

BigNFC: Novel Formal Model for NFC Based Context-aware Applications

Nabet Aicha and Boudour Rachid

Embedded systems Laboratory, Badji Mokhtar Annaba University, Algeria

E-mail: arijeyasmine@hotmail.fr, racboudour@yahoo.fr

Keywords: near field communication, context-aware application, Bigraphical Reactive System (BRS), BigMC

Received: April 6, 2021

Context-aware computing refers to system ability to sense its environment and change its behavior for delivering suitable services. Having such systems with the Near Field Communications (NFC) capability, opens new perspectives and research areas, allowing very useful type of applications known as NFC-based context-aware applications. These systems require correctness because of their applicability and then need to be proven formally using exhaustive analysis approach such as formal verification. In literature, most of works focuses on creating a general model for context-aware systems ignoring the specificity of certain applications such as NFC applications where they present a higher complexity. We emphasize the existence of little or no work in this area supporting formal modeling. To boost it, we propose BigNFC as a novel formal-model based on Bigraphical Reactive Systems (BRS) taking account the interaction mode from the beginning, so we establish mapping between BRS and BigNFC components, where the structures are modelled as bigraphs and behaviors as rewriting rules. Finally, to validate our model, we have applied it on a real-life application and some properties were checked successfully.

Povzetek: Razvit je nov formalni sistem za NFC prepoznavanje na osnovi konteksta.

1 Introduction

Every day, technology does not stop surprising us. In 1988, Wisner marked his name as one pioneer of ambient computing also called pervasive computing. Its primary objective is to create an intelligent environment. It makes environment more sensitive and more interactive around user to deliver a set of new digital, adaptive services that could improve them. It uses several innovative technologies as networks, contactless technologies (RFID, NFC), mobile technologies etc. Ambient computing involved many disciplines such as context aware computing, mobile computing and so on. Far away from the classic scheme of interaction, Near Field Communications (NFC) proposes an original communication between the physical and the virtual world which is integrated in several fields such as M-payment[1]. NFC is a contactless radio technology that can exchange data between two devices within a few centimeters of each other. It operates in the standard unlicensed 13.56 MHz frequency.

Currently, it offers data transfer rates of 106 kbit/s, 212 Kbit/s and 424 kbit/s. NFC provides a seamless medium for the identification protocols that validate secure data transfer. This enables users to perform intuitive and safe contactless transactions. NFC technology proposes three operating modes[2][3]: **Reader/Writer mode**, enables NFC phone to read/write data from/to an NFC tag; **Peer-to-Peer mode**, allows two NFC mobiles to exchange information such as contact record, a text message, or any other data; and **Card**

emulation mode, makes mobile phone able to function as contactless smart card.

NFC objects can be very useful to construct more sensitive and interactive entourage, some successful projects around world using NFC have emerged such as smart university Córdoba [4], Nice smart City [5]... They can play the role of context collectors in smart environment [6], which can be interpreted to construct an adaptable service using context awareness computing techniques. Context awareness as term has been evoked at first time by [7] in their works on localization awareness. In [8], they defined context awareness as the ability of an application to adapt the context of its execution according to location, all people nearby, machines, accessible equipment, as well as changes in these objects in time. In [9] [10], context awareness is defined as the ability of application to consider the context of the user. Dey [11] [12], explained context awareness as a capacity of system to use context to provide information and/or pertinent services to user. Context-aware system presents the ability of system to adapt services to satisfy the preferences of user in dynamic environment. Certainly, many models are used to present context such as model-based ontology [13], graphical model [14] and so on.

Such systems need correctness and precision, formal verification offers a set of very strong mathematic theories, techniques and tools to prove a validity of work. In 2001, Milner [15] [16] proposes very practical formal formalism named bigraph, which is used in many articles to formalize context aware system. In the proposed model

baptized BigNFC, we use BRS formalism to elaborate NFC based context-aware applications thus the first work takes seriously modelling of NFC based context-aware applications. BRS allows us to translate our architecture elements to bigraphs and the behaviors to bigraph reactive system BRS. Most of the articles published in context aware are most concerned with localization, although the context is more extensive than the localization. For this reason, we take an example of modelling an application focused on user preferences sensibility and we check safety properties with BigMC tool. The rest of this paper is organized as follows. In section one; we present basic concepts, which is very necessary for readers to understand our work. In section two, we give a small panorama of works in the same field. In section three, we show how to construct BigNFC model and we apply it with an experimental example; by the way, we present the related results of verification. In section four, we discuss our results. Finally, in section five, we conclude and mention our future work.

2 Basic concepts

BRS is one of the most powerful formal models. It uses some concepts and a specific grammar where it offers the possibility of using a set of mathematical tools for the verification as BigMC.

