

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR – ANNABA-
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة بادجي مختار – عنابة

Faculté de Technologie
Département d'Informatique

Année Universitaire : 2022/2023

Thèse

Présentée en vue de l'obtention du diplôme de Doctorat en sciences

Vérification formelle pour des applications sous NFC

Filière : Informatique

Option : Informatique Embarquée (INEM)

Par :

NABET Aïcha

Devant le Jury :

GHANEMI Salim	Pr.	Université Badji Mokhtar -Annaba	Président
BOUDOUR Rachid	Pr.	Université Badji Mokhtar -Annaba	Rapporteur
REDJIMI Mohamed	Pr.	Université 20 août 1955 -Skikda	Examineur
BENABDALLAH Ahcene Youcef	MCA.	Université 8 mai 1945- Guelma	Examineur

Dédicace

Je dédie humblement ce manuscrit à :

A tous ceux qui croient à la réussite par le travail et la persévérance,

A ma famille,

A mes amis.

Remerciements

En préambule à cette thèse, je remercie **ALLAH** qui m'a aidée et m'a donnée de la patience et du courage durant ces années d'études. Ces remerciements vont tout d'abord à mon encadreur Monsieur **BOUDOUR Rachid**, d'avoir accepté de m'encadrer dans ce travail de thèse. Nos échanges très efficaces, vos encouragements sont toujours venus à point. Je le remercie également, pour son orientation, sa confiance et sa patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené à bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité. Je voudrais remercier Monsieur **GHANEMI Salim**, Professeur à l'université d'Annaba de m'avoir fait l'honneur de présider mon jury. Je veux également exprimer toute ma gratitude à Monsieur **REDJIMI Mohamed**, Professeur à l'université de Skikda et à Monsieur **BENABDALLAH Ahcene Youcef**, Maître de conférences à l'université de Guelma qui ont eu la grande gentillesse de mettre leur savoir au service de mon travail en acceptant la tâche d'évaluation.

Grand Merci à Tous.

Résumé

La sensibilité au contexte est un domaine contemporain attirant l'attention de la communauté scientifique par sa capacité inouïe d'adaptation de comportement par rapport à des données spécifiques pouvant être tirées de l'environnement, formant un *contexte*. Cette technologie s'appuie sur la bonne perception de l'environnement. Near Field Communication 'NFC' offre un ensemble d'objets capables de jouer le rôle de collecteurs de contexte. L'intégration de la NFC dans ce domaine permet de créer un type particulier d'application appelé 'application sensible au contexte sous NFC'. La vérification formelle offre une pléthore d'approches et d'outils afin de prouver la correction de ce type de système dont ses applications ont en besoin. Dans la littérature, d'un côté la plupart des travaux essaient de construire un modèle général pour les systèmes sensibles au contexte en ignorant la spécificité de quelques applications. D'un autre côté les travaux ciblant la technologie NFC focalisent essentiellement sur l'utilisation de la vérification formelle afin de prouver l'efficacité d'un protocole de communication sous NFC. Ceci nous a mené à constater le manque flagrant des travaux supportant l'intégration des objets NFC dans un système sensible au contexte. De ce fait, cette thèse propose un nouveau modèle formel appelé BigNFC dédié aux applications sensibles au contexte sous NFC avec emphase sur la façon d'intégrer les objets NFC en mettant en avant le mode de communication dès le départ. BigNFC a été construit sur la base du formalisme Bigraphical Reactive System « BRS ». Initialement, on a établi un mappage entre les éléments BRS et les composants BigNFC, dont la structure est modélisée sous forme de bigraphe et le comportement étant réduit à un ensemble des règles d'interaction. Puis, on a expérimenté notre modèle sur un exemple de la vie courante avec quelques propriétés. Les résultats obtenus avec le vérificateur BigMC étant prometteurs.

Mots clé : Near field communication, application sensible au contexte sous NFC, Bigraphical Reactive System BRS, BigMC.

Abstract

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.

Keywords: Near Field Communication, context-aware application, Bigraphical Reactive System 'BRS', BigMC.

الملخص

إن التطبيقات المراعية للسياق هي تطبيقات متطورة قادرة على تكييف سلوكها على حسب تغيرات البيانات الواقعة في بيئتها. تعتمد هذه التقنية على الإدراك الجيد للبيئة، حيث تعتبر الـ NFC من التكنولوجيات الحديثة القادرة على لعب دور محوري لالتقاط البيانات المختلفة. ككل عمل تقني، يحتاج هذا النوع من التطبيقات الى الدقة حيث يوفر التحقق الرسمي عددًا كبيرًا من الأساليب والأدوات لإثبات صحة هذا النوع من التطبيقات. من جهة، إن معظم الأعمال تحاول بناء نموذج عام للأنظمة المدركة للسياق مع تجاهل خصوصية بعض التطبيقات. من ناحية أخرى، تركز جل الأعمال التي تستخدم تقنية الـ NFC بشكل أساسي على استخدام التحقق الرسمي لإثبات فعالية بروتوكول الاتصال فقط. يأتي هذا العمل لطرح إشكالية عن ماهية الطريقة الانجع لإدماج تقنية الـ NFC في التطبيقات المراعية للسياق مع مراعاة خصوصية هذه التطبيقات؟ من خلال هذه الأطروحة، نقترح نموذجًا رسميًا جديدًا يسمى BigNFC مخصصًا للتطبيقات التي تدرك السياق ضمن NFC والتي ركزنا منها على كيفية دمج تقنية الـ NFC مع مراعاة طبيعة الاتصال من البداية. لقد تم إنشاء BigNFC على أساس النظام التفاعلي BRS، حيث أنه في بداية، قدمنا خريطة تكافؤ بين عناصر BRS ومكونات BigNFC، ثم تم تصميم الهيكل على شكل بيغراف والسلوك عبارة عن مجموعة من قواعد التفاعل. أخيرًا، تحققنا من صحة نموذجنا من خلال تطبيقه على مثال و من ثم تحققنا من صحة بعض الخصائص بنجاح باستخدام لغة BigMC.

الكلمات الرئيسية: تقنية الـ NFC، التطبيقات المراعية للسياق، البيغراف، النظام التفاعلي 'BRS'، BigMC،

Table des Matières

Dédicace	
Remerciement	
Résumé	
Abstract	
الملخص	
Table des Matières.....	7
Liste des figures	10
Liste des tableaux	12
Liste des acronymes	13
Introduction générale.....	14
1 Contexte de la thèse.....	15
1.1 Domaine de travail.....	15
1.2 Motivation	16
2 Problématique scientifique et objectifs de recherche	17
3 Organisation du manuscrit.....	18
CHAPITRE 1 Systèmes Sensibles au Contexte	20
1.1 Introduction	21
1.2 Qu'est-ce qu'un contexte ?	21
1.2.1 Contexte dans la littérature	21
1.2.2 Discussion	22
1.3 Classification de contexte.....	23
1.3.1 Synthèse	26
1.4 Caractéristiques des informations contextuelles	27
1.5 Systèmes sensibles au contexte.....	28
1.5.1 Discussion	28
1.6 Méthodes de représentation du contexte	29
1.6.1 Méthode clé-valeur	29
1.6.2 Méthodes graphiques.....	29
1.6.3 Méthode basée ontologie	30
1.7 Architecture en couches d'un système sensible au contexte	31
1.8 Utilisation du contexte avec l'évolution du téléphone mobile	33
1.9 Conclusion.....	34
CHAPITRE 2 ETAT De L'Art Sur Les Applications NFC	35
2.1 Introduction	36
2.2 Partie Physique.....	37
2.2.1 Définition.....	37
2.3 Avantages de la technologie NFC.....	40

2.3.1	Sécurité.....	40
2.3.2	Simplicité et souplesse.....	40
2.3.3	Compatibilité avec les infrastructures RFID	41
2.3.4	Coût faible	41
2.4	Inconvénients	41
2.4.1	Dépendance d'opérateur mobile	41
2.4.2	Peur des utilisateurs	41
2.4.3	Attaques.....	41
2.5	NFC dans le téléphone mobile.....	42
2.6	Normes et compatibilité.....	45
2.7	Applications NFC.....	50
2.7.1	Qu'est-ce qu'une application NFC mobile ?.....	51
2.7.2	Taxonomie des applications NFC.....	51
2.7.3	Conception des applications NFC.....	54
2.7.4	Fondements théoriques	55
2.7.5	Défis de développement	56
2.8	Conclusion.....	57
CHAPITRE 3 Formalisme Bgraphical Reactive System 'BRS'		58
3.1	Introduction	59
3.2	Modélisation formelle d'un système.....	59
3.2.1	Modélisation de système en utilisant les BRS.....	60
3.2.1.1	Définition de Bigraphe.....	60
3.2.2	Types de bigraphes	62
3.2.2.1	Bigraphe de base.....	62
3.2.2.2	Bigraphe d'ion discret.....	62
3.2.2.3	Bigraphe premier	63
3.2.2.4	Bigraphe discret.....	63
3.2.3	Algèbre et la forme normale discrète d'un bigraphe	63
3.2.4	Exemples d'un Bigraphe.....	65
3.2.5	Définition Bigraphical reactive system BRS.....	67
3.2.5.1	Composition	68
3.2.5.2	Transformation	68
3.3	BRS et la modélisation des systèmes sensibles au contexte ?.....	69
3.4	Travaux existants utilisent les BRS pour la sensibilité au contexte	70
3.4.1	Synthèse	71
3.5	Vérification formelle	71
3.5.1	Vérification de modèle (model-Checking).....	72
3.5.2	Différents outils de vérification pour les BRS	72
3.6	Conclusion.....	73

CHAPITRE 4 Conception et démonstration du modèle BigNFC.....	75
4.1 Introduction	76
4.2 Démarche de travail.....	76
4.3 Modélisation d'une application sensible au contexte sous NFC	77
4.4 Comment construire BigNFC ?	79
4.4.1. Equivalence entre le modèle BigNFC et les composants BRS	81
4.4.2. Passage vers le modèle de vérification avec BigMC.....	82
4.5 Exemple démonstratif.....	82
4.5.1 Spécification du système	82
4.5.2 Construction du modèle de validation en utilisant le BigNFC	83
4.5.2.1 Passage vers le modèle à base BRS	83
4.6 Conclusion.....	89
CHAPITRE 5 Vérification formelle du modèle BigNFC	91
5.1 Introduction	92
5.2 Approche de vérification	92
5.3 C'est quoi le langage BigMC ?.....	93
5.3.1 Structure d'un programme BigMC	94
5.3.2 Termes du langage BigMC	95
5.3.2.1 Contrôles	95
5.3.2.2 Noms	95
5.3.2.3 Règles de réaction.....	95
5.3.2.4 Définition de modèle	96
5.3.2.5 Définition des propriétés.....	96
5.4 Validation de l'approche	97
5.4.1 Transformation de l'exemple démonstratif en programme BigMC	98
5.4.1.1 Programme BigMC de la zone de production	98
5.4.1.2 Programme BigMC de la zone de consommation	99
5.4.1.3 Programme BigMC des règles d'interaction	99
5.4.1.4 Programme BigMC du modèle général.....	102
5.4.1.5 Définition des propriétés avec BigMC.....	102
5.5 Résultats de la vérification formelle par BigMC	102
5.6 Discussion	105
5.7 Conclusion.....	107
Conclusion générale	108
1 Bilan général de la thèse	109
2 Perspectives et travaux futurs	109
Bibliographie.....	111

Liste des figures

Figure 1 - Schéma global de la thèse.....	19
Figure 1.1 - Evolution chronologique de la notion de contexte.....	23
Figure 1.2 - Classification du contexte	26
Figure 1.3 - Proposition de Classification du contexte.....	27
Figure 1.4 - Architecture globale en couches pour un système sensible au contexte.....	32
Figure 2.1 - NFC vs RFID.....	38
Figure 2.2 - Mode de fonctionnement RFID.....	39
Figure 2.3 - Acteurs nécessaires pour établir une communication sous NFC.....	39
Figure 2.4 - Cycle de vie d'un SE.....	43
Figure 2.5 - Composants d'un téléphone portable NFC.....	43
Figure 2.6 - Couche protocolaire NFC.....	46
Figure 2.7 - Pile protocolaire du mode P-2-P.....	47
Figure 2.8 - Pile protocolaire du mode émulation de carte.....	48
Figure 2.9 - Pile protocolaire du mode lecture.....	49
Figure 2.10 - Classification des applications NFC selon domaines et modes.....	53
Figure 3.1 – Bigraphe d'ion discret $K_{\vec{x}}$	62
Figure 3.2 - Bigraphe premier.....	63
Figure 3.3 - Forme graphique d'une clôture et d'une substitution.....	64
Figure 3.4 - Opérations de base sur un bigraphe.....	65
Figure 3.5 - Exemple d'un Bigraphe.....	66
Figure 3.6 - Bigraphe G avec son graphe de places et de liens.....	67
Figure 3.7 - Composition ($H \circ F$).....	68
Figure 3.8 - Règle de réaction.....	69
Figure 4.1 - Stratégie de travail.....	77
Figure 4.2 - Schéma simplifié des formes et des phases de traitement.....	78
Figure 4.3 - Architecture générale d'un système sensible au contexte sous NFC.....	80
Figure 4.4 - Etat initial de la zone de production.....	84
Figure 4.5 - Bigraphe de l'état initial de la zone de consommation.....	85
Figure 4.6 - Règle d'interaction : action touchez tag.....	86
Figure 4.7 - Règle d'interaction : interception du contexte brut.....	87
Figure 4.8 - Règle d'interaction : déduction des préférences.....	88
Figure 4.9 - Règle d'interaction : livres recommandés.....	89
Figure 5.1 - Processus de vérification de notre approche.....	93

