamic nature of real world conditions with respect to illumination, motion, visibility, weather change etc. As pointed out in [22], achieving robust algorithms is a challenge especially (a) under illumination variation due to weather conditions or lighting changes, for example, in outdoor scene, due to movement of clouds in sky and in an indoor scene, due to opening of doors or windows; (b) under view changes; (c) in case of multiple objects with partial or complete occlusion or deformation; (d) in the presence of articulated or non-rigid objects; (e) in case of shadow, reflections, and clutter; and (f) with video noise (e.g., Gaussian white). Figure 3 shows scenarios with (a) low light and illumination variation (b) video noise (c) boat among moving waves and (d) car object with moving background of vegetation. Significant research and advancement in solving these difficulties have been achieved but still the problem is unsolved in generic situation with dynamically varying environmental conditions and there is lack of generic multimodal framework to achieve system robustness by data fusion.

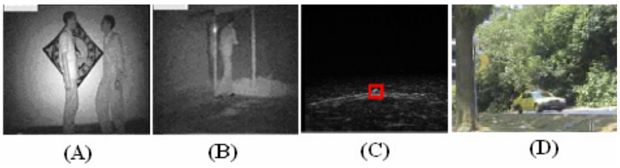


Figure 3: Some examples of complex real world situation for object detection

2. Intelligent: With the advances in sensor technology, surveillance cameras and sound recording systems are already available in banks, hotels, stores and highways, shopping centers and the captured video data are monitored by security guards and stored in archives for forensic evaluation. In a typical system, a security guard watches 16 video channels at the same time and may miss many important events. There is a need of intelligent, (semi-) automated video analysis paradigms to assist the operators in scene analysis and event classification. Event detection is a key component to provide timely warnings to alert security personnel. It deals with mapping motion patterns to semantics (e.g., benign and suspicious events). However detecting semantic events from low-level video features is major challenge in real word situation due to unlimited possibilities of motion patterns and behaviors leading to well known semantic gap issue. Furthermore, suspicious motion events in surveillance videos happen rather infrequently and the limited amount of training data poses additional difficulties in detecting these so-called rare events [25].

3. Real timeliness: A useful processing algorithm for surveillance systems should be real time, i.e., output information's, such as events, as they occur in the real scene [22]. Requirement of accuracy and robustness result in computational intensive and complex design of algorithms which makes real time implementation of system a difficult task.

4. Cost effective: For feasible deployment in a wide variety of real world surveillance applications ranging from indoor intrusion detection to outdoor surveillance of important buildings etc, a cost effective framework is required.

3 Computer Vision techniques for visual surveillance tasks

This section summarizes the research that addresses the basic computer vision problems in video surveillance like object detection and tracking. These modules constitute the low level building block necessary for any surveillance system and we briefly outline the most popular techniques used in these modules. We also present the advances made in computer vision techniques both in and beyond the visible spectrum (thermal infrared etc.) to give motivation for the discussion on data fusion in the next section.

3.1 Object Detection

Nearly every visual surveillance system starts with object detection. Object detection aims at segmenting regions corresponding to moving objects such as vehicles and humans from the rest of an image. Detecting moving regions provides a focus of attention for later processes such as tracking and behavior analysis because only these regions need be considered in the later processes. There are two main conventional approaches to object detection: 'temporal difference' and 'background subtraction'. The first approach consists in the subtraction of two consecutive frames followed by thresholding. The second technique is based on the subtraction of a background or reference model and the current image followed by a labeling process. After applying one of these approaches, morphological operations are typically applied to reduce the noise of the image difference (See figure 4 and 5). The temporal difference technique is very adaptive to changes in dynamic environment and another advantage is that it does not make assumptions about the scene. However, it can be problematic that only motion at edges is visible for homogeneous objects. On the other hand, background subtraction has better performance extracting object information but it is sensitive to dynamic changes in the environment.

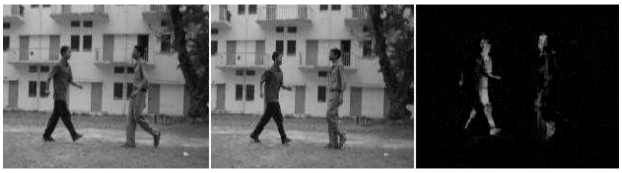


Figure 4: Example of object detection using temporal differencing technique

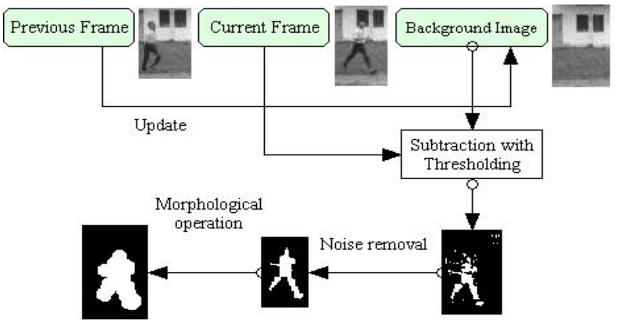


Figure 5: Object Detection using background subtraction technique

Background modeling assumes that the video scene is composed of a relatively static model of the background, which becomes partially occluded by objects that enter the scene. These objects (usually people or vehicles) are assumed to have features that differ significantly from those of the background model (their color or edge features, for example). The terms *foreground* and *background* are not scientifically defined however and thus their meaning may vary across applications. For example, a moving car should usually be considered as a foreground object but when it parks and remains still for a long period of time, it is expected to become background. Also, not all moving objects can be considered foreground. The simplest approach is to record an image when no objects are present and use this image as the background model. However, continuous updating of the model is required to make the foreground extraction more robust to the gradual changes in lighting and movement of static objects that are to be expected in outdoor scenes. Unfavorable factors, such as illumination variance, shadows and shaking branches, bring many difficulties to the acquirement and updating of background model. Background modeling is a very active research area and several techniques have been proposed to deal with various problems. A good overview of the most frequently cited background modeling algorithms is given in [16]. A comparison between various background modeling algorithms is given in [26], as well as a discussion on the general principles of background maintenance systems.

A typical approach for modeling background in outdoor conditions is using Gaussian model that models the intensity of each pixel with a single Gaussian distribution [27] or with more than one Gaussian distribution. The algorithm described in [28] models each pixel as a sum of K Gaussian distributions in RGB space ($1 \leq K \leq 5$). Each pixel's background model is updated continuously, using online estimation of the parameters. This model is well suited to cater for pixels whose background model has a multimodal distribution, such as vegetation or water. The model is unable to distinguish between foreground objects and shadows, however, and also is quite slow to initialize. The algorithm used in the *W*4 [6] system works on monochrome video and marks a pixel as foreground if:

$$|M - It| > D \text{ or } |N - It| > D \text{ (1)} \qquad (1)$$

where the (per pixel) parameters M, N, and D represent the minimum, maximum, and largest inter-frame absolute difference observed in the training frames. It also detects when its background model is invalid by detecting when 80% of the image appears as foreground. To rectify this, is re-enters a training mode to correct the background model. However reliable background modeling is difficult to achieve in certain scenarios. For example, in a crowded room with many people, the background may only ever be partially visible. Another problematic scenario is in a scene with low levels of lighting, such as a night-time scene with only street lighting. The movement (or apparent movement) of background objects is problematic too. Examples of this include moving trees and vegetation, flickering computer or TV screens, flags or banners blowing in the wind, etc.

Apart from common approached discussed above, techniques based on optic flow is useful in motion segmentation where a motion vector is assigned to every pixel of the image by comparison of successive frames. Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware. More detailed discussion of optical flow can be found in Barron's work [29].

3.2 Object Tracking

Once objects have been detected, the next logical step is to track these detected objects. Tracking has a number of benefits. Firstly, the detection phase is quite computationally expensive, so by using tracking, the detection step does not need to be computed for each frame. Secondly, tracking adds temporal consistency to sequence analysis because otherwise, objects may appear and disappear in consecutive frames due to detection failure. Also, tracking can incorporate validity checking to remove false positives from the detection phase. Thirdly, if tracking multiple objects, detection of occlusion is made easier, as we expect occlusion when two or more tracked object move past each other (as shown in figure 6). Object motion can be perceived as a result of either camera motion with a static object, object motion with static camera, or both object and camera moving. Tracking techniques can be divided into two main approaches: 2-D models with or without explicit shape models and 3-D models. For example, in [30] the 3-D geometrical models of a car, a van and a lorry are used to track vehicles on a highway. The model-based approach uses explicit a priori geometrical knowledge of the objects to follow, which in surveillance applications are usually people, vehicles or both. In [6], author use a combination of shape analysis and along with 2D Cardboard Model for representing and tracking the different body parts. Along with second order predictive motion models of the body and its parts, they used

Cardboard Model to predict the positions of the individual body parts from frame to frame.

A common tracking method is to use a filtering mechanism to predict each movement of the recognized object. The filter most commonly used in surveillance systems is the Kalman filter [31]. Fitting bounding boxes or ellipses, which are commonly called 'blobs', to image regions of maximum probability is another tracking approach based on statistical models. In [27] the author models and tracks different parts of a human body using blobs, which are described in statistical terms by a spatial and color Gaussian distribution. In some situations of interest the assumptions made to apply linear or Gaussian filters do not hold, and then nonlinear Bayesian filters, such as extended Kalman filters (EKF) or particle filters have been proposed. A good tutorial on non linear tracking using particle filter is given in [32] where the author illustrates that in highly non-linear environments particle filters give better performance than EKF. A particle filter is a numerical method, which weights (or 'particle') a representation of posterior probability densities by resampling a set of random samples associated with a weight and computing the estimate probabilities based on these weights. Then, the critical design decision using particle filters relies on the choice of importance (the initial weight) of the density function. Appearance models [33] are another way to represent objects. It consists of an observation model (usually an image) of the tracked object, along with some statistical properties (such as the pixel variances).

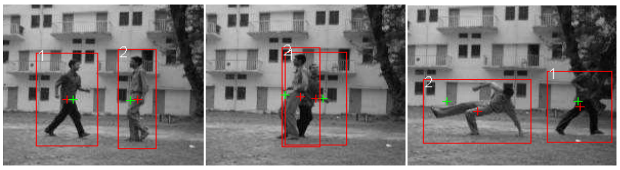


Figure 6: Object Tracking during and after Occlusion

3.3 Computer Vision beyond the Visible Spectrum

Traditionally, the majority of the computer vision community has been involved implicitly or explicitly with the development of algorithms associated with sensors that operate in the visible band of the electromagnetic spectrum [34]. In the past, imaging sensors beyond visible spectrum have been limited to special applications like remote sensing and vision based military applications, because of their high cost. Recently with the advances in sensor technologies, the cost of near and mid-infrared sensors has dropped dramatically, making it feasible for their use in more common applications like automatic video-based security and surveillance systems to enhance their capabilities.

Lin, in his 2001 technical report explores the extension of visible band computer vision techniques to infrared as well as conducts a good review of infrared imaging research [35]. Recent literature on the exploitation of near-infrared information to track humans generally deals only with the face of observed people[36] and a few are concerned with the whole body [37,38] but these approach rely on the highly limiting assumption that the person region always has a much brighter (hotter) appearance than the background. This assumption does not hold in various weather conditions and during all time. To tackle this, the author in [39] proposes a novel contour based background subtraction strategy to detect people in thermal imagery, which is robust across a wide range of environmental conditions. First of all, a standard background-subtraction technique is used to identify local region-of interest (ROI), each containing the person and surrounding thermal halo. The foreground and background gradient information within each region are then combined into a contour saliency map (highlighting the person boundary). Using a watershed-based algorithm, the gradients are thinned and thresholded into contour fragments. The remaining watershed lines are used as a guide for an A* search algorithm to connect any contour gaps. Finally, the closed contours are flood-filled to make silhouettes.

The use of infrared in pedestrian detection to reduce night time accidents is investigated in [38]. In [40], the author investigates human repetitive activity properties using thermal imagery. They employ a spatio-temporal representation, Gait Energy Image (GEI), which represent human motion sequence in a single image while preserving some temporal information. However they have developed the method for only simple activities which are repetitive in nature (like walking, running etc).

4 Data Fusion

A surveillance task using multiple modalities can be divided into two major phases: *data fusion* and *event recognition*. The data-fusion phase integrates multi-source spatio-temporal data to detect and extract motion trajectories from video sources. The event-recognition phase deals with classifying the events as to relevance for the search. This section discusses the data fusion part and the event-recognition task is discussed in the next section.

Data fusion is the process of combining data from multiple sources such that the resulting entity or decision is in some sense better than that provided by any of the individual sources. Most of the existing surveillance systems have used only one media (i.e. normal video), and therefore they do not capture different aspects of the environment. Multiple media are useful because each media captures different aspect of the environment. For example sensing environmental sound can provide reliable clue for detecting insecure events in many cases. Infrared is more informative in dark environment, especially at night. Visible and thermal infrared spectrums are intuitively complementary, since they capture information in emitted and reflected radiations, respectively. Thus combining them can be advantageous in many scenarios, especially when one modality perform poorly in detecting objects. For example visible analysis has an obvious limitation of daytime operation only and completely fails in total darkness. Additionally foggy weather condition, sudden lighting changes, shadows and color camouflage, often cause poor segmentation of actual objects and much false positive detection. Thermal infrared video is almost completely immune to lighting changes, and thus it is very robust to the above mentioned problems. However, infrared video has its unique inherent challenges due to high noise, and "Halo effect" produced by some infrared sensors, which appears as a dark or bright halo surrounding very hot or cold objects respectively. Further if people are wearing insulated clothing or infrared camera performs rapid automatic gain then it will cause foreground detection to incorrectly classify pixels. See figure 7 for illustration in two different situations. Thus by data fusion approach, it is possible to improve robustness of the system in dynamic real world conditions.

4.1 Fusion of Visible and Infrared

Depending on the application and fusion method, research in the fusion of visible and infrared imagery can be classified in two broad categories. *Image based Representational fusion* and *Video based Analytical fusion*. In *Image based Representational fusion,* the goal is to obtain best representation of the data in a single image for improved visual perception by combining multiple images to create a single fused image that somehow represents the information content of the input images. This type of fusion is generally used for remote sensing and military applications. Depending on the synergy of the information inherent in the data, it may be possible to reduce noise, to extend the field of view beyond that of any single image, to restore high frequency content, and even to increase spatial resolution [41]. Image fusion techniques have had a long history in vision. Gradient-based techniques examining gradients at multiple resolutions [42] and several region-based multi resolution algorithms have been proposed such as the pyramid approaches of [43, 44] and the wavelet-based approach of [45].

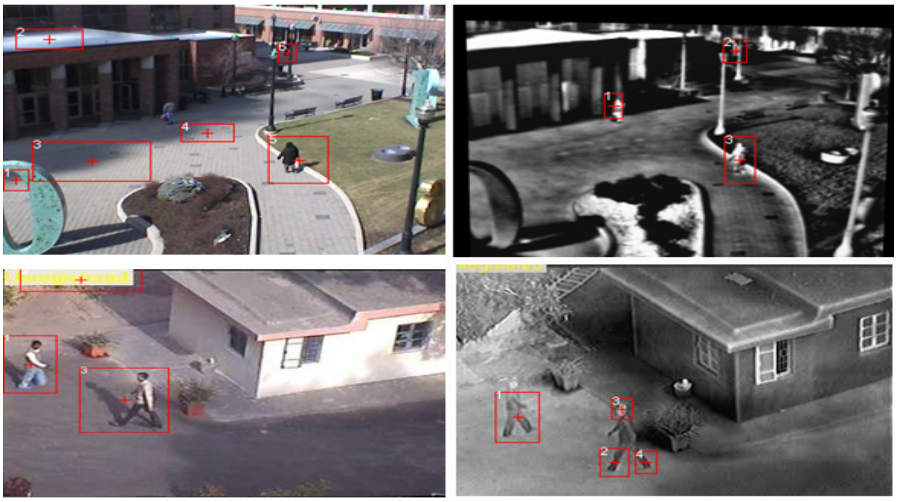


Figure 7: Visible and corresponding thermal infrared in a) Variable illumination due to cloud movement causing false detection in visible (top) b) Incorrect detection due to shadows in visible and thermally insulated clothing in infrared (bottom)

Video based Analytical Fusion, on the other hand, aims to extract knowledge by using all sources of data for better analysis, and not merely to represent the data in another way. This type of fusion methodology is required to enhance the capabilities of automatic video-based detection and tracking system for surveillance purpose. Although image fusion has received considerable attention in the past, research in the fusion of video modalities for automatic analysis, or analytical fusion is very recent. Some recent works have addressed the tracking of humans and vehicles with multiple sensors [46, 47] but issues that are involve in fusing multiple modalities for robust detection and tracking is very sparse. In [48], the fusion of thermal infrared with visible spectrum video, in the context of surveillance and security, is done at the object level. Detection and tracking of blobs (regions) are performed separately in the visible and thermal modality. An object is made up of one of more blobs, which are inherited or removed as time passes. Correspondences are obtained between objects in each modality, forming a master-slave relationship, so that the *master* (the object with the better detection or *confidence*) assists the tracking of the slave in the other modality. Their system uses many heuristics and there also seems to be many parameters to set empirically.

Davis et al. [49] propose a new contour-based background-subtraction technique using thermal and visible imagery for persistent object detection in urban settings. They perform statistical background subtraction in the thermal domain to identify the initial regions-of-interest. Color and intensity information are used within these areas to obtain the corresponding regions of-interest in the visible domain. Within each image region (thermal and visible treated independently), the input and background gradient information are combined as to highlight only the boundaries of the foreground object. The boundaries are then thinned and thresholded to form binary contour fragments. Contour fragments belonging to corresponding regions in the thermal and visible domains are then fused using the combined input gradient information from both sensors. An A* search algorithm constrained to a local watershed segmentation is then used to complete and close any contour fragments. Finally, the contours are flood-filled to make silhouettes.

In a very recent work [50], the authors use thermal infrared video with standard CCTV video for object segmentation and retrieval in surveillance video. They segment object using separate background modeling in each modality and dynamic mutual fusion based thresholding. Transferable Belief Model is used to combine the sources of information for validating the tracking of objects. Extracted objects are subsequently tracked using adaptive thermo-visual appearance. However they don't take into account the reliability of each source in the fusion process. In [51], an intelligent fusion approach using Fuzzy logic and Kalman filtering technique is discussed to track objects and obtain fused estimate according to the reliability of the sensors. Appropriate measurement parameters are identified to determine the measurement accuracy of each sensor. A comparison of multiple fusion schemes for appearance based tracking of objects using thermal infrared and visible modalities is done in [52] for different objects, such as people, faces, bicycles and vehicles.

4.2 Data Fusion Methods

For visual surveillance using multiple cameras, issues such as camera calibration and registration, establishing correspondences between the objects in different image sequences taken by different cameras, target tracking and data fusion need to be addressed. The success of information fusion depends on how well data are represented, how reliable and adequate the model of data uncertainty used and how accurate and applicable prior knowledge is. Three commonly used fusion approaches are probabilistic methods (Bayesian inference), fuzzy logic method and belief models (Dempster-Shafer model and Transferable Belief model). The Bayesian inference method quantitatively computes the probability that an observation can be attributed to a given assumed hypothesis but lacks in ability to handle mutually exclusive hypotheses and general uncertainty [53]. Fuzzy logic methods accommodate imprecise states and variables. It provides tools to deal with observations that is not easily separated into discrete segments and is difficult to model with conventional mathematical or rule-based schemes [54]. The Belief theory generalizes Bayesian theory to relax the Bayesian method's restriction on mutually exclusive hypotheses, so that it is able to assign evidence to 'propositions', i.e. unions of hypotheses. Dempster-Shafer model makes a closed world assumption, so it assigns a belief of empty set to zero. The reasoning model assumes completeness of the frame of discernment meaning that the frame includes all hypotheses. But it can very well happen that some hypotheses, because of measurements are excluded from frame of discernment or unknown. In this way meaning of empty set is changed corresponding not only for impossibilities but also for unknown possibilities [55]. This kind of approach is called open world assumption which is considered in another belief model called Transferable Belief Model (TBM) [56]. TBM offers the flexibility to model closed world or open world assumption.