BigMC is a formal verification tool for BRS model; its use needs the mastery of some notions. So, to make reading of this article easier; BRS and BigMC concepts are required:

2.1 BRS

Milner provided a new graphical formalism to present simultaneous connectivity and location of entity baptized bigraph. The combination between bigraph and reaction rules forms bigraph reactive systems (BRS). Reaction rules describe a dynamic aspect of formalism, which is a couple of bigraphs, *redex bigraph*, shows the preconditions of a reaction, and *reactum bigraph*, is the post-condition showing how the reaction will change the model.

Formally, bigraph is $G = (V, E, Ctrl, G^P, G^L) I \rightrightarrows J$

Where, V : finite set of nodes. A node can have connexion points to edges or (inner/outer) names called ports. The outer and inner names are for interfacing, they make a bigraph able to interact and to connect with externals bigraphs or roots;

E : finite set of edges;

$Ctrl$: Control map, has the following form $V \rightrightarrows K$. Control is an associate information that describes the node. We may define a control K as atomic, so nothing may be nested in a K -node otherwise it can be specified as active, that means K -node allows reactions inside. It can passive, that means before inhabitant nodes in K -node react, its destruction is required [17]. Every control K has an arity, which shows several ports in K -node;

GP : presents place graph which, is an acyclic tree, describes the location of nodes;

GL : describes link graph that is a hyper-graph defines the connectivity of nodes;

I : inner face will take the form $I = \langle m, X \rangle$, where m is a width that represents several sites in I and X describes a set of inner names;

J : outer face will be in the form $J = \langle n, Y \rangle$, where n describes several roots in J and Y presents a set of outer name. Figure 1 shows the anatomy of bigraph [18]. Certainly, all these terms need a set of operators in order to get expressions, Table 1 presents the algebraic expressions of bigraphs.

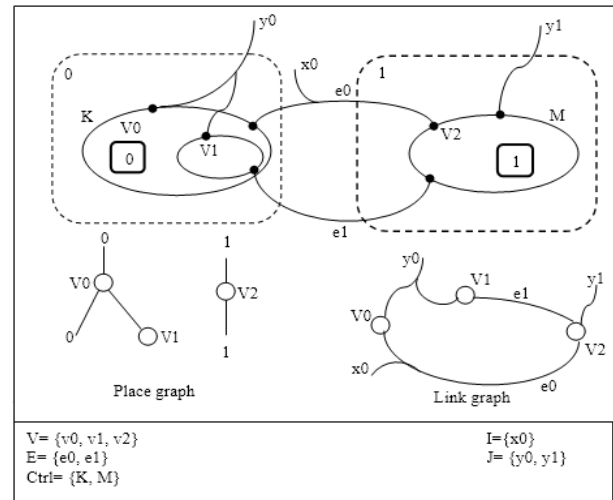


Figure 1: Anatomy of bigraph.

Term	Interpretation
$U \circ V$	Composition of nodes
$U V$	Merge product (Juxtaposition of nodes)
$U V$	Parallel (juxtaposition of roots)
$U . V$	Nesting U contains V
$U \otimes V$	Tensor product
$K_{\bar{x}}(U)$	Node with K control or arity \bar{x}
1	The barren (empty) root
d_i	Site numbered i
x/y	Connection from inner name y to outer name x

Table 1: Algebraic expressions of bigraphs.

2.2 BigMC

BiGraphical Model Checking ‘BigMC’, is a unique model checker instantiations for BRS model thus offers an automatic proof for any model based BRS. It is created by Perrone in 2012 [19], BigMC has a specific grammar. Figure 2 shows grammar for BigMC terms.

3 Related works

Including a formal method for modelling and verifying a context awareness system appears a cumbersome and painful task for designers, which explain the limited number of publications in this field.

```

M ::= E; M|E
E ::= %passive K: arity
E ::= %active K: arity
E ::= %rule n T-> T
E ::= %property n P
E ::= T -> T|T
T ::= k. T| T|T||T|$n|K|nil
K ::= k[names]|k
names ::= n, names|n
n ::= [a - z A - Z][a - z A - Z 0 - 9]*|_
P ::= matches(T)|terminalQ!P
    
```

Figure 2: Grammar for BigMC terms.

In [20], their main aim is to introduce a formal model for context awareness system using BRS. Approach focuses on the formulation of a location aware system. Model is the union of three BRS, C, A, P, where C: context, P: proxy, A: Agent. Three BRS are disjoint sets: $C \cap P \cap A = \emptyset$. This model based on the proxy architecture, where agent has not direct access to a context. They are not provided formal verification.