Figure 5.2 - Structure générale d'un programme BigMC.....	94
Figure 5.3 - Grammaire de BigMC	95
Figure 5.4 - Programme BigMC de la zone de production.....	98
Figure 5.5 - Programme BigMC de la zone de consommation.....	99
Figure 5.6 - Résultat de vérification de la propriété absence d'interblocage sur une partie du modèle.....	103
Figure 5.7 - Résultat de vérification après quatre-vingt pas.....	103
Figure 5.8 - Résultat de vérification de la propriété absence d'interblocage sur le modèle entier (cent pas).....	104
Figure 5.9 - Résultat vérification de deuxième propriété sur le modèle entier (vingt pas).....	104

Liste des tableaux

Tableau 1.1 – Classes du contexte.....	25
Tableau 1.2 - Comparaison entre les approches de représentation du contexte.....	31
Tableau 1.3 - Types des capteurs.....	32
Tableau 2.1 - Exemples sur des domaines exploitant NFC.....	40
Tableau 2.2 - Composants d'un NFC phone.....	44
Tableau 2.3 - Description des couches protocolaires du mode P2P.....	47
Tableau 2.4 - Types Tag de NFC forum.....	49
Tableau 2.5 - Description des couches protocolaires du mode lecture / écriture.....	50
Tableau 2.6 - Classification des applications NFC selon les modes de communication.....	52
Tableau 2.7 - Classification des applications NFC selon domaines et modes.....	53
Tableau 3.1 - Liste des symboles mathématiques.....	61
Tableau 3.2 - Outils de BRS.....	73
Tableau 4.1 - Equivalence entre le modèle conceptuel et les composants BRS.....	81
Tableau 4.2 - Equivalence entre la zone de production et les composants du bigraphe.....	84
Tableau 4.3 - Equivalence entre la zone de consommation et les éléments du bigraphe.....	85
Tableau 5.1 - Comparaison entre BigNFC.....	106
Tableau 5.2 - Comparaison entre BigMC	106

Liste des acronymes

BigMC	Bigraph Model Checker
BRS	Bigraph Reactive System
CE mode	Card Emulation mode ou mode émulation de carte
CHI	Contrôleur Hôte Interface
CML	Context Modeling Language
CTL	Computation Tree Logic
GPM	General profile Model
LLCP	Logical Link Control Protocol
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
ORM	Object Role Modeling
OWL	WEB Ontology Language
P-2-P mode	Peer to Peer mode ou mode point à point
R/W mode	Read/ Write mode ou mode lecture/ écriture
RDF	Resource Description Framework
RFID	Radio Frequency Identification
RTD	Record Type Definition
SE	Secured Element ou Élément Sécurisé
SWP	Single Wire Protocol
TES	Transaction Electronique Sécurisée

Introduction générale

Sommaire

1. Contexte de la thèse
 - 1.1. Domaine de recherche
 - 1.2. Motivation
 2. Problématique scientifique, et objectifs de recherche
 3. Organisation du manuscrit
-

1. Contexte de la thèse

1.1. Domaine de travail

Chaque jour, la nouvelle technologie nous surprend, avec plus de précision, de souplesse et de puissance où elle se rapproche grave à la fiction, bel et bien tout ça a eu un point de départ. En 1988, Wisser a marqué son nom comme l'un des pionniers de l'informatique pervasive nommée aussi informatique ubiquitaire qui consiste à créer un ensemble de dispositifs qui communiquent entre eux afin de pouvoir créer un environnement plus interactif, adaptatif, adaptable indépendamment de la contrainte temporelle et spatiale pour enrichir l'expérience utilisateur et le satisfaire. En effet, cette informatique contemporaine fait appel à plusieurs d'autres disciplines par exemple la technologies de communication et de réseau sans fil, à l'informatique mobile, aux systèmes embarqués, à la technologie sans fil (aux étiquettes d'identification par radiofréquence 'RFID', communication en champs proche 'NFC'), aux logiciels intermédiaires, la reconnaissance vocale, l'intelligence artificielle (IA), la sensibilité au contexte, ...

La sensibilité au contexte est l'une des disciplines qui attire la communauté scientifique, son objectif principal est de créer un système capable d'adapter son comportement en respectant des données spécifiques nommées '*contexte*'. Ce concept très large peut signifier l'ensemble des données venant de l'environnement comme la localisation, la température, taux d'humidité, ... et même l'ensemble des données qui peuvent qualifier l'utilisateur par exemple son profil.

En effet, plusieurs objets peuvent participer à la création de cette expérience tels que les objets NFC qui peuvent être des collecteurs de contexte d'une façon directe à partir de l'environnement ou indirecte en passant par une phase de déduction. Dans cette thèse, pourquoi on s'intéresse particulièrement à la technologie NFC ?

Plus loin de la réalité, NFC permet de créer un pont entre la réalité et l'imaginaire. NFC est une technologie sans contact offrant une communication sécurisée entre deux équipements NFC (téléphone, lecteur, tag) sur une distance qui ne dépasse pas les dix centimètres. Elle fonctionne sous la fréquence de 13.56 MHz avec un taux de transfert des données de 106 kbit/s, 212 Kbit/s et 424 Kbit/s. Afin de pouvoir communiquer entre deux dispositifs NFC, trois modes de communication ont été offerts :

- **Mode lecture/écriture**, permettant au téléphone NFC de lire/écrire des données depuis/vers un tag NFC ou RFID ;

- **Mode Peer-to-Peer**, offrant à deux mobiles NFC la possibilité d'échanger des informations comme par exemple un enregistrement des contacts, un message textuel, ... ;
- **Mode émulation de carte**, rendant le téléphone mobile capable de fonctionner comme une carte à puce sans contact.

Cette technologie intuitive, souple et simple allèche les utilisateurs ainsi les investisseurs. Cependant, il demeure toujours une méfiance à son égard. Peut-on renforcer le lien entre cette technologie et ses utilisateurs ?

Depuis toujours, la vérification formelle a été un choix puissant pour rajouter la rigueur à un travail scientifique et assurer sa validité en offrant une multitude de méthodes mathématiques, techniques et outils. Historiquement, l'utilisation des méthodes mathématiques a été recommandée pour construire des systèmes sensibles tels que les réacteurs nucléaires dans le but d'éviter des dégâts non souhaitables. Aujourd'hui, elle est utilisée partout quel que soit la nature, la complexité et même la sensibilité des systèmes. En 2001, Milner [1] a proposé un nouveau formalisme qu'il a baptisé Bigraph Reactive System 'BRS', qui permet de modéliser à la fois la connectivité et la localisation des entités formant un système. Actuellement, le BRS est utilisé dans plusieurs domaines par exemple la sensibilité au contexte, la bio-informatique, etc.

Comment peut-on se servir du formalisme BRS et comment l'incorporer dans la modélisation des applications sensibles au contexte sous NFC ? La réponse à cette question constitue bien la thématique de cette thèse.

1.2. Motivation

Covid 19 a entraîné le monde entier dans un état de consternation, de terreur et même d'angoisse ce qui a obligé les autorités de prendre des mesures de protection pour diminuer tant bien que mal son ampleur par exemple : distanciation, vaccination, confinement, etc. Véritablement ces mesures ont eu une influence négative sur l'économie mondiale. Afin de pouvoir canaliser voire réduire la transmission du virus, la Chine comme tant d'autres pays a adopté la technologie sans contact comme un moyen efficace permettant à la population d'interagir. Par conséquent le marché de ces technologies a pris une grande ampleur, et actuellement, la technologie NFC captive l'attention des grandes compagnies dans plusieurs domaines potentiels : En août 2021, google a annoncé le lancement de sa nouvelle sécurité Titan USB-C NFC clé [2], qui va remplacer son ancienne clé Titan Bluetooth dans laquelle il a été

intégré la technologie NFC pour offrir plus de sécurité, de facilité, et même plus de rapidité à ses anciens modèles. Ces clés augmentent la confidentialité par l'utilisation d'une seconde barrière d'authentification, par exemple même si quelqu'un s'est procuré le mot de passe de PC, la session reste verrouillée jusqu'à l'utilisation de la clé de sécurité Titan C pour vérifier son identité. A l'occasion du salon électronique Consumer Electronics Show 'CES' à Las Vegas en 2020, la compagnie Française Netatmo [3] a présenté des clés intelligentes qui utilisent la technologie NFC ; cette clé reprogrammable se montre comme étant un grand atout dans le monde des smart houses, elle permet d'ouvrir plusieurs serrures par une seule clé (bureau, maison, ...) puis en cas de perte il suffit d'annuler l'accès de la clé perdue. En 2020, CarKey [4] a été annoncé comme étant un API permettant de transformer un Iphone ou Apple Watch en une clé NFC de voiture. Cette invention va rendre les choses plus simples tout en offrant une solution plus sécurisée.

2. Problématique scientifique et objectifs de recherche

La sensibilité au contexte fait référence à la capacité d'un système de détecter son environnement et à modifier son comportement pour fournir des services adaptés. Avoir de tels systèmes associés à la puissante technologie NFC, ouvre de nouvelles perspectives et de nouveaux domaines de recherche, permettant à un type d'applications très utiles connues sous le nom des applications sensibles au contexte sous NFC.

Ces systèmes doivent faire preuve de sûreté de fonctionnement et de validation à cause des données manipulées, par conséquent l'utilisation d'une approche d'analyse exhaustive telle que la vérification formelle est plus que nécessaire. Mais existe-il un modèle formel pour les applications sensibles au contexte sous NFC ?

Dans la littérature, la plupart des travaux se concentrent sur la création d'un modèle général pour les systèmes sensibles au contexte en ignorant la spécificité de certaines applications telles que les applications NFC où elles présentent une plus grande complexité. Nous soulignons l'existence de peu ou pas de travaux dans ce domaine supportant la modélisation, et même la majorité des travaux ayant pris la NFC comme champs d'étude se focalisent essentiellement sur la validation formelle des protocoles de communication sous NFC mais est ce que ces travaux ont abordé l'intégration de la NFC lors de la modélisation de l'application et plus précisément lors de modélisation d'une application sensible au contexte sous NFC ?

Dans cette thèse, nous proposons BigNFC comme un nouveau modèle formel basé sur le formalisme BRS en tenant compte du mode d'interaction dès le début. Notre stratégie peut être scindée en trois phases, d'abord, nous commençons par établir une équivalence entre les composants BRS et le modèle BigNFC [5], où la structure est modélisée sous forme de bigraphe et le comportement exprimé comme étant des règles d'interactions tout en l'appliquant sur un exemple puis nous proposons un passage entre le modèle décrit en BRS et le langage de validation BigMC et enfin nous définissons un ensemble de propriétés qui ont été vérifiées avec succès sur l'ensemble du programme.

3. Organisation du manuscrit

Cette thèse comporte quatre parties essentielles :

Partie 1 : Nous commençons notre thèse par une introduction générale, qui contient la description générale de notre domaine de travail, notre motivation, une présentation de notre problématique ainsi que nos objectifs de recherche ;

Partie 2 : c'est une partie déclinée à l'état de l'art, elle renferme trois chapitres :

- **Chapitre 1 :** contient l'ensemble des notions spécifiques liées directement à la notion de contexte, de sensibilité au contexte, ...
- **Chapitre 2 :** se consacre à la description de la technologie NFC, les normes, les applications NFC, ...
- **Chapitre 3 :** se dédie à la présentation de la vérification formelle, les bigraphes, le formalisme BRS, ...