In [57], Bayesian probability theory is used to fuse the tracking information available from a suite of cues to track a person in 3D space. In [58], the authors uses TBM framework to solve the problem of data association in a multi target detection problem. It uses the basic belief mass $m(\Theta)$ as a measure of conflict and the sensors are clustered so that the conflict is minimized. But they tackle only partial problem of assessing how many objects are present and observed by the sensors. In [59], the author use TBM and Kalman filter for data fusion in object recognition system that analyses simulated FLIR and LADAR data to recognize and track aircraft. They demonstrated the results on an air to air missile based simulation system. In [60], the author proposes a hybrid multi-sensor data fusion architecture using Kalman filtering and fuzzy logic techniques. They feed the measurement coming from each sensor to separate fuzzy–adaptive kalman filters (FKF), working in parallel. The adaptation in each FKF is in the sense of adaptively adjusting the measurement noise coar1ance matrix R employing a fuzzy inference system (FIS) based on a covariance matching technique. Another FIS, which they call as fuzzy logic observer (FLO) monitors the performance of each FKF. Based on the value of a variable called Degree of Matching *(D0M)* and the matrix R coming from each FKF, the FLO assigns a degree of confidence, a number on the interval (0, 1], to each one of the FKFs output. The degree of confidence indicates to what level each FKF output reflects the true value of the measurement. Finally, a defuzzificator obtains the fused estimated measurement based on the confidence values. They demonstrated the result theoretically, by taking example of four noisy inputs.

4.3 Reliability of Sensor

In the fusion process, different sources may have different reliability and it is essential to account for this fact to avoid decreasing in performance of fusion results. The fused estimate should be more biased by accurate measurements and almost unaffected by inaccurate or malfunctioning ones. Therefore for fusing data collected from different sensors requires the determination of measurements' accuracy so that they can be fused in a weighted manner. The most natural way to deal with this problem is to establish reliability of the beliefs computed within the framework of the model selected. For example [61] discusses a method for assessing the reliability of a sensor in a classification problem within the TBM framework. In [62], the authors propose a multi-sensor data fusion method for video surveillance, and demonstrated the results by using optical and infrared sensors. The measurements coming from different sensors were weighted by adjusting measurement error covariance matrix used by the fusion filter. To estimate the reliability of the sensor they defined a metric called Appearance Ratio (AR), whose value is proportional to the strength of the segmented blobs from each sensor. The ARs are compared to determine which sensors are more informative and therefore selected to perform a specific video surveillance task. In [63], the authors discuss the principal concepts and strategies of incorporating reliability into classical fusion operators and provide good literature survey on main approaches used in fusion literatures to estimate reliability of sensor.

4.4 Audio and Video Information Fusion

Enhancing visual data with audio streams can serve manifold purpose like speaker tracking, environment sound recognition for event recognition in surveillance application etc. Environmental sound like that of breaking of glass, dog's barking, screaming of a person, fire alarm, gun firing and similar kind of sounds, if detected and recognized correctly, can give a reasonable degree of confidence in making a decision about 'secure' vs. 'insecure' state [64]. Multimedia researchers have often used early fusion strategy to perform the audio-visual fusion for various problems including speech processing [65] and recognition [66], speaker localization [67] and tracking [68, 69], and monologue detection [70]. In [68], the authors present a method that fuses 2-D object shape and audio information via importance filters. They used audio information to generate an importance sampling function, which guides the random search process of particle filter towards regions of the configuration space likely to contain the true configuration (a speaker). A recent work in [71], describes a process to assimilate data from coarse and medium grain sensors, namely video and audio, and a probabilistic framework to discriminate concurring and contradictory evidences. The authors enlarge the concept to information fusion with the definition of *information assimilation*: this process includes not only the real-time information fusion but also the integration with the past experience, represented by the surveillance information stored in the system.

Research in the field of environmental sound recognition is sparse. The majority of auditory research is centered on the identification and recognition of speech signals. Those systems that do exist, work on a very specialized domain like in [72], a system named *AutoAlert* is presented for automated detection of incidents using HMM, and Canonical Variates Analysis (CVA) to analyze both short-term and time-varying signals that characterize incidents. Cowling, in [73] provides more detailed literature survey, and it also investigates few existing techniques used for sound recognition in speech and music. He then presents a comparison on the accuracy of these techniques, when employed for the problem of non-speech environmental sound classification for autonomous surveillance.

4.5 Other Sensors/modalities

There are proximity sensors (like ultrasound devices, lasers scanners etc.) which detect objects without physical contact. Most proximity sensors emit an electromagnetic field or beam and look for changes in the field. Different targets demand different sensors. For example, a capacitive or photoelectric sensor might be suitable for a plastic target; an inductive sensor requires a metal target [74]. In [75], the authors integrate Laser Doppler Vibrometer (LDV) and IR video for remote multimodal surveillance. Their work mainly caters to remote area surveillance and their main focus was to study the application of LDV for remote voice detection, while IR imaging was used for target selection and localization. Few object tracking and visual servoing system for the visually impaired such as the *GuideCane* [76] and the *NavBelt* [77], use ultrasound or laser rangefinders to detect obstacles.

Gated imaging is another useful system for highly unfavorable visual conditions like underwater surveillance etc. Time gating is a temporal example of image formation whereby a light source is time pulse projected toward a target and the detector is time gated to accept image-forming illumination from a specific range [78]. LIDAR systems [79] time gate the receiver aperture to eliminate relatively intense backscatter originating from the water while allowing the return from the target to be detected. In [78] time gating is employed for using spatially and temporally varying coherent illumination for undersea object detection.

5 Event Detection

5.1 Human Activity Recognition

Computer analysis of human actions is gaining increasing interests, especially in video surveillance arenas where people identification and activity recognition are important. Using two important metrics: preciseness of the analysis outcome and the required video resolution to achieve the desired outcome, human identification and activity recognition can be classified into three categories. At one extreme, which is often characterized by high video resolution and a small amount of scene clutter, high fidelity outcome is achievable. Many techniques in face, gesture, and gait recognitions fall in this category, which aim to identify individuals against a pre-established database.

At the other extreme, which is characterized by low video resolution and potentially significant scene clutter, it is often not possible to achieve highly discriminative outcome. Instead, the goal is often to detect the presence, and identify the movement and interaction of people through "blob" tracking [6, 24]. The VSAM system [24] tracked the human body as a whole blob. They use a hybrid algorithm by combining adaptive background subtraction with a three-frame differencing technique to detect moving objects, and use the Kalman filter to track the moving objects over time. A neural network classifier is trained to recognize four classes: single person, group of persons, vehicles, and clutter. They also use linear discriminant analysis to further provide a finer distinction between vehicle types and colors. The VSAM system is very successful at tracking humans and cars, and at discriminating between vehicle types. But it did not put much emphasis on activity recognition; only gait analysis and simple human-vehicle activity recognition are handled.

In the middle of the spectrum, it is possible to refine the "blob" representation of a person through hierarchical, articulated models. For example, [80] describes an approach which attempts to recognize more generic activities and movement of body parts using MHI (motion-history images) to record both the segmentation result and the temporal motion information. The MHI is a *single* image composed of superimposing a sequence of segmented moving objects weighed by time. The most recent foreground pixels are assigned the brightest color while past foreground pixels are progressively dimmed. This allows the summarization of information on both the spatial coverage and the temporal ordering of the coverage of an activity. See figure 8 for illustration with few examples. The MHI does not use any structure to model human. A vector of seven moment values is computed for each MHI. Activities are recognized by finding the best match of the moment vectors between the query MHI and the training patterns. Other approaches allow main body parts, such as head, arms, torso, and legs, to be individually identified to specify the activities more precisely. A very detailed review of activity recognition approaches in these categories can be found in [8].

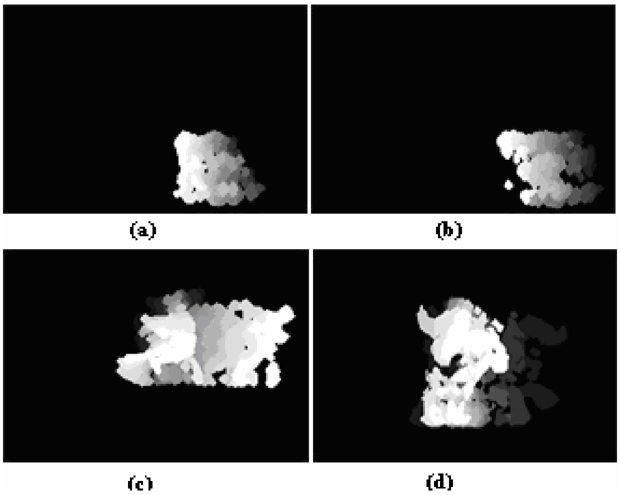


Figure 8: MHI images of person (a) walking (b) Running, (c) picking an object and (d) fighting

5.2 Semantic Information Extraction for Behavior Understanding

Understanding of behaviors may simply be thought as the classification of time varying feature data, i.e., to analyze the video to extract some feature vectors and to classify this time-varying feature data. During the recognition phase, extracted unknown test feature vector set is compared to a group of labeled reference feature vector sets representing typical human actions.

Feature extraction process is very important to achieve good results. It is impossible to train and perform classification using the currently available classification engines. The size of the feature vector should be as small as possible to have computational efficiency. At the same time it should represent each action very accurately. For example, the center of mass of the tracked object in image frame of the video can be used as a feature vector in a security application in which moving persons entering or leaving a building. In this simple problem, the "vocabulary" consists of a person leaving the building and a person entering a building and the center of mass information consisting of the horizontal and vertical coordinates in an image of the video may be good feature vector for this problem. On the other hand, detecting an 'assault' case where a person falls on the ground and other runs will require other additional parameters in the feature set. For example, so-called snaxels of an active snake contour of the human body can be added to the feature vector to distinguish a fallen person from a person standing. Compactness of the contour boundary, the speed of the center of the mass etc can be also used to distinguish the normal action of walking and the abnormal action of a falling and running. As a rule of thumb, model parameters are selected as entries of the feature vector in a model based tracking approach. Since a video consists of sequence of images a sequence of feature vectors are obtained to characterize the motion of a person(s).

5.3 Pattern Analysis and Classification Methods

Several generative and discriminative models have been proposed for modeling and classifying activity patterns. Some of the most widely used ones are Dynamic time Warping (DTW), Hidden Markov Models (HMM), Time Delay Neural Network (TDNN), and Finite State Machine (FSM) network.

a) Dynamic time warping: DTW is a template based dynamic programming matching technique widely used in the algorithms for speech recognition. It has the advantage of conceptual simplicity and robust performance, and has been used recently in the matching of human movement patterns [81]. For instance, Bobick *et al.* [82] use DTW to match a test sequence to a deterministic sequence of states to recognize human gestures. Even if the time scale between a test sequence and a reference sequence is inconsistent, DTW can still successfully establish matching as long as the time ordering constraints hold.

b) Hidden Markov Models: HMMs are stochastic finite state machines [83]. In the context of human motion analysis, a finite state Markov model is assigned for each possible scenario and its parameters are trained with feature vectors of this typical human action. The training process is an off-line iterative algorithm called Baum-Welch algorithm [84]. Here , the number of states of a HMM must be specified, and the corresponding state transition and output probabilities are optimized in order that the generated symbols can correspond to the observed image features of the examples within a specific movement class. During the classification or recognition phase, the test feature vector set is applied to all of the Markov models and output probabilities are computed. The Markov model producing the highest probability is determined and the corresponding human action scenario is selected as the result. HMMs generally outperform DTW for undivided time series data, and are therefore extensively applied to behavior understanding. In [85], authors describe an activity recognition process for visual surveillance of wide areas and experimented with image sequences acquired from an archaeological site with actors perform both legal and illegal actions. The activity recognition process is performed in three steps: first of all the binary shape of moving people are segmented, then the human body posture is estimated frame by frame and finally, for each activity to be recognized, a temporal model of the detected postures is generated by Discrete Hidden Markov Models

c) Finite state machine: The most important feature of a FSM is its state-transition function. The states are used to decide which reference sequence matches with the test sequence. However it requires hand crafted heuristic rules based on context knowledge. For example in [86], the authors propose a framework for unsupervised learning of usual activity patterns and detection of unusual activities based on a model of multi-layered finite state machines. They considered two different approaches for different scenario. First approach is unsupervised learning of usual activity patterns and detection of unusual activities (which are not recognized as normal). Other approach is to explicitly program the FSM or training using supervised learning for recognition of situation specific activities like unattended baggage detection.

d) *Time-delay neural network (TDNN):* TDNN is also an interesting approach to analyzing time-varying data. In TDNN, delay units are added to a general static network, and some of the preceding values in a time-varying sequence are used to predict the next value. As larger data sets become available, more emphasis is being placed on neural networks for representing temporal information. TDNN has been successfully applied to hand gesture recognition [87] and lip-reading [88].

Other related schemes including Dynamic Bayesian Network (DBN) and the Support Vector Machines (SVM) are also now being actively used in pattern analysis and classification problems.

Considering the limited training data for unusual events and that the distinction between two unusual events can be as large as those between unusual events and usual events, it is not feasible to train a general model for the unusual events. Therefore alternative approach that some people have taken is to train a model for usual events and events that deviate significantly from the usual event model are considered unusual. For example a simple ATM surveillance scenario is shown in figure 9, where FSM can be a useful way to discriminate between normal and abnormal patterns. The first row shows frames corresponding to normal transaction. Second and third row shows event corresponding to vandalism and robbery.

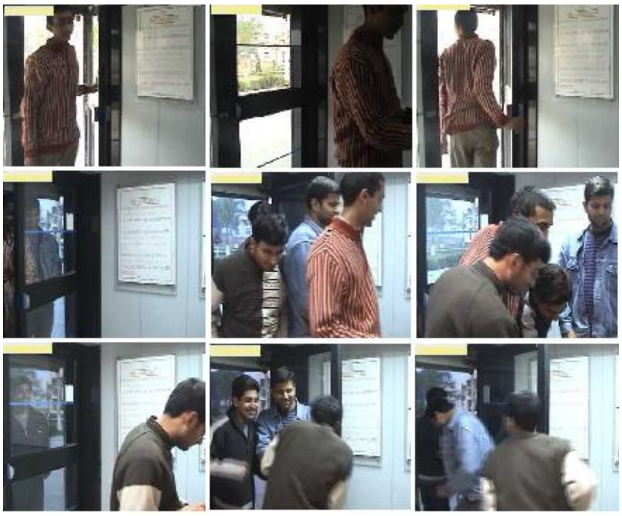


Figure 9: Sampleshots for events in a Bank ATM Surveillance

Figure 10 shows a possible FSM for activities which will be considered normal if the transitions terminated at the exit node. If exit is made by another node due to any deviant pattern, the FSM will flag an abnormal event.

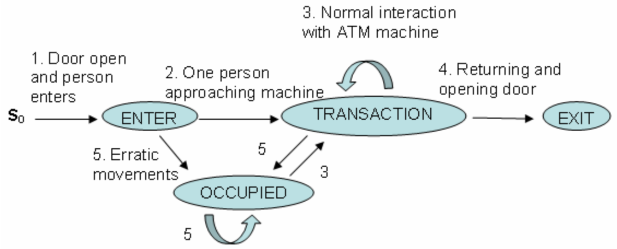


Figure 10: A possible FSM for Bank ATM Surveillance

Boiman and Irani [89] proposed to model the set of usual events as an ensemble of spatial-temporal image patches, and detect irregularity in a test video by evaluating the similarity between the test ensemble with the training ensemble. Zhang *et al.* [90] used a semi-supervised approach to train models for both usual and unusual events. They start from an ergodic hidden Markov model (HMM) for usual events. If a test event does not fit the model, they classify it as unusual and branch the usual event model to refit the usual event. This approach has a disadvantage that it may give high false positive alarm because in a practical scenario, even during the normal course of activity, unusual deviations are very likely to happen without potential threat.

6 Conclusions and Future Research Developments

Event though significant progress have been made in computer vision and other areas, there are still major technical challenges to be overcome before the dream of reliable automated surveillance is realized. These technical challenges are compounded by practical considerations such as robustness to unfavorable weather and lighting conditions, intelligent video processing for event detection and efficiency in terms of real time operation and cost. Most surveillance systems operate using single modality and lack robustness because they are limited to a particular situation. Different media source like audio, video and thermal infrared gives complementary/supplementary surveillance information of the environment. The literature survey on multi-modal data fusion shows that effective fusion of the information coming from different media streams can give robustness to real world object detection, tracking and event detection. Simultaneous fusion of thermal infrared and visible spectrum can give following advantage:

- Improved robustness against camouflage as foreground object are less likely to be of similar *color and temp* to the background
- Providing features that can be used for classification or retrieval of objects in large surveillance video archives
- Extracting a signature of an object in each modality, which indicates how useful each modality is in tracking that object

However, the key challenge for future research is to: 1) develop analysis techniques to automatically determine the reliability of data source and 2) develop a suitable fusion methodology that would intelligently utilize the information provided by these two modalities to get the best possible output. In [91], these challenges are addressed by employing Transferable Belief Model (TBM) and Kalman filter. TBM is used to determine the validity of a foreground region, detected by each source, for tracking. Kalman filter is used for the dual purpose of tracking the objects over time and fusing the measurements of the positions of the target obtained from different sensors, according to their reliability.

One of the main objectives of visual surveillance is to analyze and interpret individual behaviors and interactions between objects for event detection. Recently, related research has still focused on some basic problems like recognition of standard gestures and simple behaviors. Some progress has been made in building the statistical models of human behaviors using machine learning. However behavior recognition is complex, as the same behavior may have several different meanings depending upon the scene and task context in which it is performed. An alternative approach can be of providing *selective focus-of-attention* to the human supervisor by discriminating unusual or anomalous event from normal ones. Use of audio is encouraging in this respect because in many cases when the visual information required for a particular task is extremely subtle, audio stimulus is extremely salient for a particular anomaly detection task. Moreover, audio features can help in mining interesting patterns from the scene. Multimedia data mining has been applied for detection of events in sports video (like goal event in soccer [25] etc.) but it has not been systematically applied for surveillance videos. This requires development of a novel framework of low level feature extraction, advanced temporal analysis and multimodal data mining methods.

An obvious requirement for surveillance system is real time performance. If the system has to process signals from multiple sensor and modalities, then the required processing is multiplied. Moreover requirement for robustness and accuracy tend to make the algorithm design complex and highly computational. Generally commercial products employ embedded signal processing devices and high performance dedicated processors for faster processing but simultaneously increase the system cost heavily. Past research have focused on optimizing the low level image processing algorithms, reducing the feature space etc and recently researches on distributed surveillance system have tried to distribute the processing on the network. However there is lack of any systematic design and real time implementation of video processing algorithms using a network of multiple processors. Recent research and development in Grid technology can provide an alternative architecture for such implementation and research needs to be done to explore the feasibility of such innovative approach.

References

[1] DataMonitor. Global digital video surveillance markets: Finding future opportunities as analog makes way for digital. Market research report (July 2004). www.mindbranch.com/products/R313-6950.html.

[2] T. Bodsky, R. Cohen, E. Cohen-Solal, S. Gutta, D. Lyons, V. Philomin, and M. Trajkovic (2001). 'Visual surveillance in retail stores and in the home', *in: 'Advanced Video-based Surveillance Systems'*. Kluwer Academic Publishers, Boston, , Chapter 4, pp. 50–61

[3] J.M. Ferryman, S.J. Maybank, and A.D. Worrall (2000). 'Visual surveillance for moving vehicles', *Int. J. Comput. Vis.*, 37, (2), Kluwer Academic Publishers, Netherlands, pp. 187–19731

[4] R. Cucchiara, C. Grana, A. Patri, G. Tardini, and R. Vezzani (2004). Using computer vision techniques for dangerous situation detection in domotic applications. *Proc. IEE Workshop on Intelligent Distributed Surveillance Systems,* London, pp. 1–5.

[5] Frost & Sullivan report, (August 2005); www.frost.com

[6] W. Hu, T. Tan, L. Wang, and S. Maybank (August 2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and cybernetics*, 34(3):334–350,

[7] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who, when, where, what: A real time system for detecting and tracking people. *In Third Face and Gesture Recognition Conference*, pp. 222-227.

[8] W. Niu, L. Jiao, D. Han, and Y. Wang (2003). Real-Time Multi-Person Tracking in Video Surveillance. *Proceedings of the Pacific Rim Multimedia Conference*, Singapore.

[9] A. Bobick and J. Davis (December 1996). Real time recognition of activity using temporal templates. IEEE workshop on Applications of Computer Vision, Sarasota, FL, 4 pages.

[10] J. Gao, A. G. Hauptmann and H. D. Wactlar. Combining Motion Segmentation with Tracking for Activity Analysis (2004). *The Sixth International Conference on Automatic Face and Gesture Recognition* (FGR'04), pp. 699-704, Seoul, Korea, May 17-19.

[11] E. Stringa,, and C.S. Regazzoni. Content-based retrieval and real-time detection from video sequences acquired by surveillance systems. *Int. Conf. on Image Processing, Chicago*, 1998, pp. 138–142

[12] J. Black, T. Ellis and P. Rosin. A novel method for video tracking performance evaluation (2003). *The Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October, France, pp. 125–132.

[13] R. Cucchiara (2005). Multimedia surveillance systems. In *VSSN 05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks, New York NY, USA*, pages 3-10.

[14] C. ´O Conaire, E. Cooke, N. O'Connor, N. Murphy, and A. F.Smeaton (2005). Fusion of infrared and visible spectrum video for indoor surveillance. *In International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS),* Montreux, Switzerland, April.