[21], attempts to formalize the structure and behavior of context aware system using bigraph formalism. Their paper brings an equivalence map between context elements and bigraph components to render the formalization of context awareness simpler and more convenient for designers. This work is very interesting, but it lacks verification part.

[22], proposes an alternative approach for data flow test in context aware application with an experimental on an RFID based location-sensing software. This work is to provide a testing tool for context aware middleware centric programs. A delivered tool generates automatically an adequacy set of tests for given criteria. We derived the weakness for this work from the weakness of test approach because of the coverage tests.

In [23], a new vision for context-aware systems. They find that a context-aware system is the combination of two parts: context-awareness and context-aware part, so the designers must give the desirable parts to get the model. They used bigraph formalism to describe their idea and BigMC tool to verify some properties on an example. This work is a great effort for modelling complex systems as context aware system but we find some ambiguity to understand the meaning of context-awareness part (signaling the complete absence of the concept in literature).

After presenting a general survey on works in the field of context aware using formal model, we note these works suggest a generic formal model for context aware system without focusing on specificity of some applications. We describe BigNFC model for NFC based context-aware application considering some details characterizing these applications e.g. interaction modes then we experiment it on an example from actual life.

4 How to construct BigNFC model?

NFC based context aware is a new application that uses NFC technology as collectors of context in a smart environment.

Considering the complexity of context sensitive applications, particularly NFC applications, we

recommend it to reduce the complexity by choosing the mode of interaction. i.e. each mode has a set of specificities.

We base BigNFC on producer/consumer structure to separate between the zone delivering the raw context and the one that processes it. Thus, BigNFC allows to express clearly the links between them.

We can structure smart life environment as producer zone 'physical zone' and consumer zone 'smart platform' (see Figure 3).

- **Producer zone 'physical zone'**; provides the raw context. In the physical zone, we can distinguish two qualities of resources contextual information [24]

User information: some contextual information, which, is around his personal characteristics, social life, location...

Smart objects: they are an important part in the environment that represent instruments such as sensors, communication technology (NFC, RFID...).

- **Consumer zone 'smart platform'**; treats contextual information provided from smart environment to offer adaptive services to users. It contains at least three layers:

Perception layer: precepts or gains the contextual information from smart environment. In NFC based context aware system the perception of a raw context is from NFC phone of user after touch action accomplished on NFC objects. We can extend the role of this layer for data filtering if there are heterogeneous perception resources of contextual information.

Smart agent layer: realized by a cognitive agent, who can decide and compare to adapt services for users. Smart agent compares the information with its knowledge's after it can deduce the situation using inference rules and provides a service. In NFC based context aware system detection of contextual information as preference got from a touch action realised by user in smart environment i.e. touch action involves preference detection;

Service delivery layer: allows context aware application to deliver an adaptive service to user. Figure 3 shows interactions in smart life environment.

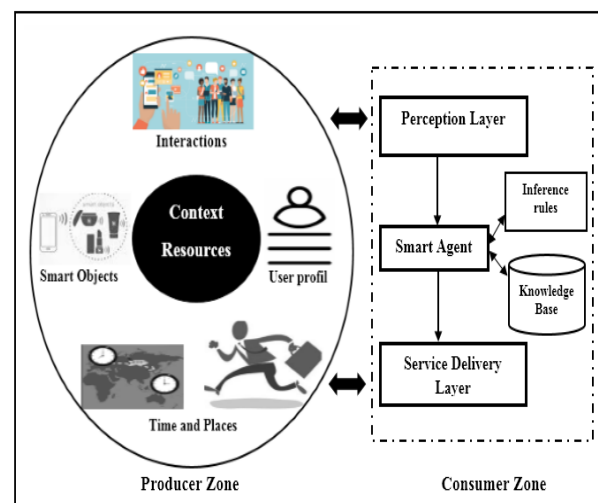


Figure 3: Interactions in smart life environment.

4.1 Equivalence map between BigNFC model and BRS components

BRS is very practical; it offers a set of elements that allows us to present separately two distinguish parts and their links with a big simplicity. In addition, it lets possible to validate formally the idea using mathematical tools as BigMC.

Technically, we use BRS formalism where, **place graph**, describes inclusion relation between elements of architecture and contextual information, **link graph**, shows the dependency between information and its transfer, **reactions rules**, define behavior system. A translation map did the mapping between the conceptual model and BRS components. Table 2 gave the mapping between the conceptual model and BRS components.

BigNFC	BRS components
Physical zone, smart platform	Root
User, NFC objects, data objects, relationship between objects, services, layers of smart platform...	Nodes
Communication links, interaction links	Edges, hyper-edges
Other applications, captures, time transaction...	Sites
Behaviours as touch action, deduction, perception...	Reaction rules

Table 2: Equivalence map between the conceptual model and BRS components.