Partie 3 : c'est la partie représentant le cœur de notre travail, dans laquelle nous essayons de satisfaire les deux objectifs signalés dans la première partie en lui consacrant deux chapitres :

- **Chapitre 4 :** contient la description complète du modèle BigNFC en utilisant le formalisme BRS ainsi que son application sur un exemple, ...
- **Chapitre 5 :** est réservé à la vérification et validation du modèle BigNFC en utilisant le langage BigMC.

Partie 4 : nous achevons notre thèse une conclusion générale, où nous fournissons un bilan global sur la thèse puis nous dévoilons quelques perspectives de nos travaux futurs. Dans ce qui suit nous donnons un schéma global de la thèse.

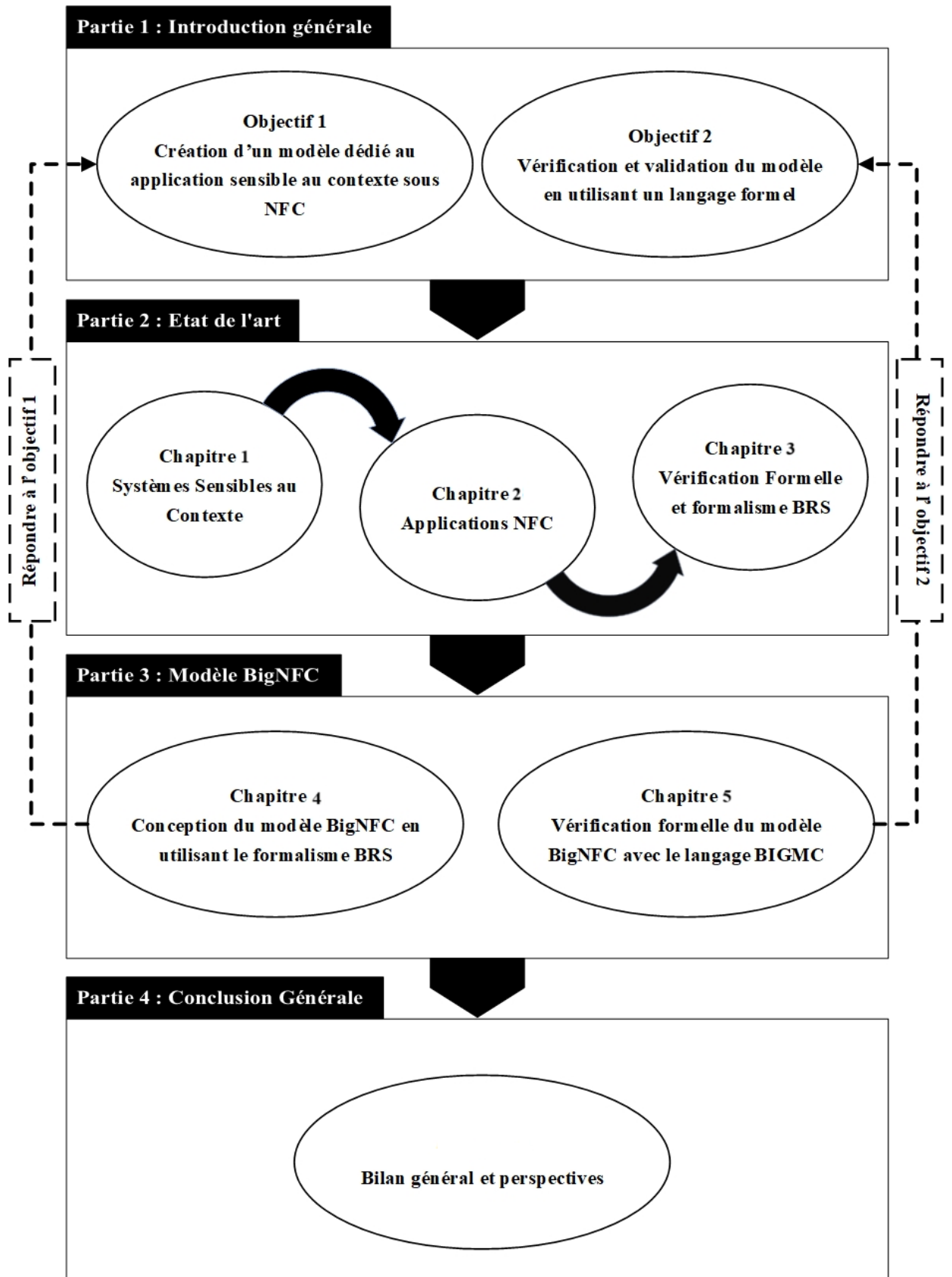


Figure 1 -Schéma global de la thèse

CHAPITRE 1

Systèmes Sensibles au Contexte

Sommaire

- 1.1 Introduction
 - 1.2 Qu'est-ce qu'un contexte ?
 - 1.2.1 Contexte dans la littérature
 - 1.2.2 Discussion
 - 1.3 Classification de contexte
 - 1.4 Caractéristiques des informations contextuelles
 - 1.5 Systèmes sensibles au contexte
 - 1.6 Méthodes de représentation du contexte
 - 1.6.1 Méthode clé-valeur
 - 1.6.2 Méthode graphique
 - 1.6.3 Méthode basée ontologie
 - 1.7 Architecture à couches d'un système sensible au contexte
 - 1.8 Utilisation du contexte avec l'évolution de téléphone mobile
 - 1.9 Conclusion
-

1.1. Introduction

L'origine du mot contexte vient du latin *contextus* qui signifie assemblage, réunion, enchaînement et de façon linguistique est l'ensemble qui forme par une liaison naturelle les différentes parties d'un texte ou d'un discours. De plus, chaque domaine de travail impose sa définition du mot contexte. Par exemple en sociologie, le contexte est l'un des facteurs de la communication qui influe sur le sens d'un message et sur sa relation à l'autre partie du message. Il correspond aussi à l'environnement dans lequel la communication a lieu, avec différentes perceptions de l'environnement général qui peuvent être associées à la communication. Ainsi, le contexte est le "cadre" de perception à travers lequel on émet ou on reçoit un message.

En informatique, ce terme a plusieurs significations. Des chercheurs le considèrent comme étant une information sur l'environnement d'un système informatique, d'autres ajoutent à cette définition le temps, les entités influençant le comportement de système, ... A la fin, nous ajoutons que trouver une définition complète et claire pour ce terme contexte s'avère une mission complexe. Dans ce qui suit, nous présentons les différents points de vue.

1.2. Qu'est-ce qu'un contexte ?

La compréhension du contexte a progressivement évolué dans les définitions existantes en littérature. Dans ce qui suit, nous dressons un panorama sur cette notion.

1.2.1. Contexte dans la littérature

- D'après **Schilit** [6], le contexte est la capacité de répondre aux questions : « Où es-tu ? », « Avec qui es-tu ? », « De quelle ressource de proximité disposes-tu ? », en conséquence, le contexte représente les changements d'environnement physique, d'utilisateur et du calcul (computationnel). Cette définition a été reprise par **Pascos** [7] puis par **Dey** [8] ;
- **Brown** [9], a proposé une nouvelle définition en réduisant les éléments de contexte aux éléments d'environnement et d'utilisateur, c'est-à-dire le contexte est « la localisation, l'identité des personnes qui entourent l'utilisateur, l'heure, la saison et la température, ... ». D'ailleurs d'autres auteurs ont exploité sa définition en ajoutant explicitement la notion du temps (définition **Rayan** [10]) ;
- **Pelc** [11], a conclu que le contexte représente l'ensemble des états possibles d'un environnement d'une application ;

- **Pascoe** [12], retrouve qu'il est nécessaire d'ajouter à la notion d'état, la notion de pertinence. Cela signifie que « le contexte est un sous ensemble des états physiques et conceptuels ayant un intérêt pour une entité particulière » ;
- **Dey** [13] précise la nature des entités en question « le contexte couvre toutes les informations pouvant être utilisées pour caractériser la situation d'une entité. Selon lui, une entité peut être une personne, un lieu ou un objet et ainsi elle peut être pertinente pour l'interaction entre l'utilisateur et l'application ». Cette définition résulte des définitions précédentes. En effet, les caractères du contexte peuvent émaner à l'utilisateur par exemple où un utilisateur été détecté par les capteurs installés directement dans l'environnement ? ;
- **Chaari** [14], propose une définition d'interaction de contexte, reposant sur le triplet <plateforme–environnement–utilisateur> où l'environnement est l'ensemble des entités (objets, personnes et événements), avec des périphériques agissant sur la tâche courante et pouvant avoir un impact sur le comportement du système ou de l'utilisateur ;
- **Miraoui** [15], propose une définition faisant abstraction à la notion de contexte lui-même. Mais il le considère du point de vue service offert par le système c'est-à-dire la trace finale d'un changement de contexte. Ils donnent une explication sur les informations contextuelles : « Toute information dont le changement de sa valeur déclenche un service ou une modification de la qualité ou de la forme d'un service ».

1.2.2. Discussion

En synthétisant ces différentes visions, nous affirmons que le contexte est une notion complexe, ce qui explique cette pléthore des travaux tentant de l'appréhender. Au début des années quatre-vingt-dix, le contexte a été réduit à deux notions : la localisation et le temps, cette restriction peut être expliquée par le style des applications réalisées. A la fin des années quatre-vingt-dix, d'autres pièces complétant le puzzle ont été rajoutées c'est-à-dire que le contexte a été élargi à l'environnement, les entités et leur statut émotionnel. Quant aux années deux mille, le contexte a connu une nouvelle variation en rajoutant la manière dont il a été conçu ainsi que la notion du contexte social, etc. La figure 1.1 illustre l'évolution chronologique de la notion de contexte.

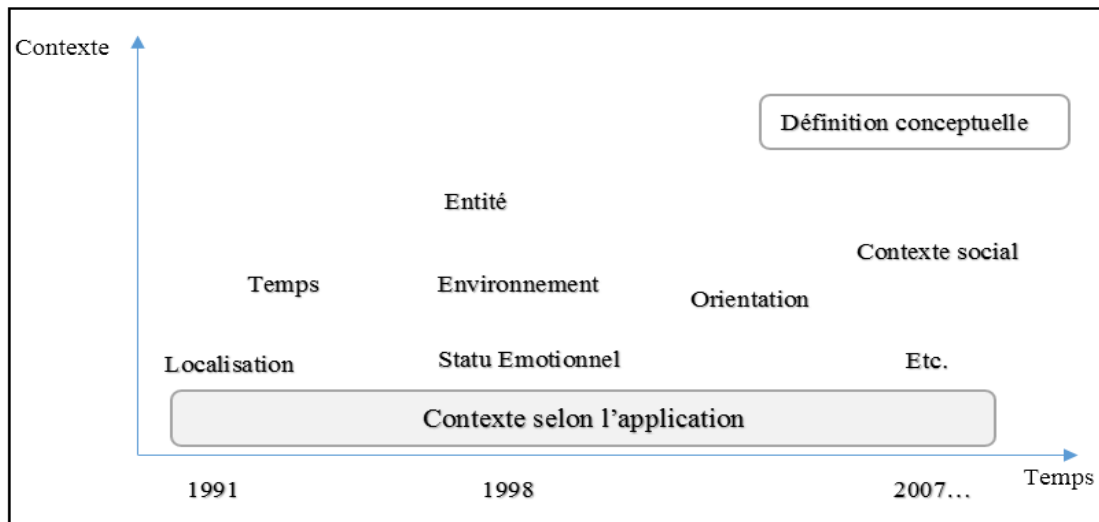


Figure 1.1 - Evolution chronologique de la notion de contexte

Après une recherche approfondie dans la littérature, nous pouvons avancer que le contexte est l'ensemble des informations associées à l'utilisateur influençant directement le comportement global du système c'est-à-dire que le contexte est l'ensemble des informations ayant une relation directe avec l'utilisateur (son profil, sa position, ses interactions, etc.) et des informations indirectes (complémentaires) pouvant être attribuées à l'utilisateur (avis d'autres utilisateurs interagissant dans le système) autrement dit qu'une information indirecte est le résultat d'une phase de déduction permettant d'avoir une information contextuelle significative.

1.3. Classification de contexte

Le contexte est une notion ancienne dans l'univers de l'informatique, restée longtemps non cernée. C'est la raison principale qui a fait diverger les chercheurs sur une classification précise. Cette divergence pourrait découler du domaine de travail de chacun d'eux. Dans ce qui suit, nous essayons de rassembler les différents types de contexte qui sont déjà cités dans la littérature avec la mise en avant des divers critères de classification :

- **Ryan** [16], présente une classification basée sur trois notions : l'identité de l'utilisateur, les ressources proches disponibles de l'environnement ainsi que la période (temps) d'exécution ou d'interaction ;
- **Rodden** [17], propose une catégorisation basée sur les notions suivantes : infrastructure, localisation de dispositif mobile, application, sans oublier les autres paramètres physiques ;
- **Not** [18], met en avant une nouvelle classification axée sur deux classes :

Contexte social, contenant les aspects sociaux d'un utilisateur tel que la relation entre individus, ...