[15] C. stauffer. Automated Audio-Visual Activity Analysis. *CSAIL Technical Reports*, MIT. http://hdl.handle.net/1721.1/30568

[16] A. M. McIvor (2000). Background subtraction techniques. In *Image and Vision Computing, Hamilton, New Zealand*, Nov.

[17] J. J. Wang and S. Singh (2003). Video analysis of human dynamics – a survey. *Real-Time Imaging 9(5):* 321-346.

[18] R. Chellappa, C. L. Wilson and S. Sirohey (1995). Human and machine recognition of faces: a survey. *Proc. of the IEEE*, vol. 83, No. 5, pp705-740.

[19] V. I. Pavlovic, R. Sharma and T. S. Huang (1997). Visual interpretation of hand gestures for human computer interaction: a review. *IEEE Transactions on PAMI*, vol.19, no.7, pp.677-695, July.

[20] A. R. Dick and M. J. Brooks (2003). Issues in Automated Visual Surveillance. DICTA: 195-204.

[21] M. Valera and S.A. Velastin (2005). Intelligent distributed surveillance systems: a review vision. *In Image and Signal Processing, IEE Proceedings*, volume 152, pages 192 – 204, April.

[22] A. Amer and C. Regazzoni. *Editorial:* Introduction to the special issue on video object processing for surveillance applications. *Real-Time Imaging* Vol. 11, pp: 167–171, 2005.

[23] Nwagboso, C (1998). User focused surveillance systems integration for intelligent transport systems', *in 'Advanced Video-based Surveillance Systems'*. Kluwer Academic Publishers, Boston, Chapter 1.1, pp. 8–12.

[24] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y.Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L.Wixson (2000). A system for video surveillance and monitoring. *Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep.*, CMU-RI-TR-00-12.

[25] M. Chen, S-C. Chen, M-L. Shyu, and K. Wickramaratna. Semantic Event Detection via Multimodal Data Mining. *IEEE Signal Processing Magazine*. pages 38-46, march 2006.

[26] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers (1999). Wallflower: Principles and practice of background maintenance. In *Proceedings of the Seventh IEEE International Conference on Computer IEEE Comput. Soc.*, volume 1, pages 255–261.

[27] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland (1997). Pfinder: real-time tracking of the human body. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, July.

[28] C. Stauffer and W.E.L. Grimson (1999). Adaptive background mixture models for real-time tracking. In *Proceedings of CVPR99*, pages II:246–252.

[29] J. Barron, D. Fleet, and S. Beauchemin (1994) .Performance of optical flow techniques. *Int. J. Comput.Vis.*, vol. 12, no. 1, pp. 42–77.

[30] J.M. Ferryman, S.J. Maybank, and A.D. Worrall (2000). Visual surveillance for moving vehicles, Int. J. Comput. Vis., 37, (2), Kluwer Academic Publishers, Netherlands, pp. 187–197

[31] G. Welch and G. Bishop (2002). An Introduction to the Kalman Filter. *UNC-Chapel Hill, TR 95-041*.

[32] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp (2002). A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Process*., 50, (2), pp. 174–188

[33] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. In *2nd IEEE Int. Workshop on PETS, Kauai, Hawaii, USA*, Dec 2001.

[34] B. Bhanu, I. Pavlidis, R. Hummel. *Guest Editorial:* Special issue on computer vision beyond the visible spectrum. Machine Vision and Applications (2000) 11: 265–266

[35] S.-S. Lin. Review: Extending visible band computer vision techniques to infrared band images. *Technical report, GRASP Laboratory, Computer and Information Science Department*, University of Pennsylvania, 2001.

[36] C. K. Eveland, D. A. Socolinsky, L. B. Wolff. Tracking human faces in infrared video. *Image and VisionComputing*, Vol. 21, pp. 579-590, 2003.

[37] M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, M. Meinecke. Pedestrian Detection in Infrared Images. *Proc.IEEE Intelligent Vehicles Symposium 2003*, pp. 662-667, Columbus (USA), June 2003.

[38] F. Xu, X. Liu, and K. Fujimura. Pedestrian Detection and Tracking With Night Vision. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 1, pp. 63-71, March 2005.

[39] J. Davis and V. Sharma. Robust detection of people in thermal imagery. In *Proc. Int. Conf. Pat. Rec.*, pages 713–716, 2004

[40] J. Han and B. Bhanu. Human activity recognition in thermal infrared imagery. *Computer Vision and Pattern Recognition*, 20-26 June, 2005.

[41] McDaniel, R., Scribner, D., Krebs, W., Warren, P., Ockman, N., McCarley, J. (1998). Image fusion for tactical applications. Proceedings of the SPIE - Infrared Technology and Applications XXIV, 3436, 685-695.

[42] J. Li. Spatial quality evaluation of fusion of different resolution images. *International Archives of Photogrammetry and Remote Sensing*, Vol.33, 2000.

[43] A. Toet. Heirarchical image fusion. *Machine Vision and Applications*, 3:1–11, 1990

[44] M. Pavel, J. Larimer, and A. Ahumada. Sensor fusion for synthetic vision. In *Conference on Computing in Aerospace, AIAA, 1991.*

[45] H. Li, B. Manjunath, and S. Mitra. Multisensor image fusion using the wavelet transform. In *Graphical Models and Image Processing*, volume 57, pages 234–245, 1995

[46] A. Utsumi, H. Mori, J. Ohya, and M. Yachida. Multipleview-based tracking of multiple humans. In *Proceedings of the 14th ICPR*, pages 597–601, 1998

[47] A. Nakazawa, H. Kato, and S.Inokuchi. Human tracking using distributed vision systems. In *Proceedings of the 14th ICPR*, pages 593–596, 1998

[48] H. Torresan, B. Turgeon, C. Ibarra-Castanedo, P. Hébert, X. Maldague,. Advanced Surveillance Systems: Combining Video and Thermal Imagery for Pedestrian Detection. In *Proc. of SPIE, Thermosense XXVI*, volume 5405 of *SPIE*, pages 506–515, April 2004.

[49] J. W. Davis and V. Sharma**.** Fusion-Based Background-Subtraction using Contour Saliency. *Computer Vision and Pattern Recognition*, 20-26 June, 2005

[50] C.O. Conaire, N. O.Connor, E. Cooke, A. Smeaton. Multispectral Object Segmentation and Retrieval in Surveillance Video. *To appear in International Conference on Image Processing,* 2006.

[51] P. Kumar, A. Mittal and P. Kumar. Fusion of Thermal Infrared and Visible Spectrum Video for Robust Surveillance. In *5th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, LNCS 4338, pp. 528 – 539, 2006.

[52] Ó. Conaire, C. O'Connor, N.E. Cooke and E. Smeaton. Comparison of fusion methods for thermo-visual surveillance tracking. *In International Conference on Information Fusion,2006.*

[53] D. L. Hall and J. Llinas. An introduction to multisensor fusion. *In Proceedings of the IEEE: Special Issues on Data Fusion,* pages 85(1):6–23, January 1997.

[54] R. R. Brooks and S.S. Iyengar. Multi-sensor fusion: fundamentals and applications with software. Upper Saddle River, N.J. : Prentice Hall PTR, 1998.

[55] S. Nazarko. Evaluation of data fusion methods using kalman filtering and TBM. Maters thesis. University of Jyväskylä. 2002.

[56] Ph. Smets. The Transferable Belief Model for Quantified Belief Representation*. In Handbook of defeasible reasoning and uncertainty management* systems. Gabbay D. M. and Smets Ph. Eds. Vol. 1, Kluwer, Doordrecht, 1998, pg. 267-30*1*.

[57] G. Loy, L. Fletcher, N. Apostolo, and A. Zelinsky. An adaptive fusion architecture for target tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, 2002.

[58] A. Ayoun, and P. Smets. Data association in multi-target detection using the transferable belief model. *International Journal of Intelligent Systems*, 16:1167-1182. 2001

[59] G. Powell, D. Marshall, R. Milliken, K. Markham. A Data Fusion System for Object Recognition based on Transferable Belief Models and Kalman Filters. *In `Proceedings of 7th International Conference on Information Fusion.* Sweden, Pp 54-61. 2004

[60] P.J. Escamilla-Ambrosio, N. Mort. A Hybrid Kalman Filter - Fuzzy Logic Architecture for Multisensor Data Fusion. *Proceedings of the 2001 IEEE International Symposium on Intelligent Control* , pp. 364-369 , 2001

[61] Z. Elouedi, K. Mellouli, P. Smets. Assessing sensor reliability for multisensor data fusion within the transferable belief model. *IEEE Transactions on Systems, Man and Cybernetics, Part B,* page 782-787. 2004

[62] L. Snidaro, G.L. Foresti, R. Niu, and P.K. Varshney. Sensor fusion for video surveillance. *Proceedings of the seventh international conference on information fusion,* Vol. 2, Stockholm, Sweden, June 28th-July 1st, 2004, pp. 739-746.

[63] G. Rogova and V. Nimier. Reliability in information fusion: Literature. Survey. *In Proceedings of 7th International Conference on Information. Fusion,* Sweden, pp. 1158-1165. 2004

[64] P. Kumar**,** A. Mittal and P. Kumar. A Multimodal Audio, Visible and Infrared Surveillance System (MAVISS)*. In Proceedings of the 3rd IEEE International Conference on Intelligent Sensing and Information Processing (ICISIP)*, pp. 151-157, 2005.

[65] J. Hershey, H. Attias, N. Jojic, and T. Krisjianson. Audio visual graphical models for speech processing. *In IEEE International Conference on Speech, Acoustics, and Signal Processing* (ICASSP04), Motreal, Canada, May 2004.

[66] A. V. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy. Dynamic bayesian networks for audio-visual speech recognition. *In EURASIP Journal on Applied Signal Processing* (JASP02), 2002.

[67] H. J. Nock, G. Iyengar, and C. Neti. Speaker localization using audio-visual synchrony: An

empirical study. *In International Conference on Image and Video Retrieval* (CIVR03), 2003. pages 488-499.

[68] D. Gatica-Perez, G. Lathoud, I. McCowan, J. Odobez, and D. Moore. Audio-visual speaker tracking with importance particle filter. *In IEEE International Conference on Image Processing* (ICIP03), 2003.

[69] N. Checka, K. W. Wilson, M. R. Siracusa, and T. Darrell. Multiple person and speaker activity tracking with a particle filter. *In International Conference on Acoustics Speech and Signal Processing* (ICASSP04), 2004.

[70] H. J. Nock, G. Iyengar, and C. Neti. Assessing face and speech consistency for monologue detection in video. *In ACM Multimedia*, 2002.

[71] P. K. Atrey, M. S. Kankanhalli and R. Jain. Information assimilation framework for event detection in multimedia surveillance systems. *Special Issue on "Multimedia Surveillance Systems" in Springer/ACM Multimedia Systems Journal*, September 2006

[72] D. Whitney, and J. Pisano, TASC, Inc., Reading, Massachusetts. *AutoAlert: Automated Acoustic Detection of Incidents.* December 26, 1995.

[73] M. Cowling, R. Sitte. Comparison of Techniques for Environmental Sound Recognition. *Pattern Recognition Letters,* Elsevier Science Inc., Vol. 24, Issues 15, pp. 2895-2907, Nov. 2003.

[74] Proximity Sensors. http://www.machinedesign.com

[75] Z. Zhu, W. Li and G. Wolberg. Integrating LDV Audio and IR Video for Remote Multimodal Surveillance. *IEEE Workshop on Object Tracking and Classification In and Beyond the Visible Spectrum*, in conjunction with IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, June 2005

[76] I. Ulrich and J. Borenstein, "The guidecane, applying mobile robot technologies to assist the visually impaired," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 31, no. 2, pp. 131–136, 2001.

[77] S. Shoval, I. Ulrich, and J. Borenstein, Computerized Obstacle Avoidance Systems for the Blind and Visually Impaired. Intelligent Systems and Technologies in Rehabilitation Engineering, CRC Press, 2000.

[78] Caimi, F.M. Bailey, B.C. Blatt, J.H. Undersea object detection and recognition: the use of spatially andtemporally varying coherent illumination. OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century ,pages 1474-1479 vol.3

[79] Heckman, P., and Hodgson, "Underwater Optical Range Gating", IEEE Journal of Quantum Electronics, QE-3, 11, Nov. I967

[80] A. F. Bobick, and J. W. Davis. The Recognition of Human Movement Using Temporal Templates. *PAMI*, Vol. 23, No. 3, 2001.

[81] K. Takahashi, S. Seki, H.Kojima, and R. Oka, "Recognition of dexterous manipulations from time varying images," in *Proc. IEEEWorkshop Motion of Non-Rigid and Articulated Objects*, Austin, TX, 1994, pp. 23–28.

[82] A. F. Bobick and A. D.Wilson. A state-based technique to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1325–1337, Dec. 1997.

[83] R. Duggad, U. B. Desai. A Tutorial on Hidden Markov Models. Technical Report No. SPANN-96.1. 1996. Indian Institute of Technology Bombay.

[84] L. Rabinier, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of IEEE*. 77 (2) (1989) 257-285.

[85] M. Leo, P. Spagnolo, T. D'Orazio, A. Distante. Human Activity Recognition in Archaeological Sites by Hidden Markov Models. *PCM (2)* , 1019-1026, 2004

[86] D. Mahajan, N. Kwatra, S. Jain, P. Kalra, S. Banerjee. A Framework for Activity Recognition and Detection of Unusual Activities. 15-21, *ICVGIP*, 2004, Kolkata, India.

[87] M. Yang and N. Ahuja. Extraction and classification of visual motion pattern recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 892–897.

[88] U. Meier, R. Stiefelhagen, J. Yang, and A.Waibel. Toward unrestricted lip reading. *Int. J. Pattern Recognit. Artificial Intell.*, vol. 14, no. 5, pp. 571–585, Aug 2000

[89] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Proc. IEEE International Conference on Computer Vision*, pages 1985–1988, Beijing, China, Oct. 15-21 2005.

[90] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted HMMs for unusual event detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, July 2005.

[91] P. Kumar, A. Mittal and P. Kumar. A Multi-modal Data Fusion Framework Using Transferable Belief Model and Kalman filter for Robust Tracking in Dynamic Environment. Communicated to *Signal, Video and Image processing Journal,* Springer Verlag publication.

Contextualizing Ontologies with OntoLight: A Pragmatic Approach

Marko Grobelnik, Janez Brank, Blaž Fortuna and Igor Mozetič
Department of Knowledge Technologies, Jožef Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia
E-mail: {marko.grobelnik, janez.brank, blaz.fortuna, igor.mozetic}@ijs.si
http://kt.ijs.si/

Keywords: lightweight ontologies, grounding, contexts, semantic web

Received: July 27, 2007

We present a pragmatic approach to using large-scale ontologies as contexts. The approach is based on a light-weight ontology model and grounding of the ontology concepts in textual documents. These assumptions allow for efficient implementation of the basic operations (classification, population and mappings between ontologies), and, as a consequence, exploitation of several large-scale ontologies as background, contextual knowledge. We demonstrate one possible scenario how contextual information can be exploited during semi-automatic ontology construction from text corpora

Povzetek: Članek opisuje pragmatičen pristop k uporabi velikih ontologij kot kontekstov.

1 Introduction

Ontologies represent isolated pieces of knowledge. By networking them, one can explore their interrelations. One form of networked ontologies are contextualized ontologies. In this case, one ontology represents a context of the other and its constituent ingredients (concepts and relations). So, for a given ontology, its ingredients can be interpreted in different contexts by selecting appropriate ontologies which represent appropriate contexts.

In this paper we describe OntoLight, a software suite which implements basic reasoning functionalities for contextualized ontologies. It is limited to light-weight ontologies which are grounded with appropriate text corpora. The representation and reasoning scales to the largest currently available ontologies, comprising up to one million concepts. In particular, OntoLight currently incorporates the following five ontologies: AgroVoc and ASFA (relevant for the Food and Agricultural Organization of the UN), EuroVoc (EU legislation), Cyc (common-sense knowledge) and DMoz (a WWW directory).

There are two basic reasoning mechanisms implemented in OntoLight. First, new textual instances without a known class can be classified into the selected ontology. Second, soft (probabilistic) mappings between a pair of selected ontologies can be computed, thus providing a contextual relationship between the ontologies.

We are using OntoLight as a basic building block for extensions to OntoGen [1], where contextual mappings are used to improve semi-automatic construction of light-weight ontologies from text corpora. The same mechanism of contextual reasoning will be used to extend OntoGen to support simultaneous, collaborative development of an ontology. Our soft mappings between grounded ontologies also complement methods for ontology alignment, where mappings are computed on the basis of common, background ontologies (as provided by Swoogle, for example) [2, 3].

The OntoLight software package presented in this paper consists of several executable modules and a data library of ontologies. The main functionality we cover is the contextualization of ontologies through generation of soft mappings between ontologies, thus enabling to view concepts of one ontology through the perspective of another one. The second goal was achieving scalability needed for large case studies – i.e. being able to deal with large ontologies such as AgroVoc and ASFA. To achieve this we constrained the representation to a light-weight ontology model which covers targeted functionality needed in the case studies. Finally, we took care of the software engineering aspects of the result – namely, the software package is built on top of an existing Text-Garden software library [4]. It is written in C++ with a proper API and accessible through several development platforms (Java, Python, Matlab, Mathematica, Prolog).

In the next section we first present the ontology model used in OntoLight. Next, in Section 3 we present the library of ontologies already incorporated in OntoLight – each ontology is presented through its main features. In Section 4, the software package is presented by describing each module separately and through possible integration of the modules which could be used in a pipeline. Finally, in Section 5, we show an integration of OntoLight with OntoGen, where light-weight ontologies are used as background, contextual knowledge which helps the users during the process of semi-automatic ontology construction from text corpora.

2 The ontology model

The ontology model used in OntoLight is a relatively simple model which covers most of the well known light-weight ontologies. The model we use is a subset of richer ontology formalisms (such as OWL) in the sense that richer ontologies could be imported but not all their expressiveness can be used. Informally, the light-weight ontology model is defined by:

- A list of languages used for lexical terms.
- A list of class-types used for representing different types of nodes in the ontology structure.
- A list of classes where each class can have several lexical representations in one or several languages. One class represents one node in the graph.
- A list of relation-types used to label relations (links) between classes in the ontology graph.
- A list of relations connecting classes in the ontology graph.
- Each ontology can have one or several grounding models. Each grounding model is a function which proposes zero, one or more classes for a given instance. This corresponds to a classification /categorization model in machine learning terminology.

The above model has a one-to-one mapping into C++ classes in the OntoLight module of the Text-Garden library [4].

3 Library of ontologies

To perform experiments on real data, we had to import several ontologies into the OntoLight framework. Since most of the larger real life ontologies are still in non-standard formats we needed to develop specialized filters for pre-processing the available data into the common ".OntoLight" format used by the rest of the OntoLight package. In the first version of the software we decided to prepare filters for importing five medium to large scale ontologies. They are all used on a daily basis in real life applications. They model different types of knowledge – from relatively specific ones (AgroVoc, ASFA), a general one with legal bias (EuroVoc) to generic ones for Web contents (DMoz) and common sense (Cyc).

3.1 AgroVoc

AgroVoc is a multilingual structured thesaurus of all subject fields in Agriculture, Forestry, Fisheries, Food security and related domains (e.g. Sustainable Development, Nutrition, etc). It consists of words or expressions (terms) in different languages and is organized in the thesaurus relationships (e.g. "broader", "narrower", and "related") used to identify or search resources. Its main role is to standardize the indexing process in order to make search simpler and more efficient, and to provide users with the most relevant resources.

The AgroVoc thesaurus was developed by the Food and Agriculture Organization of the United Nations (FAO) and the Commission of the European Communities, in the early 1980s. It is updated by FAO roughly every three months and users can see the specific changes on the AgroVoc website [5]. AgroVoc is available in the five official languages of FAO, which are English, French, Spanish, Chinese and Arabic. Additionally, it is also available in Czech, German, Japanese, Portuguese, Slovak and Thai. Other translations, such as Hindi, Hungarian, Italian and Korean are currently underway or being revised.

AgroVoc is downloadable in several formats – we used the MS Access package which includes several tables with all the data about the ontology. Specifically, AgroVoc includes 12 languages, 65 relation-types, and 47101 classes. AgroVoc classes were grounded with text abstracts from ASFA document corpus (see below) which are close to AgroVoc terms.

3.2 ASFA

ASFA (Aquatic Sciences and Fisheries Abstracts) is a thesaurus used for the Aquatic Sciences and Fisheries Information System (ASFIS), an international co-operative information system for the collection and dissemination of information covering the science, technology and management of marine, brackish water, and freshwater environments. It contains approximately 1 million bibliographic references to the world's aquatic science literature accessioned since 1971 (for some journals and/or subject areas the coverage precedes 1971). All references are machine readable.