5 Experimental example

To validate our model, we construct it on a NFC context aware application. NFC can get contextual information with intuitive manner using its three communication modes (seen in Section 1). In the example, the application allows the user to gain some information for a book and gives smart recommendation. In the physical zone, smart poster contains NFC tag (Reader-writer mode), which includes raw context (relevant information that characterises book). When the user touches a smart poster endow NFC tag, it will display important information on NFC user phone. Then smart recommendation process will be perform automatically if the book has the appropriate type and high rating. For smart recommendation, the application must know: What kind of books does user prefer? Preferences are deduced after touch action on smart poster.

5.1 Using BigNFC model

Applying BigNFC model, the structure is as follows: *Producer zone* ‘physical zone’ contains user, smart poster, NFC tag... In addition, *consumer zone* ‘smart platform’ comprises different layers of platform. Just to note, we abstract some information to simplify a modeling task.

5.2 Mapping between BigNFC model and BRS components

At first time, we give separately the equivalence of producer/ consumer structure and BRS formalism. Then we draw distinctly the bigraph of the two levels and finally, we write the two model formulas according to the grammar given by Milner. The description of behavior require several rules, but we interest to touch rule.

5.2.1 Producer zone

Below, Table 3 shows equivalence map between producer zone and Bigraph components. It gave Figure 4; it presents bigraph of initial state of producer zone and a formula according to Bigraph grammar.

Producer zone	Bigraph component
Physical Zone(Zone_PH)	Root
User, tag-NFC, smart poster(S-Poster), Book, NFC-Phone, Type Book...	Nodes
Communication links, interaction links	Edges, hyper-edges
Other applications, captures, time transaction...	Sites

Table 3: Equivalence map between producer zone elements and BRS components.

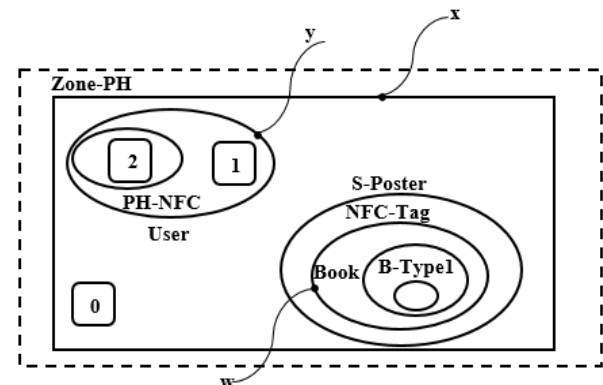


Figure 4: Initial state of produce zone with Bigraph.

Zone_PH[x].(User[y].(PH_NFC.\$₂)|\$₁) |S_Poster.
(Tag_NFC[w]. (Book.B-Type1)|\$₀)

5.2.2 Consumer zone

Table 4 shows equivalence between consumer zone and Bigraph components. Figure 5 illustrates bigraph of the initial state of consumer zone and we gave formula according to Bigraph grammar.

In the initial condition, we give the description of consumer zone using bigraph grammar.

Consumer zone	BRS Equivalence
Smart platform (S-Platform)	Root
Perception-Layer, S-Agent, Service-delivered-Layer, Knowledge base Layer, Profile-user, service, relationship between entities. Some information for entity...	Nodes
Communication links, interaction links	Edges, hyper-edges
Abstract elements	Sites

Table 4: Equivalence between the consumer zone elements and Bigraph components.

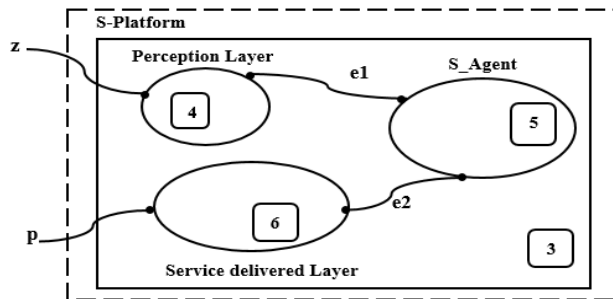


Figure 5: Initial state of consumer zone with Bigraph.

```
S-Platform. (d3 | Perception-Layer [z, e1]. $4 | S-Agent. [e1, e2]. d5 | Service delivered-Layer. d6)
```

We can describe the example application with Bigraph grammar as the concatenation of producer and consumer zones.

```
Zone_PH[x]. (User[y].(PH_NFC. $2)|$1) |S_Poster. (Tag_NFC[w. (Book.B-Type1)|$0) | S-Platform. ($3 | Perception-Layer [z,e1]. $4 | S-Agent. [e1, e2]. $5 | Service delivered-Layer. $6)
```

5.2.3 Rule: Touch action reaction rule

Touch action is major act to start the treatment phase. When user touch NFC tag, it will transfer the raw context to NFC user phone using reader mode. In our case, raw context is book data (title, authors’ names, and type of book...). Let’s remember that, in one gesture, NFC phone can read one and only one NFC tag. The selected tag, implies a preference for a book type. Figure 6 shows touch action rule.