Contexte matériel, par exemple la localisation, la machine, la plateforme existante, ...

- **Chen** [19], énoncent une classification préconisant que le contexte peut être séparé en deux classes :

Contexte actif, est l'ensemble des informations contextuelles influençant le comportement du système ;

Contexte passif, est l'ensemble des informations nécessaires au système qui ne sont pas réellement critiques.

- **Gwizdka** [20], suggère une classification qui décompose le contexte en un contexte interne et externe par exemple : un contexte interne est l'état interne de l'utilisateur et un contexte externe est l'état global de l'environnement ;
- **Hofer** [21], donne une classification qui sépare entre le contexte physique qui représente les informations capturées par les capteurs installés dans l'environnement et le contexte logique contenant les informations sur les interactions de l'utilisateur telles que ses activités, ses émotions et ses buts.
- **Kirsch Pinheiro** [22], a retenu le nombre de classes cité par les autres auteurs en distinguant entre le contexte physique (paramètres utilisateur) et le contexte organisationnel (rôles, activités, calendrier utilisateur) ;
- **Abhur** [23], a exhibé une catégorisation plus globale où le contexte est divisé en six classes : contexte utilisateur, contexte physique, contexte réseau, contexte activité, contexte matériel, contexte service. Le tableau 1.1 présente les six classes de contexte.

Chapitre 1 : Systèmes Sensibles au Contexte

Classe	Sémantique	Exemples
Contexte utilisateur	Qui ?	Profil d'utilisateur : identification, relations avec les autres, listes des choses à faire, etc.
Contexte physique	Où ?	Environnement physique : humidité, température, niveau de bruit, etc.
Contexte réseau	Où ?	Environnement réseau : connectivité, bande passante, protocole, etc.
Contexte activité	Ce qui se passe, quand ?	Ce qui se passe, à quel moment : entrer, sortir, etc.
Contexte matériel	Que peut-on utiliser ?	Profil et activités des dispositifs : identifications, emplacement, durée de vie de la batterie, etc.
Contexte service	Que peut-on obtenir ?	Informations sur les fonctions qui peuvent être fournies par le système : format de fichier, affichage, etc.

Tableau 1.1 - Classes du contexte [23]

- **Cherfia** [24], propose une nouvelle catégorisation en séparant le contexte en trois classes : contexte utilisateur, contexte informatique et contexte d'environnement. La figure 1.2 montre les trois classes de contexte proposé par [24].

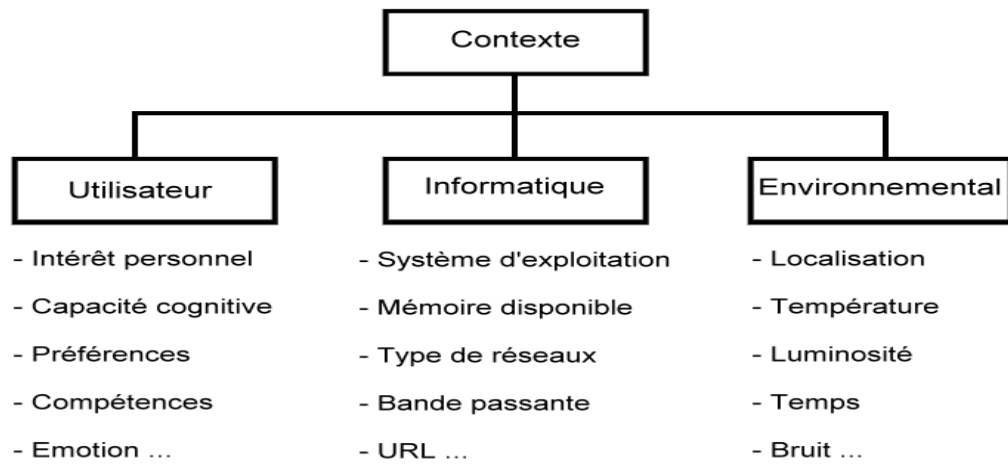


Figure 1.2 – Classification du contexte [24]

- **Lin** [25], propose une classification sur les sources de contexte, scindées en deux classes :

Contexte personnel, contient toutes les informations contextuelles concernant l'utilisateur à l'instar : profil utilisateur, profil social, profil d'activité, ...

Contexte d'environnement, inclue la sensibilité au contexte à la localisation, le temps, l'état des composants, ...

- **Henricksen et Indulska** [26], ont imaginé une nouvelle classification qu'ils ont appelé une catégorisation opérationnelle, basée sur la manière dont le contexte est manipulé. Ici le contexte est classé en deux classes : contexte statique et contexte profilé.

1.3.1. Synthèse

Suite à ce parcours panoramique sur les différentes catégorisations existantes dans la littérature, nous avons constaté qu'il existe toujours des chevauchements entre les classes. En effet, il existe toujours des informations interdépendantes empêchant ces travaux de trouver une classification unique et complète. Nous proposons une classification où nous tenons compte la source fournissant le contexte, de ce fait, nous le scindons en trois classes :

- **Contexte utilisateur**, regroupe les informations statiques recueillies autour de l'utilisateur ;
- **Contexte environnement**, concerne l'état de l'environnement autour de l'utilisateur ;
- **Contexte intelligent**, déduit ou inféré par le système. Ce type de contexte peut ne pas être lié directement à l'utilisateur.

La figure 1.3 décline cette classification.

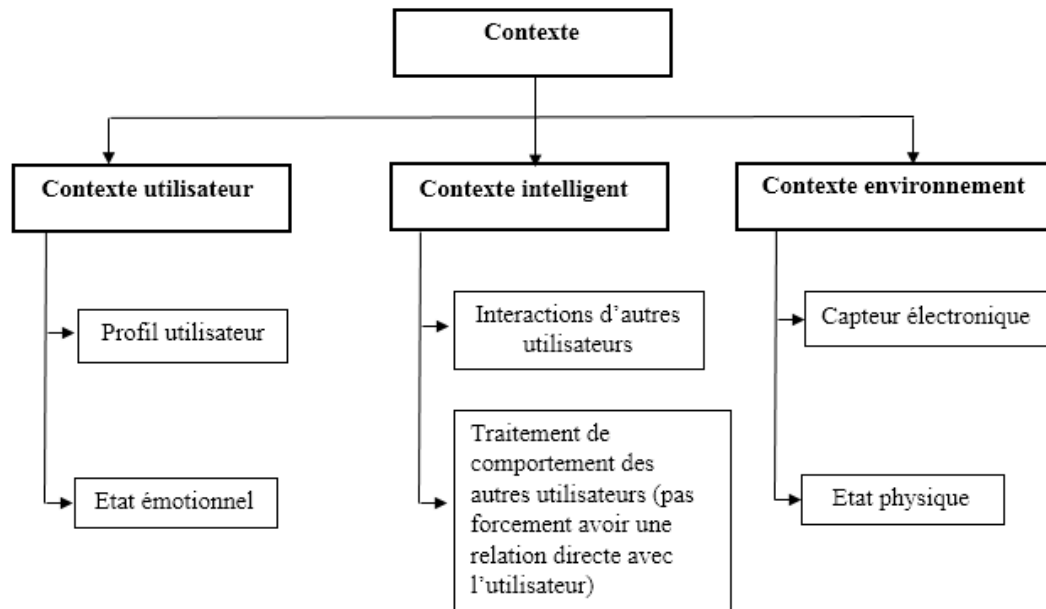


Figure 1.3 - Proposition de classification du contexte

Cette classification permet aux concepteurs des applications sensibles au contexte de cerner plus clairement les différentes sources d'informations contextuelles en distinguant la source délivrant le contexte en déterminant les deux classes du contexte (utilisateur, environnement) et celle qui le traite en déterminant la classe du contexte intelligent.

Une information contextuelle peut se voir attacher plusieurs caractéristiques, que nous allons aborder ci-dessous.

1.4. Caractéristiques des informations contextuelles

Les informations contextuelles ont des traits spécifiques. Elles sont [27] :

- **Incertaines** ; Dans un système pervasif, l'environnement est dynamique d'une façon permanente ce qui rend les informations contextuelles à un instant donné obsolètes, incohérentes ou incomplètes. Le problème d'incertitude a été pris en compte dans plusieurs papiers ;
- **Hétérogènes** ; le résultat logique d'une source de perception d'informations de différentes natures (capteurs physiques, capteurs logiciels, ...) qui peut causer des problèmes d'interprétation et de sens ;
- **Interdépendantes** : elles se manifestent dans le changement de valeur d'une variable contextuelle influant sur la valeur d'une autre variable en ayant pour conséquence un changement du comportement de système.

Dans ce qui suit, nous exhibons d'une façon approfondie la notion de sensibilité au contexte qui est une notion large et complexe.

1.5. Systèmes sensibles au contexte

Le concept sensibilité au contexte « context-awareness » a été présenté pour la première fois en 1995 dans les travaux de **Shilit** [28]. Il est l'un des pionniers à avoir considéré que les applications sont dotées d'un ensemble de mécanismes capables de changer, d'adapter leur comportement dynamiquement en prenant en compte les informations contextuelles. D'après **Li** [29], montre qu'une application sensible au contexte est une application capable de collecter, interpréter et répondre aux exigences d'environnement et d'utilisateur. **Dey** [30], donne une autre définition la plus répandue dans ce domaine. Il énonce qu'un système est sensible au contexte si et seulement s'il « utilise le contexte afin de fournir une information pertinente à l'utilisateur ». Cette pertinence dépend de la tâche utilisateur. A son tour **Miraoui** [31], croit qu'un système sensible au contexte a toujours l'aptitude de changer automatiquement ses services ou déclencher d'autres services en répondant aux changements des valeurs d'information qui caractérisent le service. Selon **Rohn** [32], un système sensible au contexte :

- **Adaptatif** ; doit être pourvu de la capacité de reconnaître les préférences de l'utilisateur et de s'accorder et s'arranger en conséquence ;
- **Réceptif** ; doit disposer du pouvoir de prédire les besoins utilisateur dans un environnement évolutif ;
- **Proactif** ; doit avoir des objectifs à atteindre. Pour y arriver, des initiatives sont à entreprendre ;
- **Autonome** ; doit avoir la capacité d'agir sans aucune intervention de l'utilisateur.

1.5.1. Discussion

La compréhension des systèmes sensibles au contexte résulte essentiellement à la bonne assimilation des deux notions basiques : contexte et sensibilité. Dans la littérature, la sensibilité se décline comme étant l'aptitude d'un organisme à réagir à des excitations internes et externes, alors nous pouvons dire qu'un système sensible au contexte a l'aptitude de réagir formellement et clairement à des excitations internes (changement d'état interne de ses composants physiques, de ses interactions, etc.) et externes (environnement).

1.6. Méthodes de représentation du contexte

Lors de la création d'un système sensible au contexte, le passage par une phase de modélisation de contexte est primordial. Au fil des années, les méthodes de représentation évoluent, distinguant trois classes de représentation (clé-valeur, méthodes graphiques, méthodes basées ontologie, méthodes logiques), où chacune d'entre-elles possède des avantages et des inconvénients. Quelles sont ces méthodes ? Quelles sont les différences entre elles ? Dans ce qui suit, nous essayons de répondre à ces interrogations et à d'autres.

1.6.1. Méthode clé-valeur

Cette représentation nommée aussi **Key-Value** en anglais, utilisée en 1994 [33]. Elle permet de représenter le contexte ainsi que sa valeur actuelle. Elle a été adoptée pour construire les premiers systèmes sensibles au contexte offrant plusieurs avantages :

- Simplicité d'implémentation : on peut le construire sous forme d'une collection spécifique comme un dictionnaire, dans lequel on peut associer à chaque clé une valeur par exemple { "température" : 30, "humidité" : 23 } ;
- Facilité de gestion d'informations.

En revanche, cette structuration présente des inconvénients :

- Incapable de montrer les liens entre les éléments contextuels ;
- Impuissante pour gérer les informations contextuelles distribuées.

1.6.2. Méthodes graphiques

Cette structuration permet de représenter les éléments contextuels en utilisant un graphe conceptuel. Plusieurs méthodes ont été proposées.

Dans son papier sur un système de contrôle du trafic aérien, **Bauer** [34] a proposé un modèle graphique pour la représentation des informations contextuelles en utilisant le langage de modélisation unifié 'UML'.