ASFA is produced as a cooperative effort by the international network of ASFA partners [6] which consists of: United Nations Co-sponsoring Partners, National and International Partners, and the Publishing Partner. The objective is to disseminate bibliographic information to the relevant research community. A good description of several aspects of ASFA is available at [7].

In our case we extracted the ASFA thesaurus and abstracts by crawling the web search interface. The extracted data were all in the English language. The thesaurus structure included two types of classes (descriptor and non-descriptor), 5 link types, and 9882 classes. ASFA classes were grounded with text abstracts available within the records of the crawled data (over 360.000 abstracts).

3.3 EuroVoc

EuroVoc is a multilingual thesaurus covering the fields in which the European Communities are active – it provides a means of indexing the documents in the documentation systems of the European institutions and of their users. The European Parliament, the Office for Official Publications of the European Communities, the national and regional parliaments in Europe, some national government departments and European organizations are currently using this controlled vocabulary. The recent version EuroVoc 4.2 exists in 21 official languages of the European Union (Bulgarian, Spanish, Czech, Danish, German, Estonian, Greek, English, French, Italian, Latvian, Lithuanian, Hungarian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene, Finnish and Swedish), and one other language (Croatian). In addition to these versions, it has been translated by the Parliaments of several other countries: Albania, Russia and the Ukraine.

The data of the thesaurus are available from [8] where we extracted the thesaurus structure by crawling the HTML pages (since the officially proposed way of getting the data was non-functioning) while the multilingual part (without the structure) was downloadable from the web site as an MS Excel file. The extracted data is available in 21 languages, it has two types of nodes (descriptors and non-descriptors), 5 relation types, and 13416 nodes (out of which 6645 are descriptors). We grounded the EuroVoc classes with the documents from Acquis Communitarian, the corpus of European legislation indexed with EuroVoc descriptors.

3.4 Cyc

The Cyc [9] knowledge base (KB) is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The original form of representation is a formal language CycL. The KB consists of terms which constitute the vocabulary of CycL and assertions which relate those terms. These assertions include both simple ground facts and rules with variables.

The Cyc KB is available for researchers from the Cycorp company homepage [10] in two different forms – OpenCyc (vocabulary only) and ResearchCyc (full version). In our case, we are using the data retrieved directly from the company under the ResearchCyc license. Since the Cyc KB is very rich (it includes ~50.000 first order logic rules) we decided to deal only with the static part of the KB. It is written only in English, it has two types of classes (concepts and lexical nodes), it has 3295 relations, and 464.988 concepts.

Since Cyc has only structure (concepts and facts) we grounded each Cyc's concept by querying Google with lexical representation for that class.

3.5 DMoz / Open directory project

The Open Directory Project (ODP), also known as DMoz, is the largest multilingual open content directory of World Wide Web links that is constructed and maintained by a community of volunteer editors. The browsing and search service is accessible from [11].

The directory data (structure and content) are available from [12] in the RDF format. The version we are using here uses only the English part of the directory, it has 3 types of relations, and 642.995 concepts.

The taxonomic part was grounded with the content which is available within the downloadable data. The main data source for grounding were short textual descriptions of the manually categorized web sites within each DMoz category.

4 Software modules

In the following subsections we present each of the OntoLight modules (or module groups) dealing with ontology data – from raw data to classification models and mappings. The software is available from [13].

4.1 Ontology data transformation utilities

The function of the ontology data transformation utilities is to process specific formats of each of the selected ontologies for the ontology library. The result of all the utilities is saving the ontology data in the unifying binary format with the file-extension ".OntoLight" and its textual counterpart with the file extension ".OntoLight.Txt". As described in section 3, the ontology library consists of five ontologies – therefore we prepared five command line utilities for processing the data:

- AgroVoc2OntoLight.Exe
- Asfa2OntoLight.Exe
- Cyc2OntoLight.Exe
- DMoz2OntoLight.Exe
- EuroVoc2OntoLight.Exe

Each of the utilities takes on the input file name or file path to the data and produces the binary file (".OntoLight") and the textual file (".OntoLight.Txt"). An example run of the transformation of the EuroVoc is the following:

```
[d:\textgarden\eurovoc2ontolight]
EuroVoc2OntoLight.exe
EuroVoc To Ontology-Light [Feb 12 2007]
=====================================
Input-EuroVoc-FilePath (-i:)=f:/data/EuroVoc/
Output-OntoLight-FileName
      (-o:)=f:/Data/OntoLight/EuroVoc.OntoLight
Output-Text-FileName
  (-ot:)=f:/Data/OntoLight/EuroVoc.OntoLight.Txt
========================
Loading 'f:/data/EuroVoc/listMultiLg_All.txt'
... 6645/6646
Done. (6645)
Loading 'f:/data/EuroVoc/eurovoc.txt' ... Done.
(48044)
Saving OntoLight to
'f:/Data/OntoLight/EuroVoc.OntoLight' ... Done.
Saving Text to
'f:/Data/OntoLight/EuroVoc.OntoLight.Txt' ...
Done.
```

4.2 Ontology grounding module

The ontology grounding module OntoLight2OntoCfier.exe creates from an ontology stored in the ".OntoLight" format an additional file with the extension ".OntoCfier" (and its textual representation ".OntoCfier.Txt"). This file includes a classification model which is used by the OntoClassify module (next subsection) for classification of new instances in the ontology classes. The current version uses a centroid-based classifier which calculates a centroid vector for each class in the ontology. It takes into account the data used for grounding and the hierarchical part of the ontology structure. The actual classification is performed with the *k*-NN (*k*-nearest-neighbour) algorithm [14].

Here is an example run of the OntoLight2OntoCfier.exe module for ontology grounding. On the input the utility takes ".OntoLight" data and a pre-processed bag-of-words file with the text documents and the descriptors from the ontology. On the output the system creates ".OntoCfier" file with a classifier and its textual representation (".OntoCfier.Txt"). With additional parameters we specify the language we are using for grounding (in the case when data exists in several languages), to see whether the document's category equals descriptors in the ontology and the threshold for writing weighted words in the textual output.

```
[d:\textgarden\ontolight2ontocfier]OntoLight2Ont
oCfier.exe
Ontology-Light To Ontology-Classifier
[Feb 12 2007]
================================================
Input-OntoLight-FileName
     (-iol:)=f:/Data/OntoLight/EuroVoc.OntoLight
Input-BagOfWords-FileName
     (-ibow:)=f:/Data/OntoLight/Acquis.Bow
Output-OntoClassifier-FileName
     (-oom:)=f:/Data/OntoLight/EuroVoc.OntoCfier
Output-OntoClassifier-Text-FileName
     (-oom:)=f:/Data/OntoLight/EuroVoc.OntoCf
Language-Name (-lang:)=EN
DocumentCategory-Is-TermId (-catisid:)=Yes
Cut-Word-Weight-Sum-Percent (-cwwprc:)=0.33
================================================
Loading Onto-Light from
'f:/Data/OntoLight/EuroVoc.OntoLight' ... Done.
Loading Bag-Of-Words from
'f:/Data/OntoLight/Acquis.Bow' ... Done.
Generating Ontology-Classifier...
  Creating BowDocWgtBs ... Done.
  Collecting documents per ontology-term ...
    Docs:7972/7972 Pos:26915 Neg:149
  Done.
  Creating sub-terms & up-terms vectors ...
Done.
  Creating centroids ...
    Active-Terms:1399
    Active-Terms:441
    Active-Terms:85
    Active-Terms:7
    Active-Terms:0
    Active-Terms:0
  Done.
Done.
Saving Onto-Classifier to
'f:/Data/OntoLight/EuroVoc.OntoCfier' ... Done.
Saving Text to
'f:/Data/OntoLight/EuroVoc.OntoCfier.Txt' ...
Done.
```

4.3 Ontology population module

The ontology population module OntoClassify.Exe takes as input a grounded ontology in the ".OntoCfier" format and instance data (in various textual formats) and produces XML and textual file with the possible categories for the given instance.

In the following example we take a grounded version of the EuroVoc and the query "Slovenia and Croatia are having a fishing industry". The result is in the files OntoCfy.Xml and OntoCfy.Txt.

```
[d:\textgarden\ontoclassify]OntoClassify.exe
Ontology-Classify [Feb 12 2007]
==============================
Input-OntoClassifier-FileName
     (-ioc:)=f:/Data/OntoLight/EuroVoc.OntoCfier
Input-Query-String (-qs:)=Slovenia and Croatia
are having a fishing industry.
Input-Query-Html-File (-qh:)=
Input-Query-CompactDocument-FileName (-qcpd:)=
Input-Query-Url (-qu:)=
Input-Query-URL-Vector-FileName (-quf:)=
Output-Classification-Xml-File
                            (-ox:)=OntoCfy.Xml
Output-Classification-Txt-File
                            (-ot:)=OntoCfy.Txt
=================
Loading Onto-Classifier from
'f:/Data/OntoLight/EuroVoc.OntoCfier' ... Done.
```

The resulting textual file lists classes from the EuroVoc grounded ontology to which the query should belong with the highest confidence. Each line of the file OntoCfy.Txt includes the following three fields: rank, confidence, and class name:

```
 1. 0.201 Croatia
 2. 0.171 fisheries policy
 3. 0.162 Slovenia
 4. 0.161 fishing area
 5. 0.159 national independence
 6. 0.159 fishing regulations
 7. 0.156 fishery management
 8. 0.147 fisheries structure
 9. 0.147 fishing fleet
10. 0.144 Community fisheries
```

4.4 Ontology mapping module

The last module in the pipeline of utilities is the utility OntoJoint.exe which takes as an input two grounded ontologies in the ".OntoCfier" format and creates soft mappings between the classes of both ontologies. This is done in the following way: first, by aligning vocabularies of grounded ontologies (this typically means aligning words from respective bag-of-words representations), and second, by classifying centroid vectors from the first ontology into the classes of the second one.

In the following example we take as an input the EuroVoc and ASFA ontologies and store mapping results into XML and textual files, OntoJoint.XML and OntoJoint.Txt, respectively:

```
[d:\textgarden\ontojoint]OntoJoint.exe
Join-Ontologies [Mar 12 2007]
=============================
Input-OntoClassifier-FileName-1
    (-ioc1:)=f:/Data/OntoLight/EuroVoc.OntoCfier
Input-OntoClassifier-FileName-2
      (-ioc2:)=f:/Data/OntoLight/Asfa.OntoCfier
Output-OntologyJoin-Xml-File
                           (-ox:)=OntoJoint.Xml
Output-OntologyJoin-Txt-File
                           (-ot:)=OntoJoint.Txt
===============
Loading Onto-Classifier-1 from
'f:/Data/OntoLight/EuroVoc.OntoCfier' ... Done.
Loading Onto-Classifier-2 from
'f:/Data/OntoLight/Asfa.OntoCfier' ... Done.
```

The following is an example mapping from the resulting OntoJoint.Txt file where we see a mapping from the

ASFA "fishing licence" class to 10 related classes from the EuroVoc ontology.

```
'fishing licence' ->
    1. 'Legal aspects' (0.003)
    2. 'Ships' (0.003)
    3. 'Disputes' (0.002)
    4. 'Ecology' (0.002)
    5. 'Military operations' (0.001)
    6. 'Rare species' (0.001)
    7. 'Public health' (0.001)
    8. 'Fish culture' (0.001)
    9. 'Commercial fishing' (0.001)
   10. 'Resource development' (0.001)
```

5 Contextualized ontology generation with OntoGen

OntoGen [1] is a software tool for semi-automatic, data-driven ontology construction. It incorporates methods for discovering concepts from a collection of documents. Documents are represented by the well known bag-of-words representation, where each document is encoded as a vector of term frequencies. The similarity of a pair of documents is calculated by the number and weights of the words that these documents share. The weights of the words are usually calculated by the so-called TFIDF weighting, but there are other alternatives.

OntoGen implements two methods for concept discovery: Latent Semantic Indexing (LSI) [15] and *k*-means clustering [16]. LSI is a method for linear dimensionality reduction by learning an optimal sub-basis which approximates documents' bag-of-words vectors. The sub-basis vectors are proposed as concepts. The *k*-means method discovers concepts by clustering the documents' bag-of-words vectors into *k* clusters where each cluster is then proposed as a concept.

We have extended OntoGen with OntoLight, specifically with five general-purpose light-weight ontologies: AgroVoc, ASFA, EuroVoc, DMoz and Cyc. These ontologies provide contexts to the user during the user-guided, data-driven generation of an ontology from a corpus of documents. OntoGen structures the documents into concepts and subconcepts, but, until now, has used only extracted keywords to suggest concept names. With contextual ontologies available, OntoGen is now able to provide much better suggestions for concept names based on the similarity between structured documents and grounded concepts from the selected contexts. As a consequence, the user can view each concept suggested by OntoGen through different "sematic lenses": each view corresponds to a different context as implemented by a different light-weight ontology. Figure 1 gives an example.

6 Conclusion

In the paper we describe OntoLight, a set of software modules for:

- transforming raw ontology data for several ontologies from their specific formats into a unifying light-weight ontology format,
- grounding the ontology and storing it into grounded ontology format,
- populating grounded ontologies with new instance data, and
- creating mappings between grounded ontologies.

As a part of OntoLight we already prepared the ontology library consisting of five different ontologies: AgroVoc, ASFA, Cyc, DMoz, and EuroVoc. Additional ontologies (e.g., WordNet) will be incorporated in the future.

We will be using OntoLight as a basic building block for extensions to OntoGen, where contextual mappings are used to improve semi-automatic construction of light-weight ontologies from text corpora. The same mechanism of contextual reasoning will be used to extend OntoGen to support simultaneous, collaborative development of an ontology. Our soft mappings between grounded ontologies also complement methods for ontology alignment, where mappings are computed on the basis of common, background ontologies. We plan to integrate our approach to mappings with the mechanisms for ontology alignments.

Acknowledgement

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under NeOn (IST-2004-27595-IP) and PASCAL (IST-2002-506778).

References

[1] B. Fortuna, M. Grobelnik, D. Mladenic: Semi-automatic Construction of Topic Ontology. Semantics, Web and Mining, Joint International Workshop, EWMF 2005 and KDO 2005, Porto, Portugal, October 3–7, 2005.

[2] M. Sabou, M. d'Aquin, and E. Motta: Using the Semantic Web as Background Knowledge for Ontology Mapping, In Proceedings of the International Workshop on Ontology Matching (OM-2006), collocated with ISWC-06.

[3] M. Sabou, M. d'Aquin, W. R. van Hage and E. Motta: Improving Ontology Matching by Dynamically Exploring Online Knowledge. Submitted for review, 2007.

[4] TextGarden: http://kt.ijs.si/Dunja/textgarden/

[5] AgroVoc: http://www.fao.org/aims/ag_intro.htm

[6] ASFA: http://www.fao.org/fi/asfa/partners.asp

[7] ASFA: http://www.fao.org/fi/website/-FIRetrieveAction.do?dom=org&-xml=asfa_prog.xml&xp_nav=2

[8] EuroVoc: http://europa.eu/eurovoc/

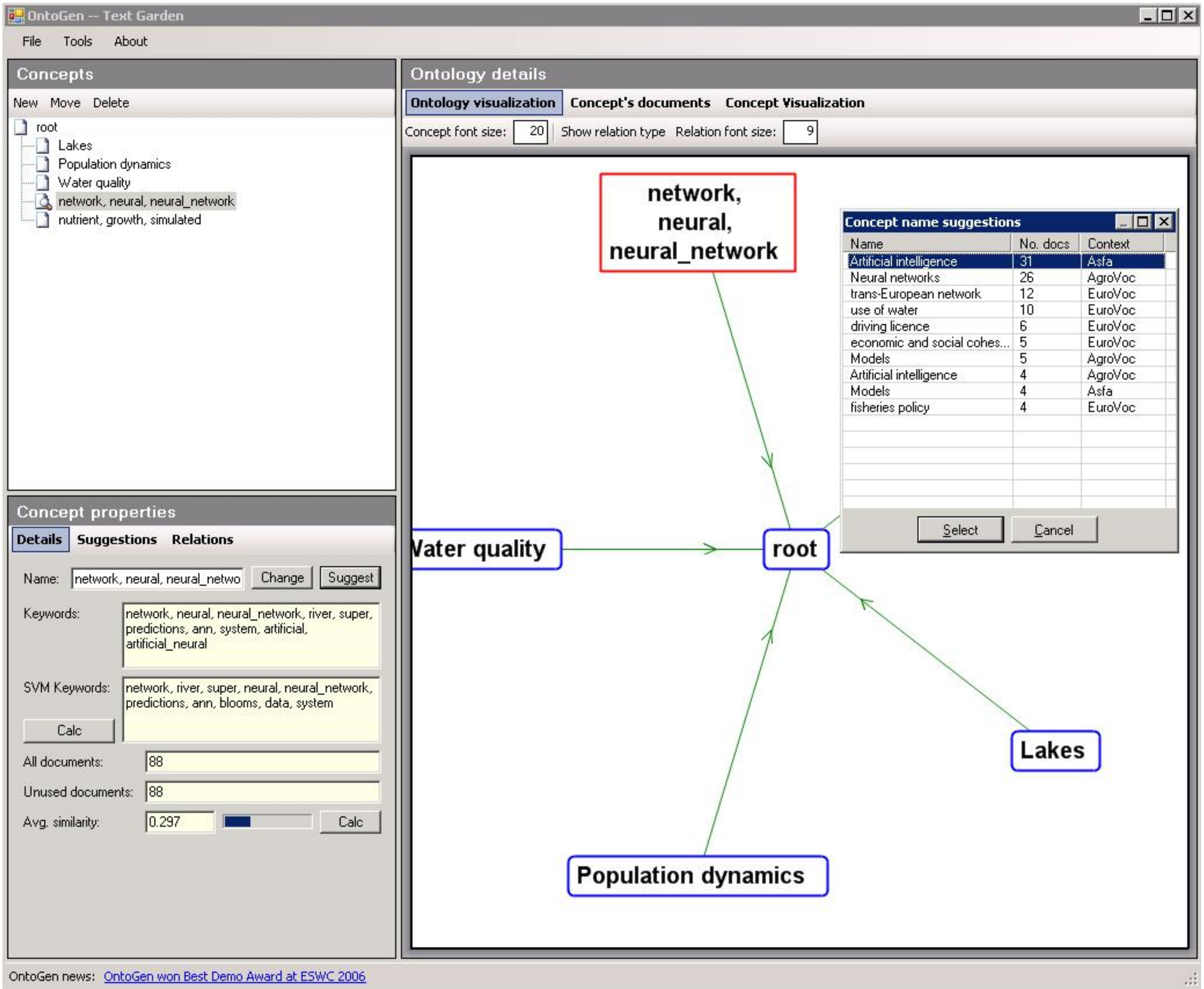
[9] Douglas B. Lenat. Cyc: A Large-Scale Investment in Knowledge Infrastructure. Comm. ACM 38, no. 11, November 1995.

[10] Cyc: http://www.cyc.com/

[11] DMoz: http://dmoz.org/

[12] DMoz: http://rdf.dmoz.org/

[13] OntoLight: http://analytics.ijs.si/Projects/NEON/-OntoLight.Zip



Name: network, neural, neural_netwo

Keywords: network, neural, neural_network, river, super, predictions, ann, system, artificial, artificial_neural

SVM Keywords: network, river, super, neural, neural_network, predictions, ann, blooms, data, system

All documents: 88

Unused documents: 88

Avg. similarity: 0.297

Concept font size: 20

Relation font size: 9

Name	No. docs	Context
Artificial intelligence	31	Asfa
Neural networks	26	AgroVoc
trans-European network	12	EuroVoc
use of water	10	EuroVoc
driving licence	6	EuroVoc
economic and social cohes...	5	EuroVoc
Models	5	AgroVoc
Artificial intelligence	4	AgroVoc
Models	4	Asfa
fisheries policy	4	EuroVoc

OntoGen news: OntoGen won Best Demo Award at ESWC 2006

Figure 1: *A screenshot of OntoGen when used to structure the abstracts of recent issues of the Ecological Modelling journal. Contexts are provided by three ontologies: AgroVoc, ASFA, and EuroVoc. Some concept names were already derived from contextual suggestions (Water quality, Population dynamics, Lakes) and the user inspects current suggestions for the top node (network, neural, neural_network). The system provides two sensible suggestions: Artificial Intelligence (from ASFA) and Neural networks (from AgroVoc), while the third suggestion: trans-European network (from EuroVoc) probably makes less sense.*

[14] Shakhnarovish, Darrell, Indyk (Eds.): *Nearest-Neighbor Methods in Learning and Vision*, The MIT Press, 2005.

[15] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman: Indexing by Latent Semantic Analysis, *Journal of the American Society of Information Science*, vol. 41, no. 6, 391-407, 1990.