6 How check BigNFC with BigMC?

In this subsection, we use BigMC grammar and we applied it on our proposed example. We present respectively the transformation of BRS model to BigMC grammar.

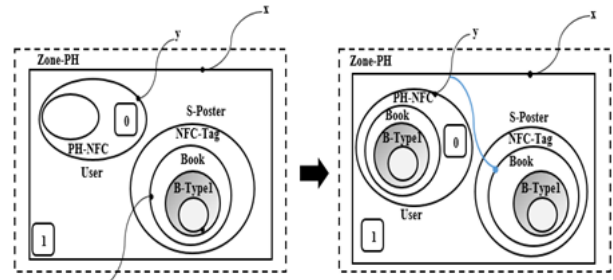


Figure 6: Touch action rule.

6.1 Producer zone with BigMC grammar

Figures 7 and 8 present respectively BigMC program of producer and consumer zones.

Figures 9 and 10 show respectively BigMC program of general model and touch action rule.

```
%Active ZonePH: 1; %name x;
%passive Sposter: 0; %name y;
%active PHNFC: 0; %name w;
%passive Bookx: 0;
%active User: 1;
%passive Type1: 1;
```

##initial representation on produce zone##

```
ZonePH[-].((User[-].PHNFC)|(Sposter.TagNFC[-.Book x.(Type1[-])));
```

Figure 7: BigMC program of producer zone.

```
%active DelivredLayer: 2;
%active SPlatform: 0;
%active PerceptionLayer: 2;
%active SAgent: 2;
%name e1;
%name e2;
%name e3;

#####initial representation of level two#####
SPlatform.(PerceptionLayer[-,e1] |SAgent[e1,e2] |DelivredLayer[e2,-]);
```

Figure 8: BigMC program of consumer zone.

```
ZonePH[-].((User[-].PHNFC)|(SPoster.TagNFC[-.BookX.(Type1[-])))|SPlatform.(PerceptionLayer[-,x] |SAgent[x,y] |DelivredLayer[y,-]);
```

Figure 9: BigMC program of general model.

```
ZonePH[.](User[w. (PHNFC.$0)|SPoster.TagNFC[w].$1) →
ZonePH[.](User[w. (PHNFC.($1))|SPoster.TagNFC[w].($1)) ;
```

Figure 10: BigMC program of touch action rule.

6.2 Property definition with BigMC grammar

In this part, we want to prove the safety i.e. the absence of deadlock using a predefinition predicate provided in BigMC grammar. :

```
# Property
```

```
%property free deadlock! terminal ();
```

We propose a secure property to confirm the validity of the model using a predefinition predicate matches. Description property is:

```
# Property
```

```
%property workable
```

```
!matches(ZonePH[].(User[e2].PHNFC.$1|$3)|SPlatform.
```

```
(PerceptionLayer[e2,x].($1|Bookn|$2)|SAGENT[x,y] $5));
```

7 Results with BigMC

In reality, with BigMC user can write its set of property with the terms or he can use redefined predicates such as matches (T), terminal ()... in our case, we verify the deadlock absence and we verify the transmission of raw context between producer and consumer zones with 20 steps. We presented results below where the green frame shows the end of verification successfully.

After checking the same property on entire proposed model with 100 steps. Result is indicated in Figure 12.

Then we verify the second property entirely on proposed model over 20 steps. We illustrate the result in Figure 13.