Dans un autre papier réalisé par **Henricksen** [35] et son équipe, le formalisme « entité/association » a été exploité afin de représenter les éléments de contexte ainsi que la relation entre ces derniers, où une :

- **Entité** ; présente un élément de contexte (matériel/ immatériel) ;
- **Attribut** ; montre une caractéristique d'un élément contextuel ;
- **Association** ; décrit la relation entre deux éléments contextuels, remarquant aussi qu'une association peut avoir des attributs.

Dans un autre esprit graphique pour la représentation des informations contextuelles, **Henricksen** [36] et ses co-auteurs ont proposé une nouvelle méthode baptisée **ORM** (Object Role Modeling). Cette méthode basée sur l'identification de type de 'fait' ainsi que les rôles des entités. ORM a connu plusieurs améliorations, au début puis elle est devenue **CML** [37] (Context Modeling Language) après **XCML** [38] qui est une version basée XML.

En effet, il existe d'autres formes graphiques pour la représentation des informations contextuelles. Nous citons à titre non exhaustif la méthode **GPM** [39] (General profile Model), le langage **ContextUML** [40], etc.

1.6.3. Méthode basée ontologie

Le terme ontologie est utilisé dans le domaine philosophique signifiant l'étude de l'être ou de l'existence. En informatique, une ontologie représente la capacité de formuler un schéma conceptuel rigoureux dans un domaine donné. Autrement dit, elle est une structuration hiérarchique des données incluant tous les éléments contextuels, leurs relations ainsi que la réglementation dans le domaine. Dû à sa puissance de partage d'informations, plusieurs systèmes sensibles au contexte l'utilisent pour modéliser les informations contextuelles. **COBRA-ONT** est un exemple d'ontologie pour les systèmes sensibles au contexte. Il utilise **OWL** (WEB Ontology Language) pour décrire les lieux, les agents, les événements et les propriétés associées dans l'optique d'une salle de réunion intelligente. Cette méthode de représentation présente plusieurs avantages :

- Offre une façon hiérarchique pour la représentation des préférences des utilisateurs afin de simplifier leur gestion ;
- Explorer le potentiel de la technologie web sémantique pour le raisonnement contextuel.

Jusqu'à présent, les langages basés sur l'ontologie sont préférables pour la représentation du contexte en raison de leurs puissantes capacités OWL et RDF en sémantique, formalité, composition de connaissance distribuée et validation partielle des connaissances. Le tableau 1.2 montre une comparaison entre les approches de représentation du contexte [40].

Méthode de représentation Critère de comparaison	Clé-valeur	Méthodes graphiques		Méthode basée ontologie
		UML pour la représentation de contexte	Entité/association	
Raisonnement	Pauvre	Pauvre	Pauvre	Bien
Qualité	Pauvre	Bien	Bien	Bien
Validation	Pauvre	Pauvre	Pauvre	Bien
Formalisation	Pauvre	Bien	Bien	Bien

Tableau 1.2 - Comparaison entre les approches de représentation du contexte

Après une phase de modélisation, le passage vers une étape d’exploitation et d’exploration de contexte est nécessaire. Plusieurs architectures pour l’exploitation de contexte ont vu le jour. Au niveau du prochain passage, nous montrons les différentes architectures des systèmes sensibles au contexte.

1.7. Architecture en couches d’un système sensible au contexte

Avec le grand succès des systèmes sensibles au contexte, une ribambelle de travaux tente de définir une architecture conceptuelle globale pour réduire la complexité de développement de ces applications [41]. Les systèmes sensibles au contexte se distinguent des systèmes classiques par la complexité de conception, de développement, ... L’architecture en couches est une structure visant à considérer le système comme un ensemble de couches : d’acquisition, de réception, de gestion, d’intégration et d’adaptation. La figure 1.4 présente l’architecture en couches.



Figure 1.4 - Architecture globale en couches pour un système sensible au contexte.

- **Couche acquisition des données**, sert à recevoir des données brutes à partir des capteurs. Il est à noter qu'on peut distinguer quatre types de capteurs : capteur physique, capteur virtuel, capteur logiciel et capteur d'identification. Le tableau 1.3 montre les différents types de capteurs.

Capteur physique	Capteur logique	Capteur virtuel	Capteur d'identification
Est un dispositif matériel perspicace, capable de percevoir les informations de l'environnement où il est installé.	C'est un capteur plus au moins de synthèse où il peut être utilisé avec les capteurs physiques afin de déduire une information contextuelle plus précise.	Capable de fournir les informations contextuelles à partir d'application ou des services logiciels. Par exemple connaître l'endroit d'un utilisateur à partir des rendez-vous saisis sur son agenda.	C'est un ensemble de capteurs capables d'identifier les objets, les êtres humains,... Nous citons à titre d'exemple : les RFID, NFC, etc.

Tableau 1.3 - Types des capteurs

- **Couche d'interprétation du contexte** ; assure l'analyse et la transformation des données brutes en données de haut niveau compréhensibles et manipulables ;
- **Couche de stockage du contexte** ; permet d'avoir un modèle pour les données contextuelles ainsi qu'un support de stockage pour les données délivrées par la couche interprétation pour un usage ultérieur ;
- **Couche diffusion du contexte** ; organise et transfère les données capturées via une interface publique ;
- **Couche application**, permet le changement et l'adaptation du comportement du système en cas de changement de contexte.

L'utilisation et l'exploration du contexte dépend de plusieurs facteurs, nous citons par exemple l'évolution de la téléphonie mobile. Dans ce qui suit nous essayons de comprendre l'influence de l'évolution du téléphone mobile sur l'exploration du contexte.

1.8. Utilisation du contexte avec l'évolution du téléphone mobile

Les années 90 glorifient l'arrivée de la deuxième génération des téléphones mobiles utilisant la norme GSM (Global System for Mobile Communications). En revanche, ces appareils restent impuissants à cause du nombre restreint des services offerts tels que : la transmission de la voix, et le transfert des messages.

Ces années ont connu aussi l'apparition des premiers systèmes sensibles au contexte, qui ne font pas appel au service de téléphonie mobile. Ces systèmes focalisent sur la détection de la localisation de l'utilisateur en utilisant des capteurs infrarouges. Nous citons à titre non exhaustif : le système **Active Badge** [42], le système **ParcTab** [43],... Par contre, nous remarquons que la totalité des applications sensibles au contexte à cette époque s'intéressent à la mobilité des utilisateurs ce qui rend l'intégration des téléphones mobiles un atout pour la création de ces systèmes.

Vers la fin des années 2000, la troisième génération des téléphones a vu le jour sous l'appellation de smartphone. Ces nouveaux gadgets deviennent plus matures en intégrant plusieurs types de capteurs GPS, caméra, capteur de rotation, ... En ce temps-là, plusieurs travaux portant sur la sensibilité au contexte intègrent le mobile pour pouvoir exploiter sa puissance. Nous citons à titre d'exemple l'application de tourisme dans le mobile **COMPASS** [44], ...

Nous remarquons une évolution parallèle entre le téléphone mobile et la sensibilité au contexte. La notion de contexte a été étendue à d'autres horizons hormis la localisation.

Ces dernières années, les smartphones ont conquis notre quotidien, ils sont devenus de plus en plus sophistiqués avec l'incorporation d'autres capteurs tels que les capteurs biologiques, capteurs d'empreinte... ainsi que des technologies de communication telle que la NFC (communication en champs proche) qui peut jouer le rôle d'un capteur d'information contextuelle. Cette technologie est potentiellement capable de récupérer plusieurs types d'informations contextuelles par exemple la localisation de l'utilisateur, ses préférences avec un simple geste, où nous nous trouvons plusieurs travaux incorporent la NFC dans des applications sensibles au contexte ce sujet va être bien décortiqué dans le prochain chapitre.

1.9. Conclusion

Dans ce chapitre, nous avons présenté en premier la notion du contexte et ses caractéristiques. En second, nous avons décrit les systèmes sensibles au contexte, leur architecture globale (la couche capture du contexte, l'interprétation du contexte, le stockage du contexte, la diffusion du contexte et la couche application). En troisième, nous avons abordé l'évolution de la notion avec les progrès des téléphones mobiles.

En effet, l'intégration de la technologie NFC dans le téléphone mobile rend l'utilisateur capable d'avoir plusieurs services en utilisant le même appareil avec un simple geste. Par exemple il peut utiliser son téléphone en tant que :

- Portemonnaie sans contact ;
- Billet sans contact ;
- Clé pour sa voiture ;
- Accès à son appartement.

Mais aussi la NFC peut offrir des services et des informations essentielles sur l'utilisateur dans d'autres applications telles que les applications sensibles au contexte. Ces dernières peuvent utiliser la NFC comme étant un capteur d'information contextuelle (localisation, préférence, ...). Qu'est-ce que NFC ? Qu'est-ce qu'une application NFC mobile ? Dans le troisième chapitre, nous apportons de la lumière sur ces concepts.

CHAPITRE 2

ÉTAT DE L'ART SUR LES APPLICATIONS NFC

Sommaire

- 2.1. Introduction
 - 2.2. Partie Physique
 - 2.2.1. Définition
 - 2.3. Avantages de la technologie NFC
 - 2.3.1. Sécurité
 - 2.3.2. Simplicité et souplesse
 - 2.3.3. Compatibilité avec les infrastructures RFID
 - 2.3.4. Coût faible
 - 2.4. Inconvénients
 - 2.4.1. Dépendance d'opérateur mobile
 - 2.4.2. Peur des utilisateurs
 - 2.4.3. Attaques
 - 2.5. NFC dans le téléphone mobile
 - 2.6. Normes et compatibilité
 - 2.7. Applications NFC
 - 2.7.1. Qu'est-ce qu'une application NFC mobile ?
 - 2.7.2. Taxonomie des applications NFC
 - 2.7.3. Conception des applications NFC
 - 2.7.4. Fondements théoriques
 - 2.7.5. Défis de développement
 - 2.8. Conclusion
-

« *Toute technologie avancée est magique* »

(Arthur Charles Clarke)

2.1. Introduction

Partant d'un monde dominé par la force d'imposture passant à une période parlant que de science et technologie et arrivant à un univers où la barrière entre le monde réel et le monde virtuel est de moins en moins visible.

En 2002, Sony et Philips dévoilent sur une invincible technologie baptisée Near Field Communication 'NFC', d'après Google¹ cette technologie mythe est capable de créer un canal entre le virtuel et le réel.

Au-delà de la nature de cette technologie, certaines organisations prennent l'initiative contribuant à ouvrir la voie à son déploiement en plein essor. En 2002, Europe ECMA adopte NFC comme un standard. Une année après, commence l'épanouissement de NFC, où elle est devenue un standard ISO/IEC. Ensuite, en 2004, les principaux développeurs Philips, Sony et Nokia ont fondé le NFC forum comme une association à but non lucratif pour promouvoir NFC en élaborant des spécifications, assurant l'interopérabilité entre les dispositifs et les services en facilitant son marché [45]. Aujourd'hui, NFC forum est constitué de cent quarante membres finançant et encourageant le développement et le déploiement des services NFC.

Au cours des années passées, un débat sur la vraie valeur de la NFC a été ouvert. En juin 2013, un reportage réalisé à Strasbourg à l'occasion du Forum services mobiles sans contact où ils ont conclu que :

- Tous les fournisseurs d'un service à base de NFC vont devoir rassurer leurs clients ;
- La technologie NFC existe mais elle manque d'un bon marketing.

Une autre enquête réalisée entre 2012 - 2013, sur l'utilité d'utilisation de la NFC dans le domaine de paiement mobile, les pourcentages sont comme suit [46] :

- 10% croient que plusieurs barrières empêchent le succès de la NFC dans le M-Paiement ;
- 15% pensent que la NFC n'est pas utile dans le M-Paiement ;
- 25% des personnes disent que la NFC a des avantages secondaires : marketing, dans la billetterie, ...
- 35% croient que la NFC est critique et essentielle pour le M-Paiement.

¹ (Google, 2011) « bridging the physical and virtual world »

Chapitre 2. Etat de l'art sur les applications NFC

L'interprétation de ces avis revient aux failles rencontrées au niveau des protocoles de sécurité où une expérience réalisée par des chercheurs britanniques sur les cartes de paiement NFC confirmant que c'est possible de voler des millions d'euros [47].

En 2020, le basculement vers le monde sans contact est primordial surtout après cette pandémie COVID 19 qui a touché le monde et qu'elle a plongé dans un état de consternation total, plusieurs pays tels que la Chine, les Etats unis, la Corée du sud, etc. ont adopté la technologie NFC afin de diminuer l'ampleur de cette pandémie.