[16] Jain, Murty, Flynn: Data Clustering: A Review, *ACM Comp. Surveys*, 1999.

Augmented Marked Graphs

King-Sing Cheung
University of Hong Kong
Pokfulam, Hong Kong
E-mail: ks.cheung@hku.hk

Keywords: augmented marked graph, Petri net, liveness, boundedness, reversibility, conservativeness

Received: August 27, 2007

Augmented marked graphs possess some structural characteristics desirable for modelling shared resource systems such as manufacturing systems. However, there are only a few known properties on augmented marked graphs, and these known properties are mainly on liveness and reversibility. In this paper, the properties of augmented marked graphs are reviewed extensively. Siphon-based and cycle-based characterisations for liveness and reversibility as well as transformation-based characterisations for boundedness and conservativeness are proposed. Pretty simple conditions and procedures are then derived for checking the liveness, reversibility, boundedness and conservativeness of augmented marked graphs. The dining philosopher problem is used for illustration.

Povzetek: Opisane so lastnosti grafov za predstavitev sistemov z deljenimi viri.

1 Introduction

Augmented marked graphs were first introduced by Chu and Xie [1]. They are not well known as compared to other sub-classes of Petri nets such as free-choice nets [2], and the properties of augmented marked graphs are not studied extensively. However, augmented marked graphs possess a structure which is desirable for modelling shared resources, and for this reason, they are often used in modelling shared resource systems, such as manufacturing systems [1, 3, 4, 5, 6, 7].

In the literature, the studies of augmented marked graphs mainly focus on deadlock-freeness, liveness and reversibility. Based on mathematical programming, Chu and Xie proposed a necessary and sufficient condition of live and reversible augmented marked graphs, which checks the existence of potential deadlocks [1]. However, this involves analysis on the flow of tokens during execution and the checking cannot be simply made by looking into the structure. Chu and Xie also proposed a siphon-based characterisation for live and reversible augmented marked graphs but it provides a sufficient condition only. The boundedness and conservativeness of augmented marked graphs were not investigated.

There are other studies of augmented marked graphs, which are mainly on the property-preserving synthesis or composition of augmented marked graphs. Jeng proposed a synthesis method of process nets for manufacturing system design [4, 5]. (Note : Process nets broadly cover augmented marked graphs.) Based on siphons and the firability of transitions, sufficient conditions for liveness and reversibility are derived. Huang also investigated the composition of augmented marked graphs via common resource places, so that some essential properties such as liveness, boundedness and reversibility can be preserved under certain conditions [6].

In our previous works on augmented marked graphs, we proposed new characterisations for live and reversible augmented marked graphs as well as the synthesis of augmented marked graphs for system design [7, 8, 9, 10, 11]. This paper extends our previous works with a focus on the properties of augmented marked graphs. It reports the following two contributions.

First, we propose a number of characterisations for live and reversible augmented marked graphs, based on siphons and cycles. In particular, a new property called R-inclusion property is introduced to characterise the siphon-trap property of augmented marked graphs. With this property, a pretty simple necessary and sufficient condition for live and reversible augmented marked graphs is then proposed. Second, for analysis of the boundedness and conservativeness of augmented marked graphs, a R-transform is introduced to transform an augmented marked graph into marked graphs. With the R-transform, a pretty simple necessary and sufficient condition for bounded and conservative augmented marked graphs is proposed. These characterisations will be illustrated using the dining philosopher problem.

The rest of this paper is organised as follows. Following this introduction, Section 2 provides the preliminaries to be used in this paper. Section 3 briefly introduces augmented marked graphs. Section 4 focus on liveness and reversibility of augmented marked graphs, where siphon-based and cycle-based characterisations are proposed. Section 5 then focus on boundedness and conservativeness of augmented marked graphs, where transformation-based characterisations are proposed. Section 6 illustrates these characterisations using the dining philosopher example. Finally, Section 7 concludes our results.

It should be noted that, in this paper, proofs of the proposed properties are shown in the appendix.

2 Preliminary

This section provides the preliminaries to be used in this paper for those readers who are not familiar with Petri nets [12, 13, 14, 15].

A place-transition net (PT-net) is a directed graph consisting of two sorts of nodes called places and transitions, such that no arcs connect two nodes of the same sort. Graphically, a place is denoted by a circle, a transition by a box, and an arc by a directed line. A Petri net is a PT-net with tokens assigned to its places, and the token distribution is denoted by a marking.

A Petri net is usually used to represent a discrete system, where the places denote conditions, the transitions denote events and the arcs between places and transitions denote the relationship between conditions and events.

Definition 1. A place-transition net (PT-net) is a 4-tuple $N = \langle P, T, F, W \rangle$, where P is a set of places, T is a set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation and $W : F \rightarrow \{ 1, 2, ... \}$ is a weight function. N is said to be ordinary if and only if the range of W is $\{ 1 \}$.

An ordinary PT-net is usually written as $\langle P, T, F \rangle$. In the rest of this paper, unless specified otherwise, all PT-nets refer to ordinary PT-nets.

Definition 2. Let $N = \langle P, T, F, W \rangle$ be a PT-net. For any $x \in (P \cup T)$, ${}^\bullet x = \{ y \mid (y, x) \in F \}$ and $x^\bullet = \{ y \mid (x, y) \in F \}$ are called the pre-set and post-set of x, respectively.

For clarity in presentation, the pre-set and post-set of a set of places or transitions $X = \{ x_1, x_2, ..., x_n \}$ can be written as ${}^\bullet X$ and $X^\bullet$ respectively, where ${}^\bullet X = {}^\bullet x_1 \cup {}^\bullet x_2 \cup ... \cup {}^\bullet x_n$ and $X^\bullet = x_1^\bullet \cup x_2^\bullet \cup ... \cup x_n^\bullet$.

Definition 3. For a PT-net $N = \langle P, T, F, W \rangle$, a path is a sequence of places and transitions $\rho = \langle x_1, x_2, ..., x_n \rangle$, such that $(x_i, x_{i+1}) \in F$ for i = 1, 2, ..., n-1. ρ is said to be elementary if and only if it contains no duplicate places or transitions.

Definition 4. For a PT-net $N = \langle P, T, F, W \rangle$, a sequence of places $\langle p_1, p_2, ..., p_n \rangle$ is called a cycle if and only if there exists a set of transitions $\{ t_1, t_2, ..., t_n \}$, such that $\langle p_1, t_1, p_2, t_2, ..., p_n, t_n \rangle$ forms an elementary path and $(t_n, p_1) \in F$.

Definition 5. For a PT-net $N = \langle P, T, F, W \rangle$, a marking is a function $M : P \rightarrow \{ 0, 1, 2, ...\}$, where M(p) is the number of tokens in p. (N, M_0) represents N with an initial marking M_0.

Semantically, a marking represents the state of a Petri net. The initial marking specifically represents the initial state of a Petri net. A transition is enabled and can be fired at a state (marking) where all the places in its pre-set hold tokens. On firing the transition, tokens will be moved from the places in its pre-set to the places in its post-set. The firing of a transition is formally defined as follows.

Definition 6. For a PT-net (N, M_0), a transition t is said to be enabled at a marking M if and only if $\forall\ p \in {}^\bullet t : M(p) \geq W(p,t)$. On firing t, M is changed to M' such that $\forall\ p \in P : M'(p) = M(p) - W(p,t) + W(t,p)$. In notation, $M\ [N,t\rangle\ M'$ or $M\ [t\rangle\ M'$.

Definition 7. For a PT-net (N, M_0), a sequence of transitions $\sigma = \langle t_1, t_2, ..., t_n \rangle$ is called a firing sequence if and only if $M_0\ [t_1\rangle\ ...\ [t_n\rangle\ M_n$. In notation, $M_0\ [N,\sigma\rangle\ M_n$ or $M_0\ [\sigma\rangle\ M_n$.

Definition 8. For a PT-net (N, M_0), a marking M is said to be reachable if and only if there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. In notation, $M_0\ [N,*\rangle\ M$ or $M_0\ [*\rangle\ M$. $[N, M_0\rangle$ or $[M_0\rangle$ represents the set of all reachable markings of (N, M_0).

The structure of a PT-net can be represented by a matrix called incidence matrix.

Definition 9. Let $N = \langle P, T, F, W \rangle$ be a PT-net, where $P = \{ p_1, p_2, ..., p_m \}$ and $T = \{ t_1, t_2, ..., t_n \}$. The incidence matrix of N is an m × n matrix V whose typical entry $v_{ij} = W(p_i,t_j) - W(t_j,p_i)$ represents the change in number of tokens in p_i after firing t_j once, for i = 1, 2, ..., m and j = 1, 2, ..., n.

Liveness, boundedness, safeness, reversibility and conservativeness are best known properties of Petri nets. Liveness implies freeness of deadlocks. Boundedness and safeness imply freeness of capacity overflow. Reversibility refers to the capability of being reinitialised from any reachable states. Conservativeness is a special form of boundedness.

Definition 10. For a PT-net (N, M_0), a transition t is said to be live if and only if $\forall\ M \in [M_0\rangle, \exists\ M' : M\ [*\rangle\ M'\ [t\rangle$. (N, M_0) is said to be live if and only if every transition is live.

Definition 11. For a PT-net (N, M_0), a place p is said to be k-bounded if and only if $\forall\ M \in [M_0\rangle : M(p) \leq k$, where k is a positive integer. (N, M_0) is said to be bounded if and only if every place is k-bounded, and safe if and only if every place is 1-bounded.

Definition 12. A PT-net (N, M_0) is said to be reversible if and only if $\forall\ M \in [M_0\rangle : M\ [*\rangle\ M_0$.

Definition 13. For a PT-net $N = \langle P, T, F, W \rangle$, a place invariant is a |P|-vector $\alpha \geq 0$ such that $\alpha V = 0$, where V is the incidence matrix of N.

Definition 14. A PT-net is said to be conservative if and only if there exists a place invariant $\alpha > 0$.

Figure 1 shows an ordinary PT-net which is live, bounded, safe, reversible and conservative.

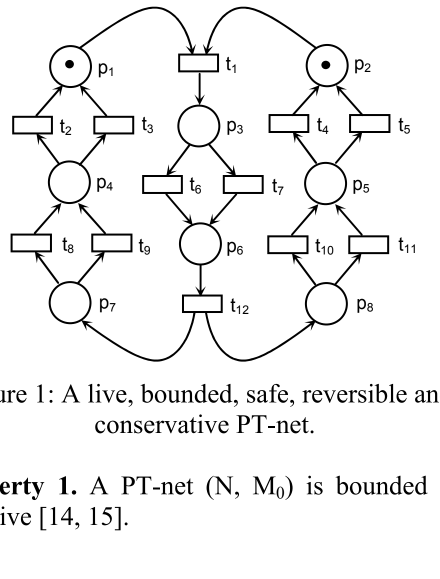


Figure 1: A live, bounded, safe, reversible and conservative PT-net.

Property 1. A PT-net (N, M_0) is bounded if it is conservative [14, 15].

Definition 15. For a PT-net N, a set of places S is called a siphon if and only if $^\bullet S \subseteq S^\bullet$. S is said to be minimal if and only if there does not exist another siphon S' in N such that $S' \subset S$.

Definition 16. For a PT-net, a set of places T is called a trap if and only if $T^\bullet \subseteq {}^\bullet T$.

Definition 17. A PT-net (N, M_0) is said to satisfy the siphon-trap property if and only if every siphon contains a marked trap (or every minimal siphon contains a marked trap).

A well known sub-class of Petri nets, marked graphs possess many special properties pertaining to its liveness, boundedness and reversibility.

Definition 18. A marked graph is an ordinary PT-net $N = \langle P, T, F, W \rangle$ such that $\forall p \in P : |^\bullet p| = |p^\bullet| = 1$.

Property 2. A marked graph (N, M_0) is live if and only if every cycle is marked by M_0 [13, 14].

Property 3. A live marked graph (N, M_0) is bounded if and only if every place belongs to a cycle marked by M_0 [13, 14].

Property 4. A live and bounded marked graph is reversible [13, 14].

Property 5. For a marked graph, the corresponding place vector of a cycle is a place invariant [13, 14].

Figure 2 shows a marked graph which is live, bounded, safe and reversible. Places $\langle p_1, p_3, p_6, p_7, p_4 \rangle$ form a cycle. The place vector is a place invariant.

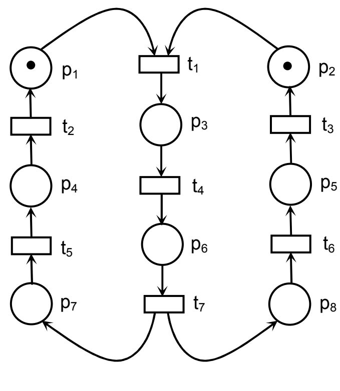


Figure 2: A live, bounded, safe and reversible marked graph.

3 Augmented marked graphs

Augmented marked graphs were first introduced by Chu and Xie [1]. This section briefly describes augmented marked graphs.

Definition 19. An augmented marked graph (N, M_0; R) is a PT-net (N, M_0) with a specific subset of places R, such that : (a) Every place in R is marked by M_0. (b) The net (N', M_0') obtained from (N, M_0; R) by removing the places in R and their associated arcs is a marked graph. (c) For each $r \in R$, there exist $k_r \geq 1$ pairs of transitions $D_r = \{ \langle t_{s1}, t_{h1} \rangle, \langle t_{s2}, t_{h2} \rangle, ..., \langle t_{skr}, t_{hkr} \rangle \}$, such that $r^\bullet = \{ t_{s1}, t_{s2}, ..., t_{skr} \} \subseteq T$ and $^\bullet r = \{ t_{h1}, t_{h2}, ..., t_{hkr} \} \subseteq T$ and that, for each $\langle t_{si}, t_{hi} \rangle \in D_r$, there exists in (N', M_0') an elementary path ρ_{ri} connecting t_{si} to t_{hi}. (d) In (N', M_0'), every cycle is marked and no ρ_{ri} is marked.

Figure 3 shows a typical augmented marked graph (N, M_0; R), where R = { r_1, r_2 }. For r_1, $D_{r1} = \{ \langle t_1, t_{11} \rangle, \langle t_3, t_9 \rangle \}$. For r_2, $D_{r2} = \{ \langle t_2, t_{11} \rangle, \langle t_4, t_{10} \rangle \}$.

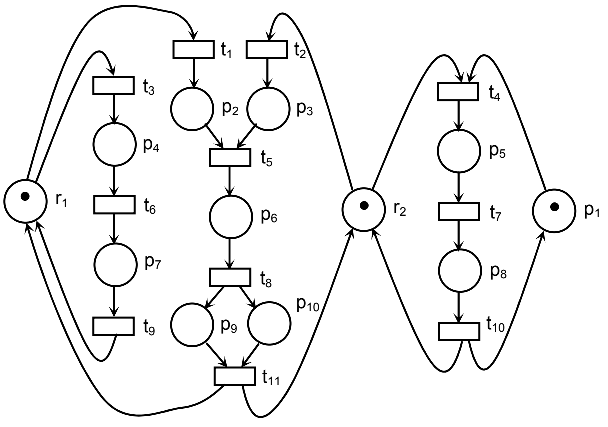


Figure 3: An augmented marked graph.

Augmented marked graphs possesses a number of special properties pertaining to liveness, boundedness, reversibility and conservativeness. In the following sections, these properties are thoroughly investigated.

4 Liveness and reversibility

This section focus on the liveness and reversibility of augmented marked graphs. After reporting several known properties, some siphon-based and cycle-based characterisations for live and reversible augmented marked graphs are proposed.

Property 6. An augmented marked graph is live if and only if it does not contain any potential deadlock [1]. (Note : A potential deadlock is a siphon which would eventually become empty.)

Property 7. An augmented marked graph is reversible if it is live [1].

Property 8. An augmented marked graph is live and reversible if and only if every minimal siphon would never become empty.

Property 9. An augmented marked graph (N, M_0; R) is live and reversible if every minimal siphon, which contains at least one place of R, contains a marked trap [1].

For the augmented marked graph (N, M_0; R) shown in Figure 3, the minimal siphons are : { p_1, p_5, p_8 }, { r_1, p_2, p_4, p_6, p_7, p_9 }, { r_1, p_2, p_4, p_6, p_7, p_{10} }, { r_2, p_3, p_5, p_6, p_8, p_9 } and { r_2, p_3, p_5, p_6, p_8, p_{10} }. Each of these minimal siphons contains a marked trap, and would never become empty. (N, M_0; R) is live and reversible.

The places and transitions generated by cycles are defined as follows.

Definition 20. For a PT-net N, Ω_N is defined as the set of all cycles in N.

Definition 21. Let N be a PT-net. For a subset of cycles $Y \subseteq \Omega_N$, P[Y] is defined as the set of places in Y, and $T[Y] = {}^\bullet P[Y] \cap P[Y]^\bullet$ is defined as the set of transitions generated by Y.

For clarity in presentation, P[{γ}] and T[{γ}] can be written as P[γ] and T[γ], to denote the set of places in a cycle γ and the set of transitions generated by γ, respectively.

Definition 22. For a PT-net N, an elementary path ρ = ⟨ x_1, x_2, ..., x_n ⟩ is said to be conflict-free if and only if, for any transition x_i in ρ, $j \neq (i-1) \Rightarrow x_j \notin {}^\bullet x_i$.

Property 10. Let S be a minimal siphon of a PT-net. For any p, p' ∈ S, there exists in S a conflict-free path from p to p' [16].

Property 11. For a minimal siphon S of an augmented marked graph (N, M_0; R), there exists a set of cycles $Y \subseteq \Omega_N$ such that P[Y] = S.

Property 12. Every cycle in an augmented marked graph is marked.

Property 13. Every siphon in an augmented marked graph is marked.

Property 14. Let (N, M_0; R) be an augmented marked graph. For every r ∈ R, there exists a minimal siphon which contains only one marked place r.

Consider the augmented marked graph (N, M_0; R) shown in Figure 3. Every minimal siphon is covered by cycles. Consider a minimal siphon S_1 = { r_1, p_2, p_4, p_6, p_7, p_9 }. There exists a set of cycles Y_1 = { γ_{11}, γ_{12} }, where γ_{11} = ⟨ r_1, p_4, p_7 ⟩ and γ_{12} = ⟨ r_1, p_2, p_6, p_9 ⟩, such that S_1 = P[Y_1]. Consider another minimal siphon S_2 = { r_2, p_3, p_5, p_6, p_8, p_{10} }. There exists a set of cycles Y_2 = { γ_{21}, γ_{22} }, where γ_{21} = ⟨ r_2, p_5, p_8 ⟩ and γ_{22} = ⟨ r_2, p_3, p_6, p_{10} ⟩, such that S_2 = P[Y_2]. For S_1, r_1 ∈ R is the only one marked place. Also, for S_2, r_2 ∈ R is the only one marked place.

For an augmented marked graph, minimal siphons can be classified into R-siphons and NR-siphons. Based on R-siphons and NR-siphons, some characterisations for augmented marked graphs are proposed.

Definition 23. For an augmented marked graph (N, M_0; R), a minimal siphon is called a R-siphon if and only if it contains at least one place in R.

Definition 24. For an augmented marked graph (N, M_0; R), a minimal siphon is called a NR-siphon if and only if it does not contain any place in R.

Definition 25. Let N be a PT-net. For a set of places Q in N, Ω_N[Q] is defined as the set of cycles that contains at least one place in Q.

For clarity in presentation, Ω_N[{p}] can be written as Ω_N[p] to denote the set of cycles that contains a place p.

Property 15. For an augmented marked graph (N, M_0; R), a R-siphon is covered by a set of cycles $Y \subseteq \Omega_N[R]$.

Figure 4 shows another augmented marked graph (N, M_0; R), where R = { r_1, r_2 }. There are five minimal siphons, namely, S_1 = { r_1, p_3, p_4, p_7, p_8 }, S_2 = { r_1, p_3, p_5, p_7, p_8 }, S_3 = { r_2, p_2, p_4, p_6, p_8, p_9, p_{10} }, S_4 = { r_2, p_2, p_5, p_6, p_8, p_9, p_{10} } and S_5 = { p_1, p_3, p_7 }. S_1, S_2, S_3 and S_4 are R-siphon as they contain at least one place in R. S_5 is a NR-siphon which does not contain any place in R. For (N, M_0; R), every R-siphon is covered by a set of cycles in Ω_N[R]. For example, S_1 = { r_1, p_3, p_4, p_7, p_8 } is covered by a set of cycles Y_1 = { γ_{11}, γ_{12} } ⊆ Ω_N[R], where γ_{11} = ⟨ r_1, p_3, p_7 ⟩ and γ_{12} = ⟨ r_1, p_4, p_8 ⟩.