8 Discussions

We can show our work as a new vision in NFC based context-aware area. First, the originality of our work lives mainly in construction of a specific formal model called BigNFC. In short, we show how we can present NFC technology in a smart environment and how to interact with it (in the chosen mode). Second, a decomposition principle is widely used to reduce complexity and applied it to model our system. So, we used producer/consumer structure and we separate between physical and virtual layers. To summarize contribution, we model the structure

```
bigmc> %check;
1: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,y].nil | DelivredLayer[y,-
].nil))
2: (ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,y].nil | DelivredLayer[y,-
].nil))
4: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,-].nil))
5: (ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) | SPlatform.(SAGENT[x,-
].nil | PerceptionLayer[-,x].BookKx.Type1[-].nil))
6: (ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,-].nil))
10: (SPlatform.(SAGENT[x,-].(BookKx.nil | BookKx.Type1[-].nil) |
PerceptionLayer[-,x].nil) | ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil))
.....
99: (SPlatform.(SAGENT[x,-].(BookKx.nil | beliveslike.nil |
KnowledgeB.Puser.LikeB.BookKx.nil) |delivredLayer[-,].Book3.nil |
PerceptionLayer[-,x].nil) | zonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) |
ZonePH[-].(User[-].PHNFC.nil))
100: (ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) | ZonePH[-].(User[-
].PHNFC.nil) | Platform.(SAGENT[x,-].(BookKx.nil | beliveslike.nil |
KnowledgeB.Puser.LikeB.BookKx.nil | Book.(Book3.Type1.nil
| Book2.Type2.nil) | DelivredLayer[-,-].nil | PerceptionLayer[-,x].nil))
[mc::step] Interrupted! Reached maximum steps: 100
[mc::report] [q: 63 / g: 163] @ 100
```

Figure 12: Deadlock property checking (entire model).

```
bigmc> %check;
1: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,y].nil | DelivredLayer[y,-
].nil))
2: (ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,y].nil | DelivredLayer[y,-
].nil))
4: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].() | SAGENT[x,-].nil))
5: (ZonePH[-].(User[-].PHNFC.BookKx.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].(BookKx.Type1[-].nil) | SAGENT[x,-].nil))
.....
12: (ZonePH[-].(User[-].PHNFC.(Bookn.nil | BookKx.Type1[-].nil | Bookn.nil
| Bookn.nil) | SPlatform.(PerceptionLayer[-,x].(Bookn.nil | BookKx.Type1[-
].nil | Bookn.nil | Bookn.nil | Bookn.nil) | SAGENT[x,-].nil))
13: (ZonePH[-].(User[-].PHNFC.(Bookn.nil | BookKx.Type1[-].nil | Bookn.nil
| Bookn.nil) | SPlatform.(PerceptionLayer[-,x].(Bookn.nil | BookKx.Type1[-
].nil | Bookn.nil | Bookn.nil) | SAGENT[x,-].nil))
14: (ZonePH[-].(User[-].PHNFC.(Bookn.nil | BookKx.Type1[-].nil | Bookn.nil
| Bookn.nil | Bookn.nil) | SPlatform.(PerceptionLayer[-,x].(Bookn.nil |
BookKx.Type1[-].nil | Bookn.nil | Bookn.nil | Bookn.nil) |
SAGENT[x,-].nil))
.....
20: (ZonePH[-].(User[-].PHNFC.(BookKx.Type1[-].nil | Bookn.nil | Bookn.nil
| Bookn.nil | Bookn.nil | Bookn.nil | Bookn.nil) | SPlatform.(PerceptionLayer[-,x].(Bookn.nil
| BookKx.Type1[-].nil | Bookn.nil | Bookn.nil) |
SPlatform.(PerceptionLayer[-,x].(Bookn.nil | BookKx.Type1[-].nil | Bookn.nil
| Bookn.nil | Bookn.nil | Bookn.nil | Bookn.nil) | SAGENT[x,-
].nil))
[mc::step] Interrupted! Reached maximum steps: 20
[mc::report] [q: 2 / g: 22] @ 20
```

Figure 13: Secure property checking (entire model).

as bigraph, behaviours as rewriting rules, and a proposed mapping between BRS elements and BigNFC components (as shown in Table 1) which can be extended to other classes of context aware system. Third, the chosen example is a none trivial application, which is focused on user preferences sensibility. Finally, we exploit the example and we check successfully correctness of our

```
bigmc> %check;
1: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookKx.(Type1[-].nil |
Grat.nil)) | SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,y].nil |
DelivredLayer[y,-].nil))
2: (ZonePH[-].(User[-].PHNFC.BookKx.(Type1[-].nil | Grat.nil)) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,y].nil | DelivredLayer[y,-
].nil))
4: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookKx.(Grat.nil |
Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,-].nil))
.....
19: (ZonePH[-].(User[-].PHNFC.BookKx.(Type1[-].nil | Grat.nil)) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,-].(BookKx.(Grat.nil |
Type1[-].nil) | Like.(Grat.nil | Type1[-].nil) | BookKx.nil | Like.nil)))
20: (ZonePH[-].(User[-].PHNFC.BookKx.(Type1[-].nil | Grat.nil)) |
SPlatform.(PerceptionLayer[-,x].nil | SAGENT[x,-].(BookKx.nil |
BookKx.(Grat.nil | Type1[-].nil))))
[mc::step] Interrupted! Reached maximum steps: 20
[mc::report] [q: 12 / g: 32] @ 20
```

Figure 11: Deadlock property checking (partially model).