En effet, les points de vue se sont mis tous d'accord sur le rôle central joué par la technologie sans contact et surtout NFC. C'est quoi la NFC ? Quelles sont les mesures qu'ils doivent prendre pour diminuer les problèmes de la NFC ? Ce chapitre va prendre en charge les réponses de toutes ces questions en deux parties, une première partie se concentre sur l'aspect physique de la NFC, une deuxième partie est réservée à la présentation des applications NFC mobiles.

2.2. Partie Physique

2.1.1. Définition

NFC est l'abréviation de Near Field Communication ou en français Communication à Champ Proche. Elle représente la seconde génération de la technologie sans contact de proximité, où l'échange de données entre deux dispositifs (tag RFID / NFC et un mobile NFC, deux mobiles NFC, mobile NFC et lecteur) s'établit avec un paradigme touching. Ce standard de communication permet de réaliser des transactions sécurisées à une distance ne dépassant pas quatre centimètres [48].

Le cerveau humain a tendance à comparer entre les choses afin de faire la distinction entre elles. Pour cela, dans une discussion contenant la technologie NFC comme acteur principal un passage via la RFID est sine-qua-none. La technologie RFID est la brique de base de la construction de la NFC, depuis il y a une grande confusion entre ces deux dernières. Afin de remettre les choses en place, des critères de comparaison ont été adoptés :

- **Fréquence de travail**

Tous les systèmes basés NFC fonctionnent sous une fréquence de 13.56 MHz, Cela va de soi, la plage de fonctionnement limite la distance de transmission des données qui peut aller jusqu'à vingt centimètres en théorie et qui ne dépasse pas les quatre centimètres en pratique cette restriction est un privilège pour l'aspect sécuritaire. En revanche, les systèmes RFID peuvent fonctionner sur quatre domaines de fréquences différents [49] :

Chapitre 2. Etat de l'art sur les applications NFC

Basses fréquences ou LF ; est une fréquence inférieure à 135 kHz, distance de transmission est de 0 à 1cm ;

Haute fréquences ou HF ; est une fréquence autour de 13,56 MHz, distance de transmission jusqu'à 1 m ;

Ultra hautes fréquences ou UHF ; est une fréquence autour de 434 MHz, de 869-915 MHz et de 2.45 GHz, distance de transmission est de 3 m ;

Micro-ondes ou SHF ; est une fréquence autour de 2.45 GHz, distance de transmission peut aller jusqu'à 1 km.

- **Mode de communication**

Les systèmes à base RFID communiquent selon le mode lecture / écriture où un tag peut être lu / écrit par le transpondeur. En revanche, la technologie NFC suggère trois modes de communication : mode lecture, mode P-2-P et le mode émulation de carte (plus de détail dans la section 2.3). La figure 2.1 présente une comparaison entre RFID et NFC.

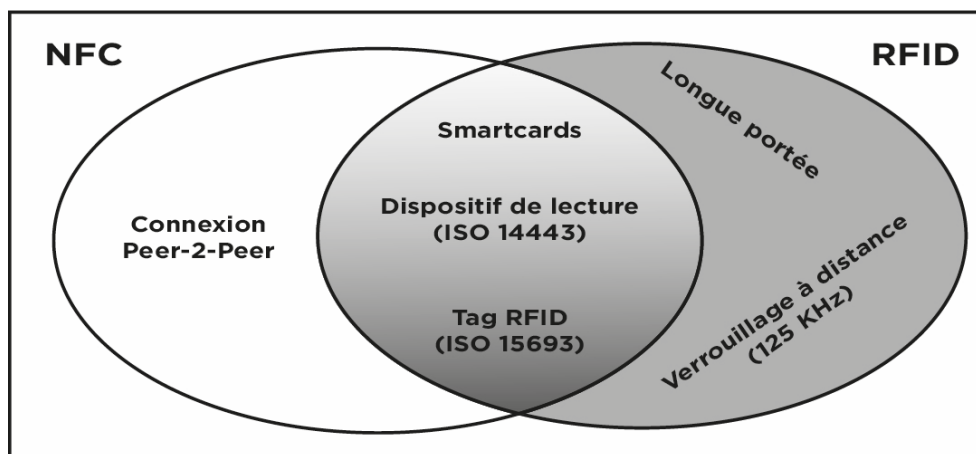


Figure 2.1 - NFC vs RFID [50]

- **Infrastructures**

Un système RFID est constitué d'une station de base qui peut être fixe ou mobile intégrant une antenne ou non, et des tags. La station émet des ondes radio magnétiques dans un espace de quelques centimètres ou de dizaines de mètres selon la puissance de l'alimentation et la fréquence de transmission. Les tags qui sont présents au niveau du champ du lecteur vont intercepter ce signal et seront activés si on parle d'un mode passif bien sûr, et puis enregistrent les données numériques au niveau d'une puce électronique intégrée. La figure 2.2 montre le mode de fonctionnement de RFID.



Figure 2.2 - Mode de fonctionnement RFID [51]

A l'opposé, le mode de fonction ainsi que les composants nécessaires sont fortement liés au mode de communication choisi et le domaine d'application déterminé. La figure 2.3 montre les acteurs nécessaires pour l'établissement d'une communication sous NFC.

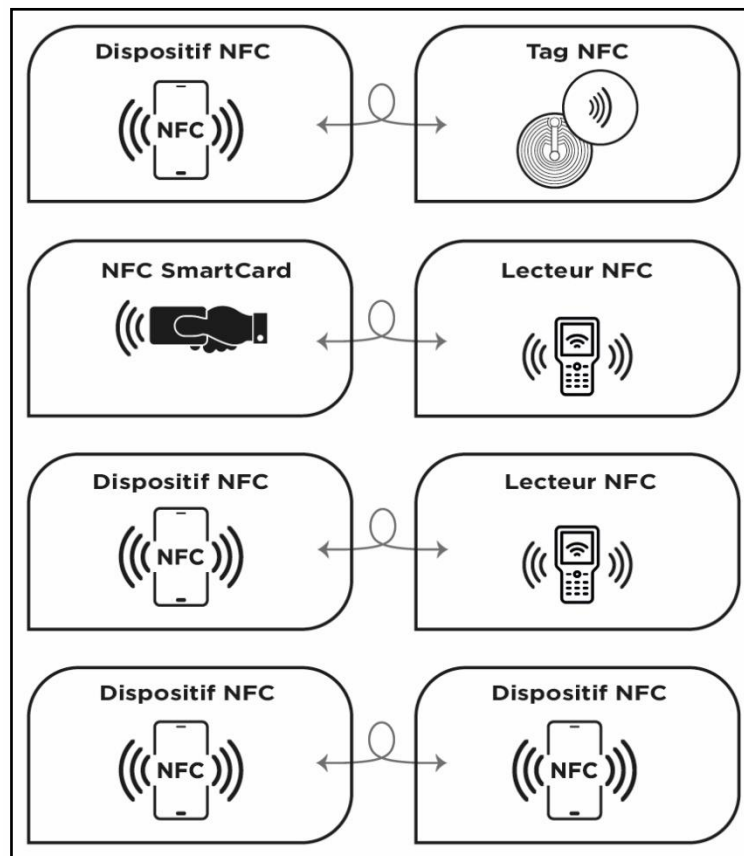


Figure 2.3 - Acteurs nécessaires pour établir une communication sous NFC

Chapitre 2. Etat de l'art sur les applications NFC

Le tableau 2.1 présente des exemples sur des domaines exploitant NFC :

Domaine d'application	Mode de communication	Infrastructure nécessaire
Paiement NFC, billetterie, carte accès	Emulation de carte	Enregistrement données sur un élément sécurisé "SE" ; Plateforme de gestion SE : existe ; Gestion des données par : des systèmes complexes
Lecteur d'un poster	Lecture / écriture	Enregistrement des données : Tag ; Plateforme de gestion SE : n'existe pas ; Gestion des données par : des systèmes simples.
Transfert des données	P-2-P	Enregistrement des données sur : mémoire de téléphone ; Plateforme de gestion SE : n'existe pas.

Tableau 2.1 - Exemples sur des domaines exploitant NFC

2.3. Avantages de la technologie NFC

2.3.1. Sécurité

Le mode de transaction offert par la technologie NFC garantit plus de sécurité par rapport à la carte bancaire actuelle. Cela est dû à sa simplicité, sa rapidité, sa courte distance de transmission et son niveau élevé de cryptage de données.

2.3.2. Simplicité et souplesse

Pour que l'utilisateur transfère ou reçoive des données telles qu'une image vidéo ou autres, il suffit de rapprocher son téléphone de la source et avec une simple confirmation, les données seront envoyées.

2.3.3. Compatibilité avec les infrastructures RFID

Parmi les grands atouts de la technologie NFC est sa compatibilité avec les infrastructures RFID. Dans la mesure où une compagnie veuille opter pour une solution NFC afin d'améliorer son service, cela est possible avec une simple adaptation de l'ancienne solution RFID, ce qui permet de diminuer le coût.

2.3.4. Coût faible

Avec un faible coût, cette technologie peut dominer le marché, nous citons le coût de 1€ pour un lecteur NFC et le coût variant de 0.1 à 1€ pour un tag.

2.4. Inconvénients

2.4.1. Dépendance d'opérateur mobile

Lorsqu'on parle d'une architecture centrée de téléphone mobile, cela signifie que l'utilisateur est dépendant de son opérateur. Ce dernier peut par la suite collecter des données NFC utilisées par l'utilisateur.

2.4.2. Peur des utilisateurs

Beaucoup de personnes ont du mal à confier leur carte de crédit à leur téléphone portable. Bien qu'il existe plusieurs solutions prouvant son succès comme Google Wallet.

2.4.3. Attaques

- **L'attaque phishing**, est un moyen simple et peu coûteux pour induire l'utilisateur en erreur. La plupart du temps, elle concerne les systèmes de lecture des tags NFC [52] [53].
- **L'attaque man-in-the-middle**, plusieurs études confirment que dans un environnement réel, l'application d'une attaque man-in-the-middle- sur une liaison NFC est impossible [54] [55].
- **Menace de relais**, l'attaque, par relais cible essentiellement les cartes de paiement sans contact, consiste à utiliser un proxy NFC. Deux appareils sont donc nécessaires : le premier sera à l'écoute de la carte, le second près du terminal de paiement. Ce dernier va initier le paiement à la borne et transmettre toutes les informations via une connexion wifi [56].

- **Menace d'écoute**, elle est possible par la récupération d'un signal radio émis par les appareils afin de subtiliser des informations personnelles. Cette attaque peut être effectuée depuis plusieurs mètres avec l'utilisation d'une antenne spécifique [57].

2.5. NFC dans le téléphone mobile

NFC a existé depuis 2002, sous forme des cartes sans contact. Mais elle a été rapidement intégrée au téléphone mobile pour de nombreuses raisons [58] [59] :

- Le téléphone mobile est un composant personnel à utiliser indépendamment de toute contrainte spatiale et temporelle.
- Le téléphone mobile possède une source d'énergie (autonomie), une mémoire, une interface conviviale, ...
- Le téléphone mobile offre plusieurs styles réseaux tels que : WIFI, Bluetooth, etc.
- Plusieurs services mobiles sont fournis.

Un mobile NFC doit contenir deux composants [60] :

- **Une extrémité frontale**

Composée d'un contrôleur NFC qui gère et contrôle la liaison entre le SE et le monde extérieur à l'aide d'une interface (Single Wire Protocol 'SWP' ou S2C) et une antenne qui intercepte les signaux de l'extérieur en respectant la norme ISO14443.

- **Un élément sécurisé (Secured Element) 'SE'**

Est l'organe de sécurité, où les applications et les données sont logées² confidentiellement. En réalité, le SE est l'ensemble de : matériels, logiciels et protocoles qui ont pour objectif la protection des informations dites sensibles de l'utilisateur contre les attaques logiques et physiques. Le cycle de vie d'un SE [61] passe par quatre phases essentielles [62] :

1. **Initialiser SE**, le fournisseur de la plateforme agit comme l'émetteur de l'élément sécurisé ;
- 2a. **Activer SE** ;
- 2b. **Affecter une plateforme manager** ;
3. **Télécharger des applications** ;
4. **Désactiver SE**.

² SE : assure le stockage et la sécurité des applications et des données issues des deux modes de communication : R/W et le mode émulation de carte.

La figure 2.4 montre le cycle de vie d'un SE.