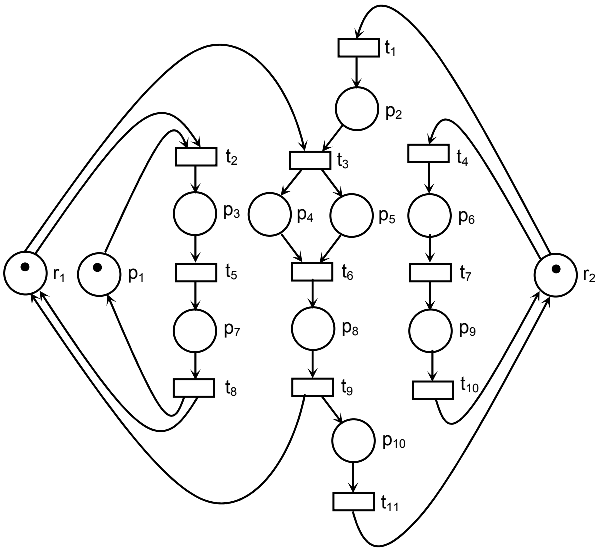


Figure 4: Another augmented marked graph.

Property 16. Let S be a R-siphon of an augmented marked graph (N, M_0; R). For every $t \in (S^\bullet \setminus {}^\bullet S)$, there does not exist any $s \in (S \setminus R)$ such that $t \in s^\bullet$.

Property 17. For an augmented marked graph (N, M_0; R), a NR-siphon contains itself as a marked trap and would never become empty.

Property 18. An augmented marked graph (N, M_0; R) is live and reversible if and only if no R-siphons eventually become empty.

Property 19. An augmented marked graph (N, M_0; R) satisfies the siphon-trap property if and only if every R-siphon contains a marked trap.

Consider the augmented marked graph (N, M_0; R) shown in Figure 4. Every R-siphon contains a marked trap. Each of the R-siphons S_1 = { r_1, p_3, p_4, p_7, p_8 }, S_2 = { r_1, p_3, p_5, p_7, p_8 }, S_3 = { r_2, p_2, p_4, p_6, p_8, p_9, p_{10} } and S_4 = { r_2, p_2, p_5, p_6, p_8, p_9, p_{10} } contains a marked trap and would never become empty. (N, M_0; R) is live and reversible.

Property 20. (characterisation of Property 9) An augmented marked graph (N, M_0; R) is live and reversible if every R-siphon contains a marked trap.

Property 18 provides a simple necessary and sufficient condition for live and reversible augmented marked graphs. With Properties 18 and 20, we can determine if an augmented marked graph is live and reversible based on R-siphons. Besides, Property 15 provides a characterisation for R-siphons so that R-siphons can be identified by finding cycles in Ω_N[R]. We may now derive a strategy for checking the liveness and reversibility of an augmented marked graph (N, M_0; R) :

(a) Find all R-siphons based on Ω_N[R].

(b) Check if every R-siphon contains a marked trap. If yes, report (N, M_0; R) is live and reversible. Otherwise, go to (c).

(c) For each R-siphon which does not contain any marked trap, check if it would never become empty. If yes, report (N, M_0; R) is live and reversible. Otherwise, report (N, M_0; R) is neither live nor reversible.

In the following, conflict-free cycles are introduced. Based on conflict-free cycles, a new property called R-inclusion is proposed. It is then used for characterising liveness and reversibility of augmented marked graphs.

Definition 26. For a PT-net N, a set of cycles $Y \subseteq \Omega_N$ is said to be conflict-free if and only if, for any q, q' $\in P[Y]$, there exists in $P[Y]$ a conflict-free path from q to q'.

Figure 5 shows a PT-net N. Consider three cycles $\gamma_1, \gamma_2, \gamma_3 \in \Omega_N[p_3]$, where $\gamma_1 = \langle p_3, p_2, p_7 \rangle$, $\gamma_2 = \langle p_3, p_4 \rangle$ and $\gamma_3 = \langle p_3, p_1, p_6, p_{10}, p_8 \rangle$. The set of cycles $Y_1 = \{ \gamma_1, \gamma_2 \}$ is conflict-free because for any q, q' $\in P[Y_1]$, there exists in $P[Y_1]$ a conflict-free path from q to q'. The set of cycles $Y_2 = \{ \gamma_2, \gamma_3 \}$ is not conflict-free. We have $p_4, p_8 \in P[Y_2]$. p_4 is connected to p_8 via only one path $\rho = \langle p_4, t_5, p_3, t_1, p_1, t_3, p_6, t_6, p_{10}, t_9, p_8 \rangle$ in Y_2, and ρ is not conflict-free because $p_4, p_8 \in {}^\bullet t_5$.

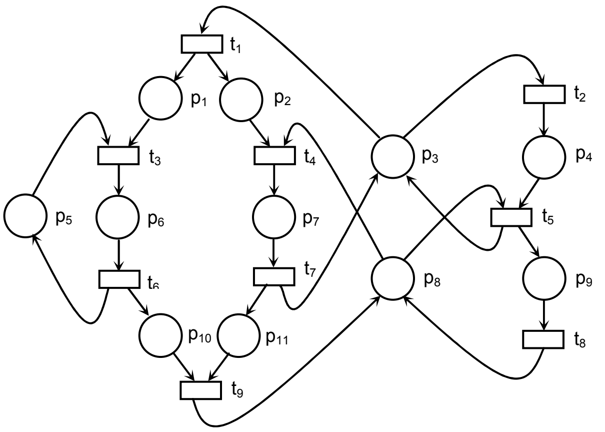


Figure 5: A PT-net for illustration of conflict-free cycles.

Property 21. Let S be a minimal siphon of an augmented marked graph $(N, M_0; R)$, and $Y \subseteq \Omega_N$ be a set of cycles such that $S = P[Y]$. Then, Y is conflict-free.

For the augmented marked graph shown in Figure 3, $\{ r_1, p_2, p_4, p_6, p_7, p_9 \}$ is a minimal siphon covered by a set of cycles $\{ \langle r_1, p_4, p_7 \rangle, \langle r_1, p_2, p_6, p_9 \rangle \}$ which is conflict free. $\{ r_2, p_3, p_5, p_6, p_8, p_{10} \}$ is another minimal siphon covered by a set of cycles $\{ \langle r_2, p_5, p_8 \rangle, \langle r_2, p_3, p_6, p_{10} \rangle \}$ which is conflict-free. For the augmented marked graph shown in Figure 4, $\{ r_1, p_3, p_4, p_7, p_8 \}$ is a minimal siphon covered by a set of cycles $\{ \langle r_1, p_3, p_7 \rangle, \langle r_1, p_4, p_8 \rangle \}$ which is conflict free. $\{ r_1, p_3, p_5, p_7, p_8 \}$ is another minimal siphon covered by a set of cycles $\{ \langle r_1, p_3, p_7 \rangle, \langle r_1, p_5, p_8 \rangle \}$ which is conflict free.

Definition 27. Let $(N, M_0; R)$ be an augmented marked graph. A place $r \in R$ is said to satisfy the R-inclusion if and only if, for any set of cycles $Y \subseteq \Omega_N[R]$ such that Y is conflict-free, ${}^\bullet r \subseteq T[Y] \Rightarrow r^\bullet \subseteq T[Y]$.

Figure 6 shows an augmented marked graph $(N, M_0; R)$, where $R = \{ r_1, r_2 \}$. Consider r_1. For any set of cycles $Y_1 \subseteq \Omega_N[R]$ such that Y_1 is conflict-free, ${}^\bullet r_1 \subseteq T[Y_1] \Rightarrow r_1^\bullet \subseteq T[Y_1]$. Next, consider r_2. For any set of cycles $Y_2 \subseteq \Omega_N[R]$ such that Y_2 is conflict-free, ${}^\bullet r_2 \subseteq T[Y_2] \Rightarrow r_2^\bullet \subseteq T[Y]$. Both r_1 and r_2 satisfy the R-inclusion.

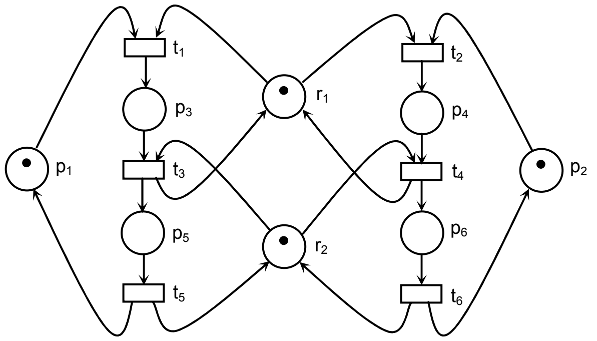


Figure 6: An augmented marked graph for illustration of R-inclusion.

Figure 7 shows another augmented marked graph $(N, M_0; R)$. For $r_1 \in R$, there exists a set of cycles $Y_1 = \{ \gamma_{11}, \gamma_{12} \} \subseteq \Omega_N[R]$, where $\gamma_{11} = \langle r_1, p_5 \rangle$ and $\gamma_{12} = \langle r_1, p_5, r_2, p_6 \rangle$. ${}^\bullet r_1 = \{ t_5, t_6 \} \subseteq T[Y_1] = \{ t_3, t_4, t_5, t_6 \}$ but $r_1^\bullet = \{ t_2, t_3 \} \not\subseteq T[Y_1]$. For $r_2 \in R$, there exists a set of cycles $Y_2 = \{ \gamma_{21}, \gamma_{22} \} \subseteq \Omega_N[R]$, where $\gamma_{21} = \langle r_2, p_6 \rangle$ and $\gamma_{22} = \langle r_2, p_6, r_1, p_5 \rangle$. ${}^\bullet r_2 = \{ t_5, t_6 \} \subseteq T[Y_2] = \{ t_3, t_4, t_5, t_6 \}$ but $r_2^\bullet = \{ t_1, t_4 \} \not\subseteq T[Y_2]$. Both r_1 and r_2 do not satisfy the R-inclusion property.

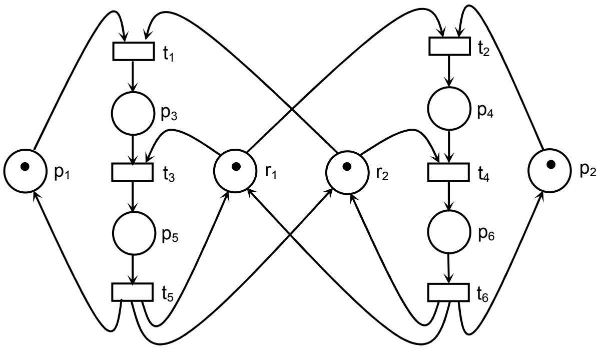


Figure 7: Another augmented marked graph for illustration of R-inclusion.

Property 22. For an augmented marked graph $(N, M_0; R)$, a R-siphon S contains itself as a marked trap if every place $r \in R$ in S satisfies the R-inclusion property.

Property 23. An augmented marked graph $(N, M_0; R)$ satisfies the siphon-trap property if and only if every place $r \in R$ satisfies the R-inclusion property.

Property 24. An augmented marked graph $(N, M_0; R)$ is live and reversible if every place $r \in R$ satisfies the R-inclusion property.

Consider the augmented marked graph $(N, M_0; R)$ shown in Figure 6. We have $R = \{ r_1, r_2 \}$, where both r_1 and r_2 satisfy the R-inclusion property. Any R-siphon, such as $\{ r_1, p_3, p_4 \}$ or $\{ r_2, p_5, p_6 \}$, contains itself as a marked trap. $(N, M_0; R)$ satisfies the siphon-trap property, and is live and reversible.

Property 24 provides a cycle-based condition for live and reversible augmented marked graphs. We need to check the R-inclusion property which involves finding cycles and checking their pre-sets and post-sets.

Based on Properties 15, 18, 20, 22 and 24, we revise the strategy for checking the liveness and reversibility of an augmented marked graph (N, M_0; R) with the use of the R-inclusion property, as follows.

(a) Check if every r ∈ R satisfies the R-inclusion property. If yes, report (N, M_0; R) is live and reversible. Otherwise go to (b).

(b) Let R' ⊆ R be the set of places which do not satisfy the R-inclusion property. Based on $\Omega_N[R']$, find all R-siphons which contain at least one place in R'.

(c) For each R-siphon identified in (b), check if it contains a marked trap. If yes, report (N, M_0; R) is live and reversible. Otherwise, go to (d).

(d) For each R-siphon identified in (b) that does not contain any marked trap, check if it would never become empty. If yes, report (N, M_0; R) is live and reversible. Otherwise, report (N, M_0; R) is neither live nor reversible.

5 Boundedness and conservativeness

This section focus on the boundedness and conservativeness of augmented marked graphs, which are less studied in the literature. Some transform-based characterisations for bounded and conservative augmented marked graphs are proposed.

In the following, we introduce a new transformation called R-transform for augmented marked graphs. It simply transforms an augmented marked graphs (N, M_0; R) into a number of marked graphs by replacing each place in R by a set of places.

Property 25. Let (N, M_0; R) be an augmented marked graph to be transformed into (N', M_0') as follows. For each place r ∈ R, where $D_r = \{ \langle t_{s1}, t_{h1}\rangle, \langle t_{s2}, t_{h2}\rangle, ..., \langle t_{skr}, t_{hkr}\rangle \}$, replace r with a set of places $\{ p_1, p_2, ..., p_{kr} \}$, such that $M_0'[p_i] = M_0[r]$ and $p_i^\bullet = \{ t_{si} \}$ and $^\bullet p_i = \{ t_{hi} \}$ for i = 1, 2, ..., k_r. Then, (N', M_0') is a marked graph.

Definition 28. Let (N, M_0; R) be an augmented marked graph. The marked graph (N', M_0') transformed from (N, M_0; R) as stated in Property 25 is called the R-transform of (N, M_0; R).

Property 26. The R-transform of an augmented marked graph is live.

Figure 8 shows an augmented marked graph. Figure 9 shows its R-transform, where r is replaced by { q_1, q_2 }.

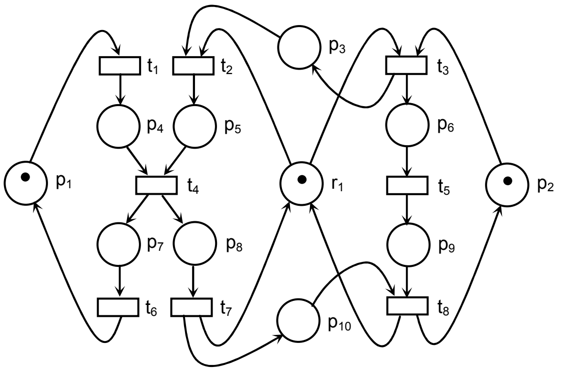


Figure 8: An augmented marked graph for illustration of R-transform.

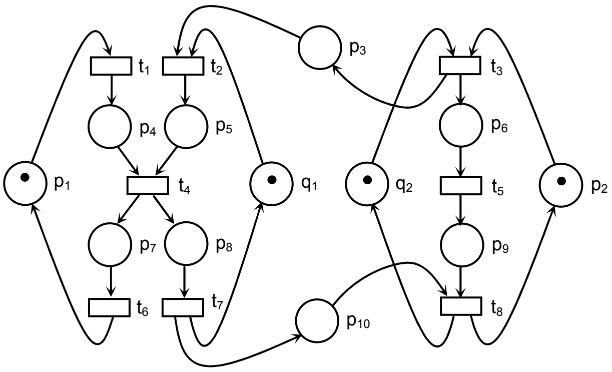


Figure 9: The R-transform of the augmented marked graph shown in figure 8.

Property 27. Let (N', M_0') be the R-transform of an augmented marked graph (N, M_0; R), where r ∈ R is replaced by a set of places Q = { q_1, q_2, ..., q_k }, and P_0 be the set of marked places in N'. Then, for each q_i in N', there exists a place invariant α_i of N' such that $\alpha_i[q_i] = 1$ and $\alpha_i[s] = 0$ for any place s ∈ $P_0 \setminus \{q_i\}$.

Property 28. Let (N, M_0; R) be an augmented marked graph, where R = { r_1, r_2, ..., r_n }. Let (N', M_0') be the R-transform of (N, M_0; R), where each r_i is replaced by a set of places Q_i, for i = 1, 2, ..., n. If every place in (N', M_0') belongs to a cycle, then there exists a place invariant α of N' such that α > 0 and $\alpha[q_1] = \alpha[q_2] = ... = \alpha[q_k]$ for each Q_i = { q_1, q_2, ..., q_k }.

Consider the R-transform (N', M_0') shown in Figure 9. It is a live marked graph. For q_1, there exists a place invariant α_1, such that $\alpha_1[q_1] = 1$ and $\alpha_1[q_2] = \alpha_1[p_1] = \alpha_1[p_2] = 0$. For q_2, there also exists a place invariant α_2, such that $\alpha_2[q_2] = 1$ and $\alpha_2[q_1] = \alpha_2[p_1] = \alpha_2[p_2] = 0$. In (N', M_0'), every place belongs to a cycle. There also exists a place invariant α > 0, where $\alpha[q_1] = \alpha[q_2]$.

Based on R-transform, a necessary and sufficient condition for bounded and conservative augmented marked graphs is proposed.

Property 29. Let (N', M_0') be the R-transform of an augmented marked graph (N, M_0; R). (N, M_0; R) is bounded and conservative if and only if every place in (N', M_0') belongs to a cycle.

With Properties 29, we derive the following strategy for checking the boundedness and conservativeness of an augmented marked graph (N, M_0; R) :

(a) Create the R-transform (N', M_0') of (N, M_0; R).

(b) For each place p in (N', M_0'), check if there exists a cycle that contains p. If yes, report (N, M_0; R) is bounded and conservative. Otherwise, report (N, M_0; R) is neither bounded nor conservative.

Property 30. Let (N', M_0') be the R-transform of an augmented marked graph (N, M_0; R). (N, M_0; R) is bounded and conservative if and only if (N', M_0') is bounded.

Consider the augmented marked graph (N, M_0; R) shown in Figure 8, and the R-transform (N', M_0') of (N, M_0; R) in Figure 9. Every place in (N', M_0') belongs to a cycle. (N, M_0; R) is bounded and conservative. (N', M_0') is also bounded and conservative.

Figure 10 shows an augmented marked graph (N, M_0; R), and Figure 11 shows the R-transform (N', M_0') of (N, M_0; R). For (N', M_0'), p_3 does not belong to any cycle. Also, p_8 does not belong to any cycle. (N, M_0; R) is neither bounded nor conservative.

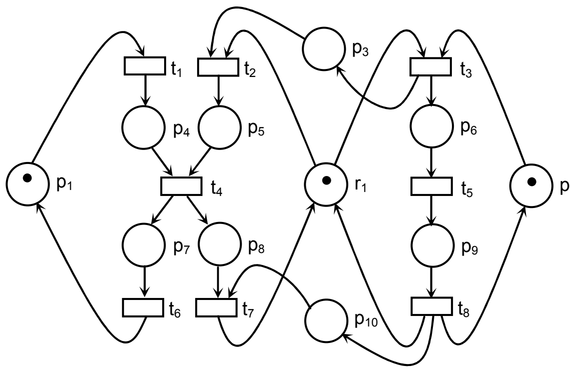


Figure 10: Another augmented marked graph for illustration of R-transform.

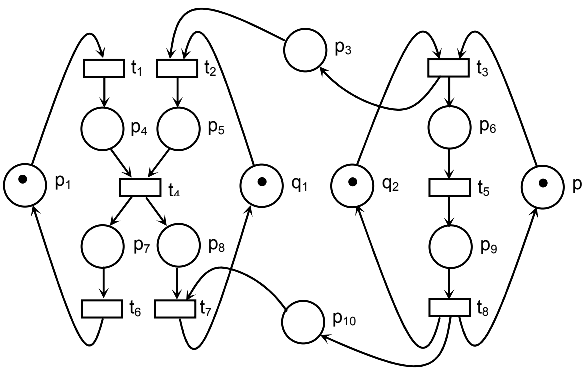


Figure 11: The R-transform of the augmented marked graph shown in figure 10.

6 The dining philosopher problem

This section illustrates the properties of augmented marked graphs obtained in the previous sections using the dining philosopher problem.

The dining philosopher problem (version 1) :

Six philosophers (H_1, H_2, H_3, H_4, H_5 and H_6) are sitting around a circular table for dinner. They are either meditating or eating the food placed at the centre of the table. There are six pieces of chopsticks (C_1, C_2, C_3, C_4, C_5 and C_6) shared by them for getting the food to eat, as shown in Figure 12. For one to get the food to eat, both the chopstick at the right hand side and the chopstick at the left hand side must be available. The philosopher then grasps both chopsticks simultaneously and then takes the food to eat. Afterwards, the chopsticks are released and returned to their original positions simultaneously.

Figure 13 shows the augmented marked graph (N, M_0; R) which represents the dining philosopher problem (version 1).