BigNFC model over two properties (free deadlock partially and entirely, secure) with a specific tool for bigraph model. We can interpret the absence of deadlock by the good fluidity of the contextual information even if after its evolution and after each passage between the levels of the smart platform. Secure property, mentions the relevance in communication between levels.

The works [20][21][22][23] described in the third section can be divided in three categories: the first one, is interested to the specification and next one, described a new tool for verification and applied to an RFID example and last one, depicted the two phases of verification process.

Compared to our work, authors in [20], [21] focussed mainly on using the powerful of bigraph formalism in order to design context aware application. However, they have only proposed a general model exploiting bigraph applied over localisation application without the checking step.

However, in [22] authors presented a testing tool for context aware application. This latter generates automatically an adequacy set of tests for a criteria applied over RFID example. The Table 5 brought comparison between BigNFC model and works in [20][21][22] over four criteria.

Compared to [23], the only work found in literature, the result showed a single deadlock property ran over 20 iterations, however we have tested our model with two properties on the entire model with over iterations. To synthesize, we draw Table 6 indicating comparison between the two works.

In fine, we have tried to give more rigor for these applications to reassure users and simplify the designer tasks. However, we estimate that there are some weaknesses for example the presentation of distance separation of NFC components. To our knowledge, none of the works in the lively field NFC system takes a perspective similar to ours.

9 Conclusion

In this paper, we proposed BigNFC model as a new formal model based on bigraph formalism for NFC based context-aware applications. Therefore, we use a producer /consumer structure to separate between the zone delivering the raw context and the one that performs it. To achieve our idea, the static aspect of system, which is a set of components and elements, is modelled as bigraphs, and the dynamic aspect (behaviours) as graph rewriting rules. In this regard, we present an equivalence map between BRS elements and context-aware system and we applied our model on example that integrates user preferences. In order to verify properties on the model we used BigMC tool, which gives an automatic, proof of correctness. By the application of the safety properties on an example model and results are very satisfied after 100 steps.

In actual life, we know that the environment is dynamic, because of that a probabilistic model is more appropriate. We try to satisfy some points such as extend BigNFC model to a probabilistic model, add more complicated rules create synchronisation situations, and

	BigNFC	[20]	[21]	[22]
Specification step	Yes	Yes	Yes	Yes
Verification step	Model checker approach	Test approach	No	No
example based on	User preferences	User localisation	User localisation	User localisation
checker	Yes		No	No

Table 5: Comparison between works.

	BigNFC	[23]
Model with BRS	Specific model (NFC based context awareness applications)	General model (context awareness applications)
Context awareness (Demonstrative example)	User preferences	User localisation
Used verification Tool	BigMc	BigMc
properties Checked	Two properties (free deadlock, secure)	One property (free deadlock)
Max verification steps	100	20

Table 6: Comparison between works.

verify an alternative model by a probabilistic tool. We try also to explore specificities of interaction modes especially the one of emulation card.

References

- [1] Feng, T.-H., Hwang, M.-S. & Syu, L.-W (2016). *An Authentication Protocol for Lightweight NFC Mobile Sensors Payment*. Informatica, 27(4), pp.723–732. <http://dx.doi.org/10.15388/informatica.2016.108>.
- [2] Coskun, V. Ok, K. Ozdenizcin, B (2012). *Near Field Communication (NFC): From Theory to Practice*, John Wiely & Sons Ltd edition, 390 pages. Publisher. <http://dx.doi.org/10.1002/9781119965794>.
- [3] Mehmet N. Aydin. Busra Ozdenizci (2013). Design Science Perspective on NFC Research: Review and Research Agenda. Informatica 37(2), pp. 203-218. <http://www.informatica.si/index.php/informatica/article/viewFile/450/453>.
- [4] Borrego-Jaraba, Gonzalo Cerruela García, Irene Luque Ruiz and Miguel Ángel Gómez-Nieto (2013). *An NFC based context-aware solution for access to bibliographic sources in university Environments*, *Journal of Ambient Intelligence and Smart*