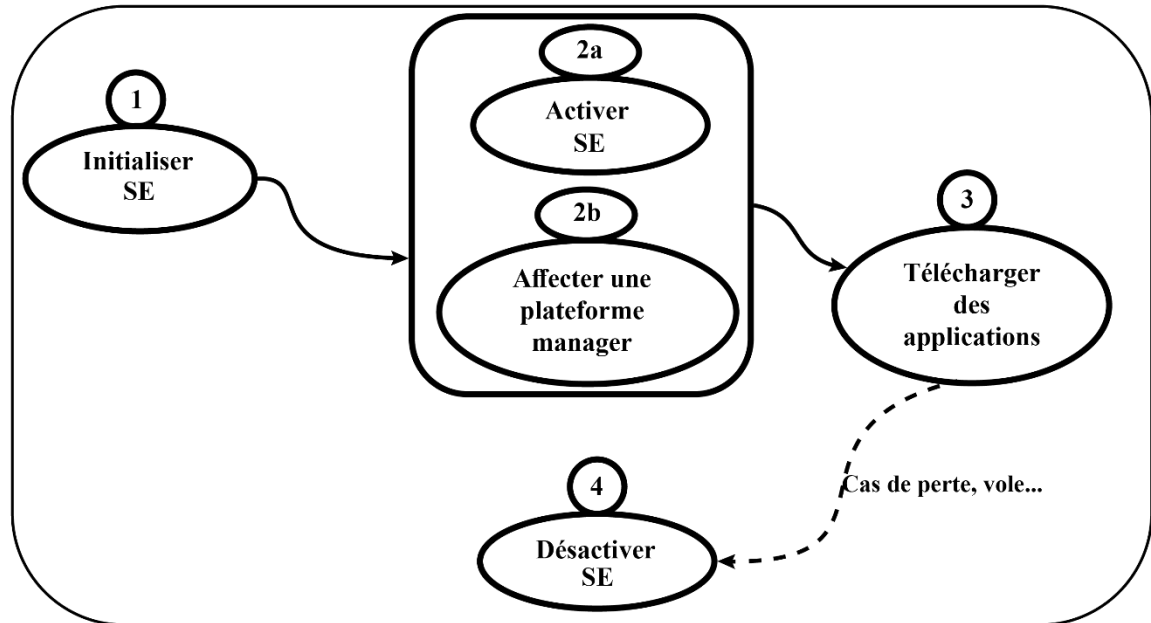


Figure 2.4 - Cycle de vie d'un SE.

La figure 2.5 montre les composants formant un téléphone portable NFC.

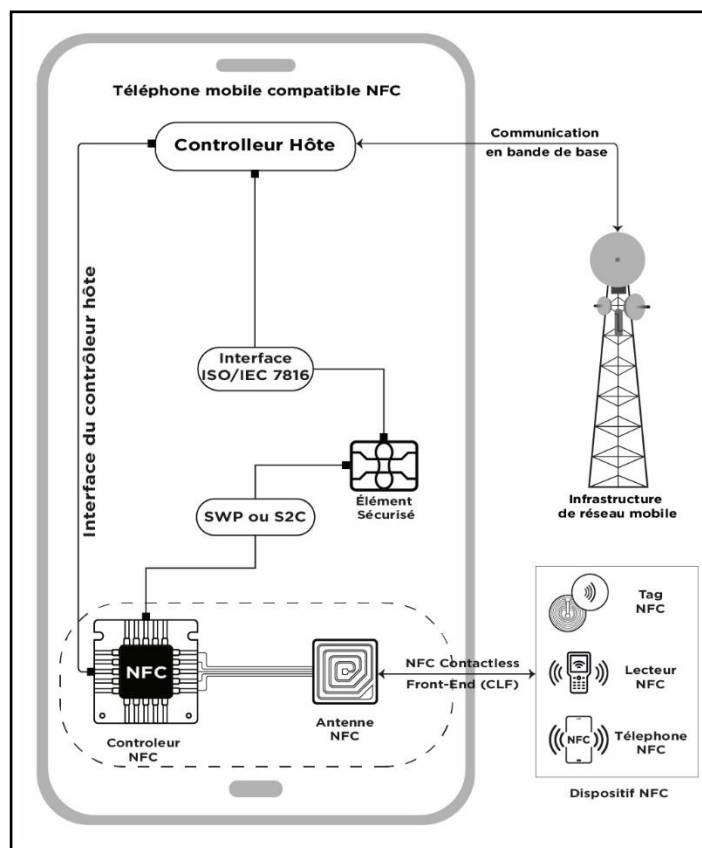


Figure 2.5 - Composants d'un téléphone portable NFC [62]

Chapitre 2. Etat de l'art sur les applications NFC

Le tableau 2.2 contient une explication sur les composants constituant le téléphone NFC.

Nom	Description	Observation
Contrôleur NFC	<p>Puce dédiée avec son antenne offre une communication sans contact selon trois modes.</p> <p>Nécessaire pour la conversion numérique analogique des signaux transférés via la connexion de proximité.</p>	Le standard de fonctionnement est ISO/IEC 18092.
SWP or S2C	Interface physique NFC.	<p>SWP : interface physique avec une seule liaison ;</p> <p>S2C : interface physique avec deux fils.</p>
SE	C'est l'organe le plus important dans un système NFC. C'est l'ensemble hardware, software et protocoles assurant la haute sécurité des données et des applications.	<p>Existe sous quatre formes :</p> <ul style="list-style-type: none"> • Carte SIM ; • Intégré directement sur le téléphone mobile ; • Carte amovible ; • Intégré directement sur le processeur de téléphone portable.
Contrôleur hôte	Capable d'échanger des données avec l'élément sécurisé SE (en mode interne par exemple, pour recharger d'argent dans l'élément sécurisé sur OAT ³) à travers l'interface SWP.	Contrôleur hôte et le SE, communiquent via l'interface JSR 177, selon le standard ISO 7816.
Contrôleur hôte Interface (HCI)	Est une interface logique permettant à une interface NFC de communiquer directement avec les applications, le processeur et des SE multiples.	Fonctionnelle sous le standard ETSI TS 102 622.

Tableau 2.2 - Composants d'un NFC phone.

³ Over the Air

Chapitre 2. Etat de l'art sur les applications NFC

En ce moment, le marché dispose d'une manière très large des mobiles NFC. De ce fait, nous citons d'une manière non exclusive quelques noms des mobiles NFC [63] :

Acer Liquid S2 de Acer ; BlackBerry Z30 (l'appareil fonctionne avec BlackBerry 10 OS version 10.2) ; Google Nexus 9 (annoncé en Octobre 2014 comme un mobile NFC) ; Huawei P8 (smartphone Android avec NFC) ; LG T530, Motorola Razr i/MT788 ; Nokia Lumia 930 (utilise Windows OS) ; Samsung Galaxy S6 (est l'un des dispositifs les plus puissants pour 2015 du géant de la téléphonie coréenne) ; Sony Xperia Z Ultra (fonctionne avec Android OS version 4.4)

Au départ, Apple avait une réticence pour intégrer la NFC dans ces appareils mais à partir de l'iPhone 5S avec son système d'exploitation IOS13, tous les smartphones d'Apple sont dotés de la puce NFC afin d'offrir cette technologie à ses clients.

A la lumière de ce qui précède, un sérieux problème provoqué par l'hétérogénéité des composants qui s'impose nécessitant l'intervention immédiate de toutes les parties concernées. En conséquence, un processus de standardisation est commencé. Dans ce qui suit, nous discuterons d'une manière détaillée les normes et les standards NFC.

2.6. Normes et compatibilité

L'objectif phare d'un processus de standardisation est de rendre l'interopérabilité entre les composants hétérogènes de plus en plus facile. A ce titre, deux conditions doivent être vérifiées [64] :

- Définir des protocoles communs tels que : le standard ISO 14443 A et B ;
- Déterminer des protocoles d'organisation numérique des structures de données par exemple le format NDEF.

Le processus de standardisation NFC a été spécifié au sein de l'ECMA [65], puis la tâche de normalisation a été confiée essentiellement au forum NFC, ETSI, ISO, Global Platform et GSM. La figure 2.6 présente les couches protocolaires de la NFC, qui seront présentées en détails par la suite.

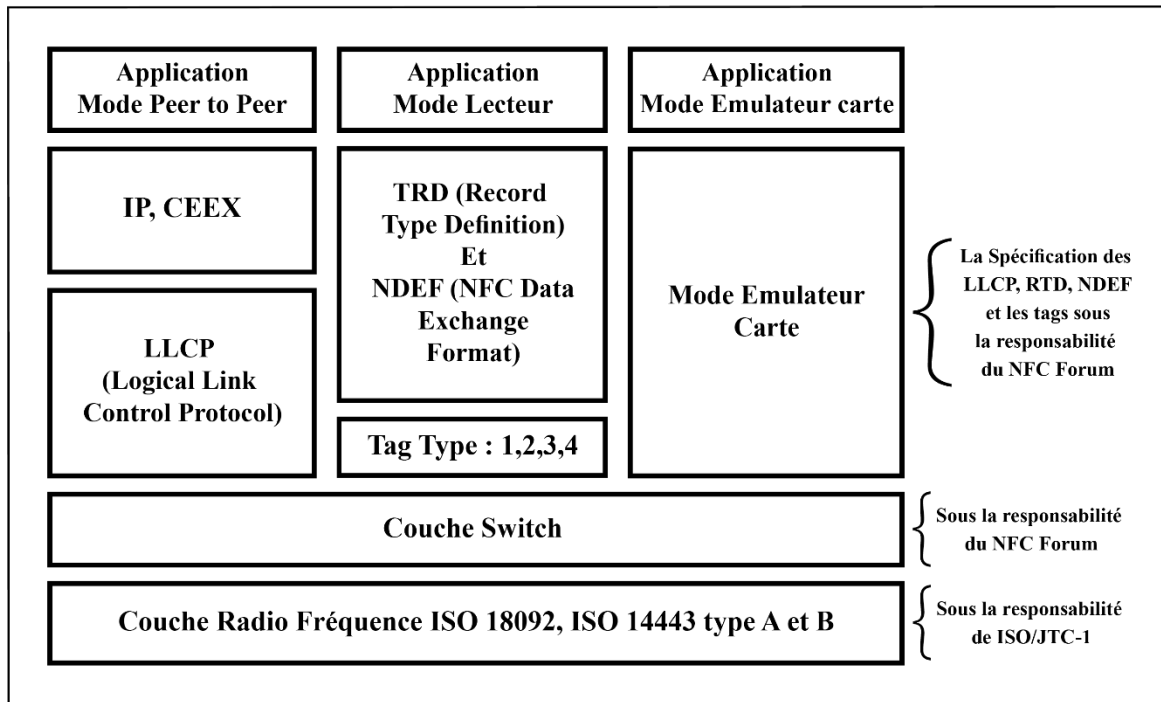


Figure 2.6 - Couche protocolaire NFC [64]

Pour la couche radiofréquence ; il y a deux types de standards : Le premier est l'ISO 18092 qui permet de définir la vitesse de transfert, les méthodes de codage, la modulation et les architectures des trames, etc. Le second est l'ISO 14443 avec ses deux versions A et B, qui sont des versions étendues du standard ISO 14443 pour la communication de cartes sans contact.

La couche mode switch ; permet la gestion de la technologie radiofréquence ainsi que la gestion des deux modes lecteurs et émulation de carte par contre le mode P-2-P n'a pas d'incidence sur cette couche.

La couche placée au-dessus de la couche switch ; dépend du mode de fonctionnement où la définition des standards appropriés à ces modes est sous la responsabilité de NFC Forum. Il existe les trois modes présentés ci-dessous [66] :

Le mode P-2-P ; nécessite la présence d'une couche de liaison de données pour rendre la communication symétrique entre les deux entités communicantes. Les normes ISO 18092 et 14443 sont de nature maître/esclave. Il n'y a que le maître qui a la possibilité de transférer les données. La figure 2.7 présente les couches protocolaires du mode P-2-P ainsi qu'une description détaillée fournie dans le tableau 3.3.