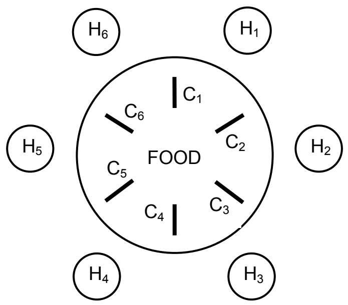


Figure 12: The dinning philosopher problem.

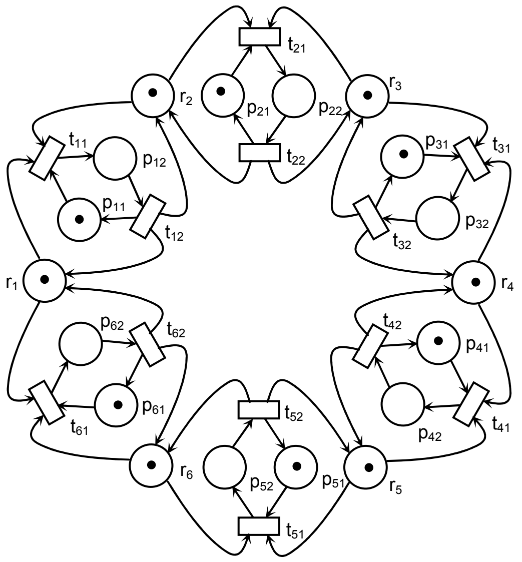


Figure 13 : The dining philosopher problem (version 1).

<u>Semantic meaning for places and transitions</u>

p_{11}	H_1 is meditating.
p_{12}	H_1 has got C_1 and C_2 and takes the food.
p_{21}	H_2 is meditating.
p_{22}	H_2 has got C_2 and C_3 and takes the food.
p_{31}	H_3 is meditating.
p_{32}	H_3 has got C_3 and C_4 and takes the food.
p_{41}	H_4 is meditating.
p_{42}	H_4 has got C_4 and C_5 and takes the food.
p_{51}	H_5 is meditating.
p_{52}	H_5 has got C_5 and C_6 and takes the food.
p_{61}	H_6 is meditating.
p_{62}	H_6 has got C_6 and C_1 and takes the food.
r_1	C_1 is available for pick.
r_2	C_2 is available for pick.
r_3	C_3 is available for pick.
r_4	C_4 is available for pick.
r_5	C_5 is available for pick.
r_6	C_6 is available for pick.
t_{11}	H_1 takes the action to grasp C_1 and C_2.
t_{12}	H_1 takes the action to return C_1 and C_2.
t_{21}	H_1 takes the action to grasp C_2 and C_3.
t_{22}	H_1 takes the action to return C_2 and C_3.
t_{31}	H_1 takes the action to grasp C_3 and C_4.
t_{32}	H_1 takes the action to return C_3 and C_4.
t_{41}	H_1 takes the action to grasp C_4 and C_5.
t_{42}	H_1 takes the action to return C_4 and C_5.
t_{51}	H_1 takes the action to grasp C_5 and C_6.
t_{52}	H_1 takes the action to return C_5 and C_6.
t_{61}	H_1 takes the action to grasp C_6 and C_1.
t_{62}	H_1 takes the action to return C_6 and C_1.

For (N, M_0; R), every R-siphons contains a marked trap and would never become empty. Every place in its R-transform belongs to a cycle. Based on the results obtained in Sections 4 and 5, (N, M_0; R) is live, bounded, reversible and conservative.

The dining philosopher problem (version 2) :

The Dining Philosopher Problem is revised. For one to get the food to eat, he or she first grasps the chopstick at the right hand side if available, then grasps the chopstick at the left hand side if available, and then takes the food to eat. Afterwards, the chopsticks are released and returned to their original positions simultaneously.

Figure 14 shows the augmented marked graph (N, M_0; R) which represents the dining philosopher problem (version 2). The set of places $\{r_1, p_{13}, r_2, p_{23}, r_3, p_{33}, r_4, p_{43}, r_5, p_{53}, r_6, p_{63}\}$ is a R-siphon which would become empty after firing the sequence of transitions $\langle t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}\rangle$. Deadlock would occur after firing $\langle t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}\rangle$. Based on the results obtained in Section 4, (N, M_0; R) is neither live nor reversible. On the other hand, for the R-transform of (N, M_0; R), every place belongs to a cycle. Based on the results obtained in Section 5, (N, M_0; R) is bounded and conservative.

7 Conclusion

In the past decade, augmented marked graphs have evolved into a sub-class of Petri nets for modelling shared resource systems. One major reason is that augmented marked graphs possess a structure which is desirable for modelling shared resources. However, the properties of augmented marked graphs are not extensively studied.

In this paper, a number of new characterisations for live and reversible augmented marked graphs are proposed. In particulars, some of these characterisations are based on cycles, instead of siphons. Besides, a R-transform is introduced. Based on the R-transform, a number of new characterisations for bounded and conservative augmented marked graphs are proposed. Consolidating these results, pretty simple conditions and procedures for checking the liveness, reversibility, boundedness and conservativeness of augmented marked graphs are derived. These have been illustrated using the dining philosopher problem.

Augmented marked graphs are often used for modelling shared-resource systems wherein the system analyst need to achieve the system design objectives on two folds. On one hand, the resources are scarce and should be maximally shared. On the other hand, the system should be carefully designed so that erroneous situations, such as deadlock and capacity overflow, due to sharing of resources should be avoided. For a shared-resource system modelled as an augmented marked graph, essential properties such as liveness, reversibility, boundedness and conservativeness can be effectively analysed with the new characterisations for augmented marked graphs. These contribute to ensuring the design correctness of shared resource systems.

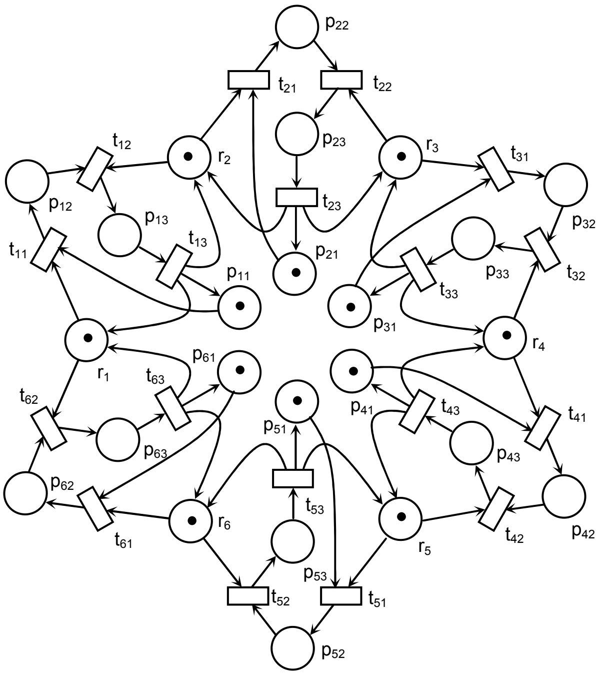


Figure 14 : The dining philosopher problem (version 2).

Semantic meaning for places and transitions

p_{11}	H_1 is meditating.
p_{12}	H_1 has got C_1 and prepares to pick C_2.
p_{13}	H_1 has got C_1 and C_2 and takes the food.
p_{21}	H_2 is meditating.
p_{22}	H_2 has got C_2 and prepares to pick C_3.
p_{23}	H_2 has got C_2 and C_3 and takes the food.
p_{31}	H_3 is meditating.
p_{32}	H_3 has got C_3 and prepares to pick C_4.
p_{33}	H_3 has got C_3 and C_4 and takes the food.
p_{41}	H_4 is meditating.
p_{42}	H_4 has got C_4 and prepares to pick C_5.
p_{43}	H_4 has got C_4 and C_5 and takes the food.
p_{51}	H_5 is meditating.
p_{52}	H_5 has got C_5 and prepares to pick C_6.
p_{53}	H_5 has got C_5 and C_6 and takes the food.
p_{61}	H_6 is meditating.
p_{62}	H_6 has got C_6 and prepares to pick C_1.
p_{63}	H_6 has got C_6 and C_1 and takes the food.
r_1	C_1 is available for pick.
r_2	C_2 is available for pick.
r_3	C_3 is available for pick.
r_4	C_4 is available for pick.
r_5	C_5 is available for pick.
r_6	C_6 is available for pick.
t_{11}	H_1 takes the action to grasp C_1.
t_{12}	H_1 takes the action to grasp C_2.
t_{13}	H_1 takes the action to return C_1 and C_2.
t_{21}	H_2 takes the action to grasp C_2.
t_{22}	H_2 takes the action to grasp C_3.
t_{23}	H_2 takes the action to return C_2 and C_3.
t_{31}	H_3 takes the action to grasp C_3.
t_{32}	H_3 takes the action to grasp C_4.
t_{33}	H_3 takes the action to return C_3 and C_4.
t_{41}	H_4 takes the action to grasp C_4.
t_{42}	H_4 takes the action to grasp C_5.
t_{43}	H_4 takes the action to return C_4 and C_5.
t_{51}	H_5 takes the action to grasp C_5.
t_{52}	H_5 takes the action to grasp C_6.
t_{53}	H_5 takes the action to return C_5 and C_6.
t_{61}	H_6 takes the action to grasp C_6.
t_{62}	H_6 takes the action to grasp C_1.
t_{63}	H_6 takes the action to return C_6 and C_1.

References

[1] Chu, F. and Xie, X. (1997), Deadlock Analysis of Petri Nets Using Siphons and Mathematical Programming, *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 6, pp. 793-804.

[2] Desel, J. and Esparza, J. (1995), *Free Choice Petri Nets*, Cambridge University Press.

[3] Zhou, M.C. and Venkatesh, K. (1999), *Modeling, Simulation and Control of Flexible Manufacturing Systems : A Petri Net Approach*, World Scientific.

[4] Jeng, M.D. et al. (2000), Manufacturing Modeling Using Process Nets with Resources, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2185-2190, IEEE Press.

[5] Jeng, M.D. et al. (2002), Process Nets with Resources for Manufacturing Modeling and their Analysis, *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, pp. 875-889.

[6] Huang, H.J. et al. (2003), Property-Preserving Composition of Augmented Marked Graphs that Share Common Resources, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1446-1451, IEEE Press.

[7] Cheung, K.S. and Chow, K.O. (2005), Manufacturing System Design Using Augmented Marked Graph, *Proceedings of the Chinese Control Conference*, pp. 1209-1213, SCUT Press.

[8] Cheung, K.S. (2004), New Characterisations for Live and Reversible Augmented Marked Graphs, *Information Processing Letters*, Vol. 92, No. 5, pp. 239-243.

[9] Cheung, K.S. and Chow, K.O. (2005), Cycle-Inclusion Property of Augmented Marked Graphs, *Information Processing Letters*, Vol. 94, No. 6, pp. 271-276.

[10] Cheung, K.S. and Chow, K.O. (2005), A Synthesis Approach to Deriving Object-Based Specifications from Object Interaction Scenarios, In : Nilsson, A.G. et al. (Eds.), *Advances in Information Systems Development*, pp. 647-656, Springer.

[11] Cheung, K.S., et al. (2005), A Petri-Net-Based Synthesis Methodology for Use-Case-Driven System Design, *Journal of Systems and Software*, Vol. 79, No. 6, pp. 772-790.

[12] Peterson, J.L. (1981), *Petri net theory and the modeling of systems*, Prentice Hall.

[13] Reisig, W. (1985), *Petri Nets : An Introduction*, Springer-Verlag.

[14] Murata, T. (1989), Petri Nets : Properties, Analysis and Applications, *Proceedings of the IEEE*, Vol. 77, No. 4., pp. 541-580.

[15] Desel, J. and Reisig, W. (1998), Place Transition Petri Nets, *Lectures on Petri Nets I : Basic Models*, Lecture Notes in Computer Science, Vol. 1491, pp. 122-173, Springer-Verlag.

[16] Barkaoui, K. et al. (1995), On Liveness in Extended Non Self-Controlling Nets, *Application and Theory of Petri Nets*, Lecture Notes in Computer Science, Vol. 935, pp. 25-44, Springer-Verlag.

Appendix

For clarify in presentation, proofs of the proposed properties for augmented marked graphs are shown in this appendix as follows.

Proof of Property 8. ($\Leftarrow$) For an augmented marked graph, if every minimal siphon would never become empty, every siphon which contains at least one minimal siphon would never become empty. It follows from Properties 6 and 7 that the augmented marked graph is live and reversible. ($\Rightarrow$) It follows from Property 6 that every siphon (and hence, every minimal siphon) would never become empty.

Proof of Property 11. Let $S = \{ p_1, p_2, ..., p_n \}$. For each p_i, by definition of augmented marked graphs that ${}^\bullet p_i \neq \emptyset$. Then, there exists $p_j \in S$, where $p_j \neq p_i$, such that $(p_j{}^\bullet \cap {}^\bullet p_i) \neq \emptyset$. Since S is a minimal siphon, according to Property 10, p_i connects to p_j via a conflict-free path in S. Since p_j connects to p_i, this forms a cycle γ_i in S, where $p_i \in P[\gamma_i] \subseteq S$. Let $Y = \{ \gamma_1, \gamma_2, ..., \gamma_n \}$. We have $P[Y] = P[\gamma_1] \cup P[\gamma_2] \cup ... \cup P[\gamma_n] \subseteq S$. On the other hand, $S \subseteq (P[\gamma_1] \cup P[\gamma_2] \cup ... \cup P[\gamma_n]) = P[Y]$ because $S = \{ p_1, p_2, ..., p_n \}$. Hence, $P[Y] = S$.

Proof of Property 12. (by contradiction) Let $(N, M_0; R)$ be an augmented marked graph. Suppose there exists a cycle γ in $(N, M_0; R)$, such that γ is not marked. γ does not contain any place in R, and also exists in the net (N', M_0') obtained from $(N, M_0; R)$ after removing the places in R and their associated arcs. However, by definition of augmented marked graphs, γ is marked.

Proof of Property 13. For an augmented marked graph, according to Properties 11 and 12, every minimal siphon contains cycles and is marked. Hence, every siphon, which contains at least one minimal siphon, is marked.

Proof of Property 14. Let $D_r = \{ \langle t_{s1}, t_{h1} \rangle, \langle t_{s2}, t_{h2} \rangle, ..., \langle t_{sn}, t_{hn} \rangle \}$, where $r^\bullet = \{ t_{s1}, t_{s2}, ..., t_{sn} \}$ and ${}^\bullet r = \{ t_{h1}, t_{h2}, ..., t_{hn} \}$. For each $\langle t_{si}, t_{hi} \rangle \in D_r$, t_{si} connects to t_{hi} via an elementary path ρ_i which is not marked. Let $S = P_1 \cup P_2 \cup ... \cup P_n \cup \{ r \}$, where P_i is the set of places in ρ_i. We have ${}^\bullet P_i \subseteq (P_i{}^\bullet \cup r^\bullet)$ because, for each $p \in P_i$, $| {}^\bullet p | = | p^\bullet | = 1$. Then, $({}^\bullet P_1 \cup {}^\bullet P_2 \cup ... \cup {}^\bullet P_n) \subseteq (P_1{}^\bullet \cup P_2{}^\bullet \cup ... \cup P_n{}^\bullet \cup r^\bullet)$. Besides, ${}^\bullet r = \{ t_{h1}, t_{h2}, ..., t_{hn} \} \subseteq (P_1{}^\bullet \cup P_2{}^\bullet \cup ... \cup P_n{}^\bullet)$. Hence, ${}^\bullet S = ({}^\bullet P_1 \cup {}^\bullet P_2 \cup ... \cup {}^\bullet P_n \cup {}^\bullet r) \subseteq (P_1{}^\bullet \cup P_2{}^\bullet \cup ... \cup P_n{}^\bullet \cup r^\bullet) = S^\bullet$. Therefore, S is a siphon in which r is the only one marked place. Let S' be a minimal siphon in S. According to Property 13, S' is marked. Since r is the only one marked place in S, r is also the only one marked place in S'.

Proof of Property 15. (By contradiction) Let S be a R-siphon. According to Property 11, S is covered by cycles. Suppose there exists a cycle γ in S, such that $\gamma \notin \Omega_N[R]$. By definition of augmented marked graphs, for any $p \in P[\gamma]$, $| {}^\bullet p | = | p^\bullet | = 1$. Hence, ${}^\bullet P[\gamma] = P[\gamma]^\bullet$, and $P[\gamma]$ is a siphon. Since there exists a place $r \in R$ such that $r \in S$ but $r \notin P[\gamma]$, we have $P[\gamma] \subset S$. However, since S is a minimal siphon, there does not exists any siphon $S' = P[\gamma] \subset S$.

Proof of Property 16. (by contradiction) Suppose there exists $s \in (S \setminus R)$ such that $t \in s^\bullet$. By definition of augmented marked graphs, $|\,{}^\bullet s\,| = |\,s^\bullet\,| = 1$. S is covered by cycles in accordance with Property 15. Hence, t is the one and only one transition in $s^\bullet$, where $t \in T[Y] = (S^\bullet \cap {}^\bullet S)$. This however contradicts $t \in (S^\bullet \setminus {}^\bullet S)$.

Proof of Property 17. Let S be a NR-siphon. According to Property 13, S is marked. By definition of augmented marked graphs that, for any $s \in S$, $|\,{}^\bullet s\,| = |\,s^\bullet\,| = 1$. Then, ${}^\bullet S = S^\bullet$ and S is also a trap. Hence, S contains itself as a marked trap and would never become empty.

Proof of Property 18. (⇐) According to Property 17, a NR-siphon would never become empty. Given that no R-siphons (and hence, no minimal siphon) eventually become empty, according to Property 8, (N, M_0; R) is live and reversible. (⇒) It follows from Property 6 that no R-siphons eventually become empty.

Proof of Property 19. (⇐) According to Property 17, a NR-siphon contains a marked trap. Given that every R-siphon contains a marked trap, every minimal siphon contains a marked trap. (⇒) Since R-siphons are minimal siphons, every R-siphon contains a marked trap.

Proof of Property 20. For (N, M_0; R), if every R-siphon contains a marked trap, according to Property 19, the siphon-trap property is satisfied. Hence, every minimal siphon contains a marked trap and would never become empty. It then follows from Property 8 that (N, M_0; R) is live and reversible.

Proof of Property 21. Since S is a minimal siphon, according to Property 10, for any $q, q' \in S = P[Y]$, there exists in $S = P[Y]$ a conflict-free path from q to q'. Hence, Y is conflict free.

Proof of Property 22. Let $S = \{ p_1, p_2, ..., p_n \}$. According to Property 13, S is marked. It follows from Properties 15 and 21 that there exists a set of cycles $Y \subseteq \Omega_N[R]$, such that Y is conflict-free and $P[Y] = S$. Since S is a siphon, for each $p_i \in S$, ${}^\bullet p_i \subseteq ({}^\bullet S \cap S^\bullet) = ({}^\bullet P[Y] \cap P[Y]^\bullet) = T[Y]$. In case $p_i \notin R$, $p_i^\bullet \subseteq T[Y]$ because $|\,{}^\bullet p_i\,| = |\,p_i^\bullet\,| = 1$. In case $p_i \in R$, given that p_i satisfies the R-inclusion property, $p_i^\bullet \subseteq T[Y]$. Every $p_i^\bullet \subseteq T[Y] = ({}^\bullet P[Y] \cap P[Y]^\bullet)$ and $p_i^\bullet \subseteq {}^\bullet P[Y] = {}^\bullet S$. Since $S^\bullet = (p_1^\bullet \cup p_2^\bullet \cup ... \cup p_n^\bullet) \subseteq {}^\bullet S$, S is also a trap.

Proof of Property 23. (⇐) It follows from Properties 19 and 22. (⇒ by contradiction) Suppose there exists $r \in R$, not satisfying the R-inclusion property. According to Property 14, there exists a R-siphon S, in which r is the only marked place. It follows from Properties 15 and 21 that there exists $Y \subseteq \Omega_N[R]$, such that Y is conflict-free and $S = P[Y]$. According to Property 19, S contains a marked trap Q. Then, $r \in Q$ and $r^\bullet \subseteq ({}^\bullet Q \cap Q^\bullet)$. Since S is a siphon, we have ${}^\bullet r \subseteq ({}^\bullet S \cap S^\bullet) = ({}^\bullet P[Y] \cap P[Y]^\bullet) = T[Y]$. However, as r does not satisfy the R-inclusion property, $r^\bullet \not\subseteq T[Y] = ({}^\bullet P[Y] \cap P[Y]^\bullet) = ({}^\bullet S \cap S^\bullet)$, implying $r^\bullet \not\subseteq ({}^\bullet Q \cap Q^\bullet)$.

Proof of Property 24. According to Property 23, (N, M_0; R) satisfies the siphon-trap property. It follows from Property 20 that (N, M_0; R) is live and reversible.