- Environments*, PP.105–118, 2013. Publisher. <http://dx.doi.org/10.3233/ais-120188>.
- [5] Danflous D (2012). *Billetique sur téléphone mobile à Nice: retour d'expérience, Report of CETE Méditerranée ALR* PP.35. <http://www.bv.transports.gouv.qc.ca/mono/1122337.pdf>.
- [6] Jianchao Luo. and Hao Feng (2015). *A Framework for NFC based Context-aware Applications*. International Journal of Smart Home Vol. 9, No. 1, pp. 111-122. Publisher. <http://dx.doi.org/10.14257/ijsh.2015.9.1.12>.
- [7] Schilit, B.N. Adams, N.I. and Want, R (1994). *Context-Aware computing Applications*, Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA). IEEE Press. Pp 85-90. <http://dx.doi.org/10.1109/wmcsa.1994.16>.
- [8] Prasad, R. Temdee, P. and Punnarumol (2018). *Context_ Aware communication and computing applications for smart environment*, Springer Series in Wireless Technology. Publisher. http://dx.doi.org/10.1007/978-3-319-59035-6_1.
- [9] Brown P.J., Bovey J.D and Chen X (1997). *Context-Aware applications*, From the Laboratory to the marketplace, IEEE Personal Communications, 4(5), 1997, pp. 58- 64. Publisher. <http://dx.doi.org/10.1109/98.626984>.
- [10] Brown P.J (1995). *The Stick-e Document: a framework for creating Context-aware applications*, Electronic Publishing, pp. 259-272. <http://cajun.cs.nott.ac.uk/compsci/epo/papers/volume8/issue2/2point1.pdf>.
- [11] Dey A.K. D. Salber and G.D. Abowd and M. Futakawa D. Futakawa, M. Gregory D (1999). combining context-awareness with wearable computing Third International Symposium on Wearable Computers. Pp99-23. Publisher. <http://dx.doi.org/10.1109/iswc.1999.806639>.
- [12] Dey, A. K (2001). *Understanding and using context, Personal and ubiquitous computing*, vol. 5, pp. 4-7. <http://dx.doi.org/10.1007/s007790170019>.
- [13] Feng, L (2017). *Context-Aware Computing*, Beijing China. Publisher. <https://doi.org/10.1515%2F9783110556674>.
- [14] Bauer, J (2003). *Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic*, Paper presented at Diplomarbeit. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.7895&rep=rep1&type=pdf>.
- [15] Milner, R (2005). *Axioms for bigraphical structure*. Technical Report UCAM-CL-TR-581, University of Cambridge. Mathematical Structures in Computer Science, 15(06), p.1005. Publisher. <http://dx.doi.org/10.1017/s0960129505004809>.
- [16] Milner, R (2008). *Bigraphs and Their Algebra*. Electronic Notes in Theoretical Computer Science, 209, pp.5–19. Publisher. <http://dx.doi.org/10.1016/j.entcs.2008.04.002>.
- [17] Jensen, O.H. & Milner, R (2003). *Bigraphs and transitions*. Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages- POPL '03. Available at: <http://dx.doi.org/10.1145/604131.604135>.
- [18] Dib, A.T.E., Barkaoui, K. & Sahnoun, Z(2016). Specification and verification of reconfigurable multi-agent system architectures. Multiagent and Grid Systems, 12(2), pp.105–124. Available at: <http://dx.doi.org/10.3233/mgs-160246>.
- [19] Perrone, G, S Debois, TT Hildebrandt(2012). A Model Checker for Bigraphs, Proceedings of the 27th Annual ACM Symposium on Applied Computing, Pp1320–1325. Publisher. <https://doi.org/10.1145/2245276.2231985>.
- [20] Birkedal, L. Debois, S. Elsborg, E. Hildebrandt, T. and Niss, H (2006). *Bigraphical Models of Context-aware Systems*, International Conference on Foundations of Software Science and Computation Structures (FoSSaCS.Vol 3921, Springer, Berlin, Heidelberg, pp 187-201. http://dx.doi.org/10.1007/11690634_13.
- [21] Wang, J.S (2011). *Formalizing the structure and behavior of context-aware systems in Bigraphs*’. First ACIS International Symposium on Software and Network Engineering, Pp. 89-94. <http://dx.doi.org/10.1109/ssne.2011.17>.
- [22] Lu, H. Chan, W.K, Tse, T (2006). *Testing Context Aware Middleware Centric Programs: a Data Flow Approach and an RFID Based Experimentation*. Paper presented at Special Interest Group Software engineering, (ACM SIGSOFT), USA, Pp. 242-252, <http://dx.doi.org/10.1145/1181775.1181805>.
- [23] Cherfia, T.A., Belala, F. and Barkaoui, K (2016). *A bigraph-based framework for specification and analysis of context-aware systems*, International Journal of Critical Computer-Based Systems (IJCCBS), Vol. 6, No. 4. Publisher. <http://dx.doi.org/10.1504/ijccbs.2016.081808>.
- [24] Chih-Hao Lin, Pin-Han Ho, and Hong-Chuan Lin (2014). *Framework for NFC-Based Intelligent Agents: A Context-Awareness Enabler for Social Internet of Things*, Paper Hindawi Publishing Corporation International Journal of Distributed Sensor Networks. Volume, Article ID 978951. <http://dx.doi.org/10.1155/2014/978951>.