Proof of Property 25. For each place $p \notin R$ in N, M_0; R), $|\,{}^\bullet p\,| = |\,p^\bullet\,| = 1$. Each place $r \in R$ is replaced by a set of places $\{ p_1, p_2, ..., p_{kr} \}$, where $|\,{}^\bullet p_i\,| = |\,p_i^\bullet\,| = 1$ for $i = 1, 2, ..., k_r$. Hence, for every place q in N', $|\,{}^\bullet q\,| = |\,q^\bullet\,| = 1$. (N', M_0') is a marked graph.

Proof of Property 26. Let (N', M_0') be the R-transform of an augmented marked graph (N, M_0; R). As the transformation does not create cycles, cycles in (N', M_0') exist in (N, M_0; R). According to Property 12, cycles in (N, M_0; R) are marked, and hence, cycles in (N', M_0') are marked. Since (N', M_0') is a marked graph, it follows from Property 2 that (N', M_0') is live.

Proof of Property 27. Let $D_r = \{\langle t_{s1}, t_{h1}\rangle, \langle t_{s2}, t_{h2}\rangle, ..., \langle t_{skr}, t_{hkr}\rangle\}$. By definition of augmented marked graphs, for each $\langle t_{si}, t_{hi}\rangle$, there exists an unmarked path $\rho = \langle t_{s1}, ..., t_{h1}\rangle$ in (N, M_0; R). Hence, ρ also exists as an unmarked path in (N', M_0'), and ρ together with q_i forms a cycle γ_i which is marked at q_i only. Since (N', M_0') is a marked graph, according to Property 5, the corresponding vector of γ_i is a place invariant α_i of N'. Since q_i is the only one marked place in γ_i, $\alpha_i[q_i] = 1$ and $\alpha_i[s] = 0$ for any $s \in P_0 \setminus \{q_i\}$.

Proof of Property 28. Let $P = \{ p_1, p_2, ..., p_n \}$ be the places in N', and $P_0 \subseteq P$ be those marked places. Since each p_i belongs to a cycle γ_i and (N', M_0') is a marked graph, according to Property 5, the corresponding vector of γ_i is a place invariant α_i' of N'. Then, $\alpha' = \alpha_1' + \alpha_2' + ... + \alpha_n' > 0$ is a place invariant of N'. Consider $Q_i = \{ q_1, q_2, ..., q_k \}$. Let $q_m \in Q_i$ such that $\alpha'[q_m] \geq \alpha'[q_j]$ for any $q_j \in Q_i$. For each q_j, according to Property 27, there exists a place invariant $\alpha_j' > 0$ such that $\alpha_j'[q_j] = 1$ and $\alpha_j'[s] = 0$ for any $s \in P_0 \setminus \{q_j\}$. There also exists a place invariant $\alpha'' = \alpha' + h\alpha_j'$, where $h \geq 1$, such that $\alpha''[q_j] = \alpha''[q_m]$ and $\alpha''[s] = \alpha'[s]$ for any $s \in P_0 \setminus \{q_j\}$. Therefore, there eventually exists a place invariant α of N' such that $\alpha[q_1] = \alpha[q_2] = ... = \alpha[q_k]$.

Proof of Property 29. (⇐) Let $R = \{ r_1, r_2, ..., r_n \}$, where each r_i is being replaced by a set of places Q_i, for $i = 1, 2, ..., n$. Since every place in (N', M_0') belongs to a cycle, according to Property 28, there exists a place invariant α' of N' such that $\alpha' > 0$ and $\alpha'[q_1] = \alpha'[q_2] = ... = \alpha'[q_k]$ for each $Q_i = \{ q_1, q_2, ..., q_k \}$. Intuitively, there also exists a place invariants α of N such that $\alpha > 0$ and $\alpha[r_i] = \alpha'[q_1] = \alpha'[q_2] = ... = \alpha'[q_k]$ for each Q_i. Hence, (N, M_0; R) is conservative. According to Property 1, (N, M_0; R) is also bounded. (⇒) Since (N, M_0; R) is conservative, there exists a place invariant α of N such that $\alpha > 0$. Consider each $r_i \in R$ which is being replaced by $Q_i = \{ q_1, q_2, ..., q_k \}$. Intuitively, there also exists a place invariant α' of N' such that $\alpha' > 0$ and $\alpha'[q_1] = \alpha'[q_2] = ... = \alpha'[q_k] = \alpha[r_i]$ and $\alpha'[s] = \alpha[s]$ for any $s \in P' \setminus Q_i$. Hence, (N', M_0') is conservative. It follows from Property 1 that (N', M_0') is also bounded. Since (N', M_0') is a marked graph, according to Property 3, every place in (N', M_0') belongs to a cycle.

Proof of Property 30. It follows from Properties 3 and 29.

Learning Predictive Clustering Rules

Bernard Ženko
Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
E-mail: bernard.zenko@ijs.si
http://www.fri.uni-lj.si/file/73407/zenko-phd-thesis.pdf

Thesis Summary

Keywords: machine learning, predictive clustering, rule learning

Received: December 14, 2007

The article presents the abstract of doctoral dissertation on learning predictive clustering rules.

Povzetek: Članek predstavlja povzetek doktorske disertacije o učenju pravil za napovedno razvrščanje.

1 Introduction

In the thesis [10] we developed and empirically evaluated a method for learning predictive clustering rules. The method [10, 9] combines ideas from supervised and unsupervised learning and extends the predictive clustering approach to methods for rule learning. In addition, it generalizes rule learning and clustering. The newly developed algorithm is empirically evaluated, in terms of performance, on several single and multiple target classification and regression problems. The new method compares favorably to existing methods. The comparison of single target and multiple target prediction models shows that multiple target models offer comparable performance and drastically lower complexity than the corresponding sets of single target models.

2 Thesis overview

The predictive clustering approach [1, 2] builds on ideas from two machine learning areas, predictive modeling and clustering [6]. Predictive modeling is concerned with the construction of models that can be used to predict some object's target property from the description of this object. Clustering, on the other hand, is concerned with grouping of objects into classes of similar objects, called clusters; there is no target property to be predicted, and usually no symbolic description of discovered clusters. Both areas are usually regarded as completely different tasks. However, predictive modeling methods that partition the example space, such as decision trees and rules are also very similar to clustering [7]. They partition the set of examples into subsets in which examples have similar values of the target variable, while clustering produces subsets in which examples have similar values of all descriptive variables. Predictive clustering builds on this similarity. As is common in 'ordinary' clustering, predictive clustering constructs clusters of examples that are similar to each other, but in general taking both the descriptive and the target variables into account. In addition, a predictive model is associated with each cluster which describes the cluster, and, based on the values of the descriptive variables, predicts the values of the target variables.

Methods for predictive clustering enable us to construct models for predicting multiple target variables which are normally simpler and more comprehensible than the corresponding collection of models, each predicting a single variable. So far, this approach has been limited to the tree learning methods. The aim of the thesis was to extend predictive clustering towards methods for learning rules, i.e., to develop a method for learning predictive clustering rules. Of the existing rule learning methods, majority are based on the sequential covering algorithm [8], originally designed for learning ordered rule lists for binary classification domains. We have developed a generalized version of this algorithm that enables learning of ordered or unordered rules, on single or multiple target classification or regression domains. The newly developed algorithm is empirically evaluated on several single and multiple target classification and regression problems.

3 Conclusion

The work presented in the thesis comprises several contributions to the area of machine learning. First, we have developed a new method for learning unordered single target classification rules. It is loosely based on the commonly used rule learning method CN2 [4, 3], but uses a generalized *weighted covering* algorithm [5].

Second, the developed method is generalized for learning ordered or unordered rules, on single or multiple target classification or regression domains. It uses a search heuristic that takes into account several rule quality measures and is applicable to all the above mentioned types of domains.

The third contribution is the extension of the predictive clustering approach to models in the form of rules. The developed method combines rule learning and clustering. The search heuristic takes into account the values of both the target and the descriptive attributes. Different weighting of these two types of attributes enable us to traverse from predictive modeling to clustering.

The final contribution is an extensive empirical evaluation of the newly developed method on single target classification and regression problems, as well as multiple target classification and regression problems. Performance of the new method is compared to some existing methods. The results show that on single target classification problems, the performance of predictive clustering rules (PCRs) is comparable to that of CN2 rules and predictive clustering trees (PCTs), while in the case of unordered rules, PCRs are better than CN2 rules. Unordered PCRs are in general better than ordered PCRs. On multiple target classification problems, PCRs are comparable to PCTs, but PCRs tend to produce smaller rule sets than (transcribed) trees. Single target regression PCRs are comparable to existing regression rule methods, however, their performance is much worse than that of PCTs; on multiple target regression problems, PCRs are also much worse than PCTs. We believe the main reason that PCTs are better than PCRs on regression problems is the fact that PCTs use a state-of-the-art post-pruning method, while PCRs use no post-pruning. The comparison of the performance of single target and multiple target PCRs on multiple target problems shows, that multiple target prediction provides comparable accuracy as single target prediction, but multiple target prediction rule sets are much smaller than the corresponding single target rule sets.

References

[1] H. Blockeel (1998) *Top-down Induction of First Order Logical Decision Trees*. PhD thesis, Katholieke Universiteit Leuven, Belgium.

[2] H. Blockeel, L. De Raedt, J. Ramon (1998) Top-down induction of clustering trees. In *Proc of the ICML 98*, pp 55–63, San Francisco, Morgan Kaufmann.

[3] P. Clark, R. Boswell (1991) Rule induction with CN2: Some recent improvements. In *Proc of the* 5^{th} *EWSL*, pp 151–163, Berlin, Springer.

[4] P. Clark, T. Niblett (1989) The CN2 induction algorithm. *Machine Learning*, 3(4): 261–283.

[5] D. Gamberger, N. Lavrač (2002) Expert guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527.

[6] L. Kaufman, P. J. Rousseeuw (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley & Sons, New York.

[7] P. Langley (1996) *Elements of Machine Learning*. Morgan Kaufmann, San Francisco.

[8] R. S. Michalski (1969) On the quasi-minimal solution of the general covering problem. In *Proc of the FCIP 69*, pp 125–128, Bled.

[9] B. Ženko, S. Džeroski, J. Struyf (2006) Learning predictive clustering rules. In *Knowledge Discovery in Inductive Databases,* 4^{th} *Int Wshp, Revised Selected and Invited Papers*, pp 234–250. Springer.

[10] B. Ženko (2007) *Learning predictive clustering rules,* PhD thesis, University of Ljubljana, Slovenia.

Estimation of Individual Prediction Reliability Using Sensitivity Analysis of Regression Models

Zoran Bosnić
University of Ljubljana, Faculty of Computer and Information Science,
Tržaška 25, Ljubljana, Slovenia
E-mail: zoran.bosnic@fri.uni-lj.si
http://lkm.fri.uni-lj.si/zoranb/dissertation.htm

Thesis Summary

Keywords: regression, predictions, correction of predictions, sensitivity analysis, prediction error, prediction accuracy

Received: February 10, 2008

The paper is the extended abstract of dissertation which is concerned with the estimation of reliability for the individual predictions of regression models (in contrast to estimating the accuracy of the whole model) and with the use of sensitivity analysis in that area. The dissertation studies the ways of optimal reliability estimate selection among 9 studied reliability estimates and evaluates the methodology on large number of standard benchmark domain as well as on real domains.

Povzetek: Članek povzema doktorsko disertacijo, ki se ukvarja z ocenjevanjem zanesljivosti posameznih regresijskih napovedi v strojnem učenju.

1 Introduction

The dissertation [1, 2, 3] discusses the reliability estimation of individual regression predictions in the field of supervised learning. In contrast with average measures for the evaluation of model accuracy (e.g. mean squared error), the reliability estimates for individual predictions can provide additional information which could be beneficial for evaluating the usefulness of the prediction and possible consequential actions.

Measuring the expected prediction error is very important in risk-sensitive areas where acting upon predictions may have financial or medical consequences (e.g. medical diagnosis, stock market, navigation, control applications). In such areas, appropriate *local* accuracy measures may provide additional necessary information about the prediction confidence. The described challenge is illustrated in Figure 1.

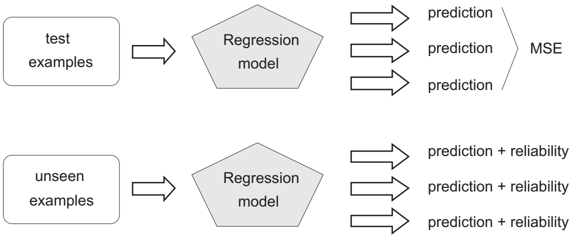


Figure 1: Reliability estimate for the whole regression model (above) in contrast to reliability estimates for individual predictions (below).

The methods for reliability estimation of individual prediction can be either bound to a particular model formalism [4] or be model-independent and therefore more general. The dissertation focuses on the objective to develop a new approach from the second group.

2 Reliability Estimates

The dissertation proposes 9 new individual prediction reliability estimates which are independent of the regression model. Three of newly proposed estimates are developed by adapting the sensitivity analysis [5] approach for the use in supervised learning. To apply the principles of the sensitivity analysis, we propose a framework for controlled modification of the input (learning set) and outputs (regression predictions) in supervised learning setting. By making minor modifications in the learning set we exploit the instabilities in predicted values and use them to compose reliability estimates. Six remaining estimates are either adapted from related work (three estimates are generalized for use with all 8 regression models), newly proposed using local error modeling approach (two estimates) or based on linearly combining two individual estimates among former (one estimate). The linear combination of estimates is performed for all combinations of eight individual estimates by equally weighing (averaging) estimates in the combination. The best combined estimate was selected for further evaluation.

3 Automatic Selection of the Best Estimate

We study the problem of the reliability estimate selection based on the given problem domain and the regression model. We discuss and define two possible solutions of this problem, based on meta-learning and internal cross-validation approach. In the context of meta-learning we propose a possible attribute description of the meta-learning problem and define it as a classification problem, where each class represents one of the 9 proposed reliability estimates. We also use the meta-classifier to explain which estimate is optimal for the given model and domain properties.

In the approach with internal cross-validation, we select such estimate for the use with the test examples which achieved the best results on the separate subset of learning examples. This approach was tested in standard cross-validation manner.

4 Results and Conclusion

The testing of reliability estimates was performed by correlating the estimates with the prediction error and by statistically evaluating the obtained correlations. For testing, 28 standard benchmark domains from publicly accessible repositories and 8 regression models were used (regression trees, linear regression, neural networks, bagging, support vector regression, locally weighted regression, random forests, generalized additive model). The testing results showed usefulness of the proposed reliability estimates especially for the use with regression trees, where one of the proposed estimates correlated with the prediction error in 86% of the testing domains. Both methods for automatic selection of reliability estimates outperformed individual estimates.

The individual estimates and both approaches for automatic selection of the optimal estimate were tested in a real domain from the area of medical prognostics. The results exhibited a significant number of correlations between the reliability estimates and the prediction error in the majority of tests. The statistical comparison of reliability estimates to prediction evaluations of the medical experts showed that our reliability estimates correlate to prediction error with statistically equal correlation as the manual evaluations of the experts do. This results therefore showed the potential of the proposed methodology in practice.

References

[1] Z. Bosnić (2007) *Estimation of individual prediction reliability using sensitivity analysis of regression models (in Slovene)*, PhD Thesis, University of Ljubljana, Faculty of Computer and Information Science, `http://lkm.fri.uni-lj.si/zoranb/dissertation.htm`.

[2] Z. Bosnić and I. Kononenko (2007) Estimation of individual prediction reliability using the local sensitivity analysis, *Applied intelligence (online edition)*, `http://www.springerlink.com/content/e27p2584387532g8/?p=d31c3f43a44546ee91350625fe68e4e1&pi=2`, pp. 1–17.

[3] Z. Bosnić and I. Kononenko (2008) Estimation of regressor reliability, *Journal of intelligent systems, volume 17(1/3)*, pp. 297–311.

[4] C. Saunders, A. Gammerman and V. Vovk (1999) Transduction with Confidence and Credibility, *Proceedings of IJCAI 2*, pp. 722–726.

[5] O. Bousquet and A. Elisseeff (2002) Stability and generalization, *Journal of Machine Learning Research 2*, pp. 499–526.

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 800 staff, has 600 researchers, about 250 of whom are postgraduates, nearly 400 of whom have doctorates (Ph.D.), and around 200 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.

From 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.

Part of the Institute was 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 was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, 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 the Economy, 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 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit an email with the manuscript to one of the editors from the Editorial Board or to the Managing Editor. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

QUESTIONNAIRE

□ Send Informatica free of charge

□ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

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 ten 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://www.informatica.si/

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Iván Araujo, Vladimir Bajič, Michel Barbeau, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Simon Beloglavec, Sondes Bennasri, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszoermenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Pavel Brazdil, Bostjan Brumen, Jerzy Brzezinski, Marian Bubak, Davide Bugali, Troy Bull, Sabin Corneliu Buraga, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Giacomo Cabri, Netiva Caftori, Particia Carando, Robert Cattral, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, Mel Ó Cinnéide, David Cliff, Maria Cobb, Jean-Pierre Corriveau, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Bart de Decker, Deborah Dent, Andrej Dobnikar, Sait Dogru, Peter Dolog, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Marjan Družovec, Jozo Dujmović, Pavol Ďuriš, Amnon Eden, Johann Eder, Hesham El-Rewini, Darrell Ferguson, Warren Fergusson, David Flater, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Stan Franklin, Violetta Galant, Hugo de Garis, Eugeniusz Gatnar, Grant Gayed, James Geller, Michael Georgiopolus, Michael Gertz, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Jozsef Gyorkos, Marten Haglind, Abdelwahab Hamou-Lhadj, Inman Harvey, Jaak Henno, Marjan Hericko, Henry Hexmoor, Elke Hochmueller, Jack Hodges, John-Paul Hosom, Doug Howe, Rod Howell, Tomáš Hruška, Don Huch, Simone Fischer-Huebner, Zbigniew Huzar, Alexey Ippa, Hannu Jaakkola, Sushil Jajodia, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričič, Marko Juvancic, Sabhash Kak, Li-Shan Kang, Ivan Kapustøk, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Fabio Kon, Kevin Korb, Gilad Koren, Andrej Krajnc, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Sofiane Labidi, Les Labuschagne, Ivan Lah, Phil Laplante, Bud Lawson, Herbert Leitold, Ulrike Leopold-Wildburger, Timothy C. Lethbridge, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Vincenzo Loia, 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, Armin R. Mikler, Gautam Mitra, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Jari Multisilta, Hari Narayanan, Jerzy Nawrocki, Rance Necaise, Elzbieta Niedzielska, Marian Niedq'zwiedziński, Jaroslav Nieplocha, Oscar Nierstrasz, Roumen Nikolov, Mark Nissen, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Daniel Olejar, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, Jens Penberg, William C. Perkins, Warren Persons, Mitja Peruš, Fred Petry, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Peter Planinšec, Gabika Polčicová, Gustav Pomberger, James Pomykalski, Tomas E. Potok, Dimithu Prasanna, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Vojislav D. Radonjic, Luc de Raedt, Ewaryst Rafajlowicz, Sita Ramakrishnan, Kai Rannenberg, Wolf Rauch, Peter Rechenberg, Felix Redmill, James Edward Ries, David Robertson, Marko Robnik, Colette Rolland, Wilhelm Rossak, Ingrid Russel, A.S.M. Sajeev, Kimmo Salmenjoki, Pierangela Samarati, Bo Sanden, P. G. Sarang, Vivek Sarin, Iztok Savnik, Ichiro Satoh, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer, Zhongzhi Shi, Mária Smolárová, Carine Souveyet, William Spears, Hartmut Stadtler, Stanislaw Stanek, Olivero Stock, Janusz Stokłosa, Przemysław Stpiczyński, Andrej Stritar, Maciej Stroinski, Leon Strous, Ron Sun, Tomasz Szmuc, Zdzislaw Szyjewski, Jure Šilc, Metod Škarja, Jiři Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, Gheorge Tecuci, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Drago Torkar, Vladimir Tosic, Wieslaw Traczyk, Denis Trček, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Romana Vajde Horvat, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Jan Verschuren, Zygmunt Vetulani, Olivier de Vel, Didier Vojtisek, Valentino Vranić, Jozef Vyskoc, Eugene Wallingford, Matthew Warren, John Weckert, Michael Weiss, Tatjana Welzer, Lee White, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Tatyana Yakhno, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Ales Zivkovic, Zonling Zhou, Robert Zorc, Anton P. Železnikar

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

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 2008 (Volume 32) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: Dikplast Kregar Ivan s.p., Kotna ulica 5, 3000 Celje.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) 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)
ACM Slovenia (Dunja Mladenič)

Informatica is surveyed by: Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

The issuing of the Informatica journal is financially supported by the Ministry of Higher Education, Science and Technology, Trg OF 13, 1000 Ljubljana, Slovenia.

Volume 32 Number 1 April 2008 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Informatica **32** (2008) Number 1, pp. 1–99