Chapitre 2. Etat de l'art sur les applications NFC

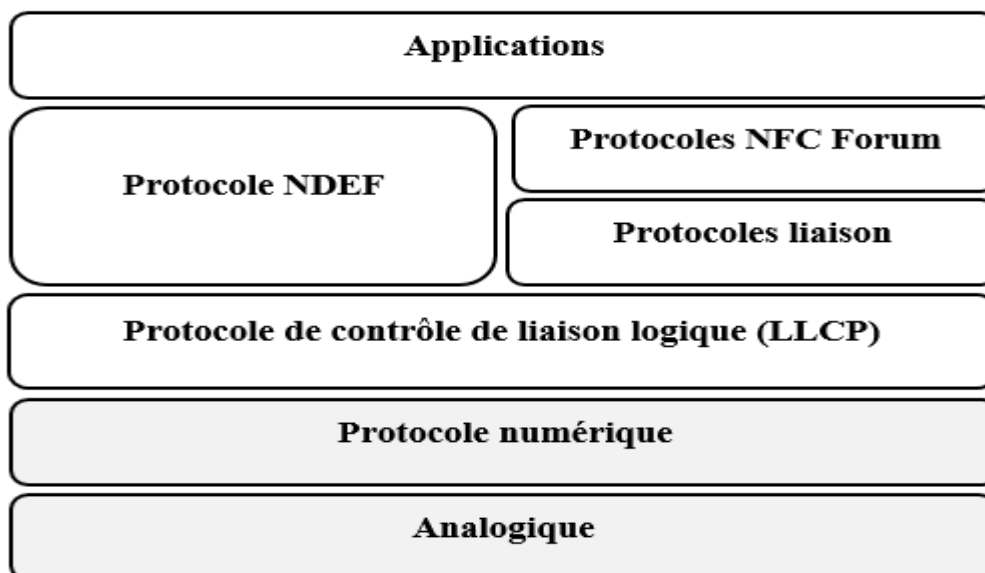


Figure 2.7 - Pile protocolaire du mode P-2-P [66]

Nom de couche	Description
Protocoles analogique/numérique	Représentent les couches physiques standardisées par NFCIP-1 [67] [68].
LLCP	Permet le transfert des unités d'information de couche supérieure entre deux appareils NFC.
Protocoles de liaison	Fournir une liaison pour les protocoles NFC Forum ; Permettre l'utilisation de l'interopérabilité des protocoles enregistrés.
Protocoles NFC Forum	Sont ceux que NFC Forum a défini afin de réaliser la liaison avec la couche LLCP, comme OBEX et IP.
NDEF	Protocole d'échange de données, permet l'échange de messages NDEF. Il est également possible d'exécuter d'autres protocoles sur la couche de liaison de données fournies par LLCP.
Applications	Peut fonctionner sur le protocole simple d'échange NDEF, d'autres protocoles ou les protocoles NFC Forum.

Tableau 2.3 - Description des couches protocolaires du mode P2P

Chapitre 2. Etat de l'art sur les applications NFC

Le mode émulation de carte ; la couche protocolaire dépend de l'application placée dessus. Elle est soit compatible avec le standard ISO 7816-4 commun aux cartes mémoire à contact, soit propriétaire tels que Felica, Mifare, ... la figure 3.8 montre la pile protocolaire du mode émulation de carte.

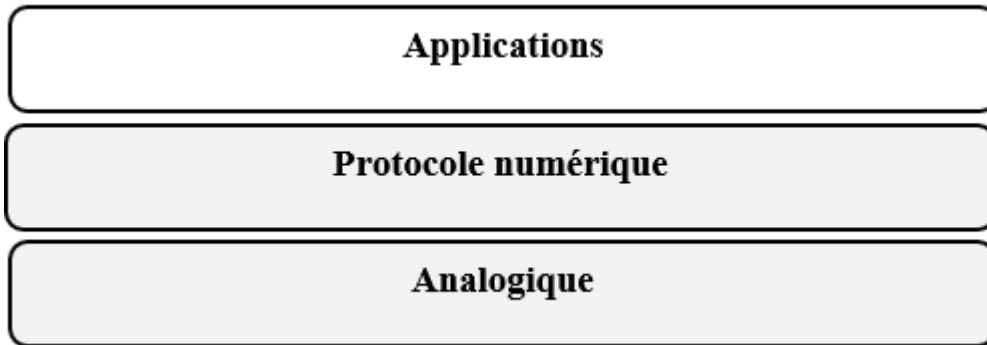


Figure 2.8 - Pile protocolaire du mode émulation de carte [66]

Le mode lecteur ; le terminal mobile devient un lecteur de cartes sans contact ou de tags passifs. Un usager peut lire des informations en approchant son mobile à des étiquettes électroniques déposés dans la rue, sur des affiches, sur des colis, etc. NFC forum offre quatre types de tags. Le tableau 2.4 montre ces types [66].

Chapitre 2. Etat de l'art sur les applications NFC

Tag de NFC forum	Basés sur le standard ISO	Noms de tag	Capacité mémoire	Débit de transfert de données	Prix	Sécurité	Disponible depuis
Type 1	ISO 14443 A	TOPAZ	96 bytes	106 kbit/s	Bas	16 ou 32 bits pour signature	Innovision Research and Technology
Type 2	ISO 14443 A	MIFARE	48 bytes	106 kbit/s	Bas	Non sécurisé	Philips / NXP
Type 3	Felica	Felica	2Kbytes	212kbit/s	Elevé	16 ou 32 bits pour signature	Sony
Type 4	ISO 14443 A/B	Plusieurs noms : MIFARE, DESFire	>3kbyte	106 kbit/s ou plus que 424kbit/s	Elevé	Variable	Plusieurs Fournisseurs

Tableau 2.4 - Types Tag de NFC forum

La figure 2.9 définit la pile protocolaire du mode lecture et le tableau 2.5 décrit la pile protocolaire en mode lecture / écriture.

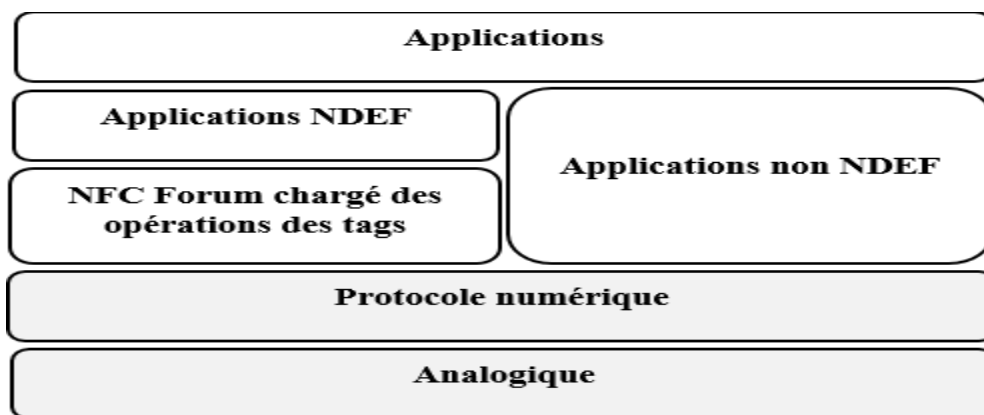


Figure 2.9 - Pile protocolaire du mode lecture [66]

Nom de couche	Description
Protocoles analogiques/numériques	Représentent les couches physiques (inférieures), standardisées par NFCIP-1.
NFC forum chargé des opérations des tags	Indiquent les commandes et les instructions utilisées par les appareils NFC pour fonctionner. NFC Forum propose des tags obligatoires qui sont de type 1, type 2, type 3, et de type 4. Ils permettent de lire / écrire les opérations en utilisant le format de données NDEF et RTD de /vers le tag.
Applications NDEF	Sont basées sur les spécifications NDEF telles que l'affiche intelligente et informations sur les produits. de lecture de NFC, dépliants commerciaux intelligents.
Applications non NDEF	Sont des applications spécifiques des fournisseurs comme un lecteur de billet électronique de lecteur de l'équilibre de sac à main et sans contact qui ne sont pas basées sur les spécifications NDEF.

Tableau 2.5 - Description des couches protocolaires du mode lecture / écriture [66]

NFC a évolué au point où elle offre des avantages significatifs dans plusieurs domaines. En addition, la grande gamme des mobiles intégrant la NFC encourage plusieurs compagnies à son adaptation afin d'améliorer les services. Dans la section suivante, certaines applications sont introduites pour donner un aperçu sur l'objectif et les capacités de NFC.

2.7. Applications NFC

L'apparition des nouvelles technologies a beaucoup influencé le comportement des individus, leur vision et surtout leur manière d'interagir avec l'environnement. Tenant l'exemple d'argent, jadis les gens ont manipulé des billets mais tout d'un coup les distributeurs automatiques ont révolutionné la façon dont les gens agissent et pourtant ils ont été accueillis avec beaucoup de scepticisme initialement. Dans un passé proche, obtenir de l'argent du mur a été considéré comme ridicule, alors qu'aujourd'hui nous ne pouvons pas penser à vivre sans eux.

De même, les cartes de crédit ou de débit sont également devenues un jalon. Le scepticisme a été alimenté par le fait que les gens ne pouvaient pas penser à payer quelque chose avec un simple morceau de plastique. NFC est destiné à fournir la prochaine génération de "gadgets" pour payer et pour d'autres applications. Citons quelques applications références de NFC : paiement, billettique, contrôle d'accès, etc.

2.7.1. Qu'est-ce qu'une application NFC mobile ?

En 1988, Mark Weiser aperçoit une nouvelle ère d'informatique dite informatique ubiquitaire ou aussi pervasive. Le concept va conquérir notre quotidien et il est devenu possible de rendre les objets communicants, interactifs et intelligents. Weiser propose trois formes d'équipements pour les systèmes ubiquitaires :

- Des appareils utilisables indépendamment des contraintes temporelles et spatiales ;
- Des dispositifs de taille de poche ;
- Des dispositifs avec un affichage interactif.

L'apparition des dispositifs mobiles ainsi que les technologies de communication telles que (Bluetooth, ZIGBEE, RFID, NFC, Wi-Fi, WiMAX) a donné une âme à cette idée. Dans cette thèse, nous nous intéressons aux applications mobiles NFC.

Une application NFC mobile est toute application enfouie dans un téléphone portable, qui s'articule nécessairement sur le standard NFC pour le transfert ainsi que la sauvegarde des données d'une manière sécurisée. En général, une application NFC est une application interactive et distribuée, où cette interaction se réalise avec d'autres applications (application web, application banque, etc.) et plusieurs acteurs exigeant un taux de souplesse et de sécurisation très élevé.

2.7.2. Taxonomie des applications NFC

Dès le début de l'ère NFC, plusieurs scientifiques ont montré beaucoup d'intérêt ce qui s'est bien manifesté par le nombre des articles publiés. En revanche, il est presque impossible de cerner tous les types d'applications NFC et la réalisation d'une classification voire une catégorisation est primordiale. Ainsi, nous avons lu une pléthore d'articles publiés dans les grandes bases telles que : IEEE, ACM, ...dans le domaine, nous avons relevé deux classifications saillantes des applications NFC. Nous avons ensuite proposé notre propre classification.

Chapitre 2. Etat de l'art sur les applications NFC

Première classification, découle de l'analyse de soixante-quinze articles publiés en 2012 sur les domaines de services de la NFC. Les applications NFC ont été regroupées en huit domaines de services [69] :

1. Services de santé (D1) ;
2. Services environnementaux intelligents (D2) ;
3. Services de paiement, billetterie et fidélité (D3) ;
4. Services de divertissement (Entertainment) (D4) ;
5. Services de réseaux sociaux (D5) ;
6. Services éducatifs (D6) ;
7. Services de géolocalisation (D7) ;
8. Services de gestion de ventes (D8).

Malgré ce travail colossal et respectable, quelques préoccupations demeurent et ont besoin d'être élucidées :

- Pourquoi choisir uniquement le domaine de services comme un critère de classification ?
- Sur quelles bases ont été regroupées des services ensemble dans un même domaine comme le paiement, la billetterie, la fidélité, ... ?

Deuxième classification, quant à elle, repose sur les modes de communication implémentant des applications NFC mobiles. Les résultats de cette classification sont synthétisés dans le tableau 2.6 [70].

Application	Modes NFC
Environnement automatique	CE, R/W, P2P
Billetterie	CE, R/W
Coupon mobile	CE, R/W
Paiement	CE
Clé électronique	CE
Système de surveillance de santé	CE
Navigation intérieure	R/W
Poster intelligent	R/W
Transfert de fichier	P-2-P

Tableau 2.6 - Classification des applications NFC selon les modes de communication

Chapitre 2. Etat de l'art sur les applications NFC

Cette classification, est plus réaliste, mais, elle a été réalisée sur un nombre restreint de références de ce fait il est difficile de la généraliser.

Nous proposons une classification en s'inspirant des deux dernières. Elle repose sur quatre-vingt articles publiés au sein de différentes bases : ACM, IEEE, Science Direct d'Elsevier, Scopus, etc.

Au départ, nous classifions les articles par domaines puis par modes de communication à l'intérieur du domaine. Les résultats sont consignés dans le tableau 2.7.

Domaine	Mode			
	RW %	P-2-P %	CE %	Comb.de deux %
D1	43.75	18.75	31.25	6.35
D2	50	73.5	12.5	-
D3	33.33	11.115	13.11	44.5
D4	45.4	18.2	18.2	18.1
D5	83.33	16.66	16.66	-
D6	33.3	44.5	-	22.2
D7	16.6	83.4	-	-
D8	50	33.3	-	16.67

Tableau 2.7 - Classification des applications NFC selon domaines et modes

La figure 2.10 montre l'histogramme des résultats.

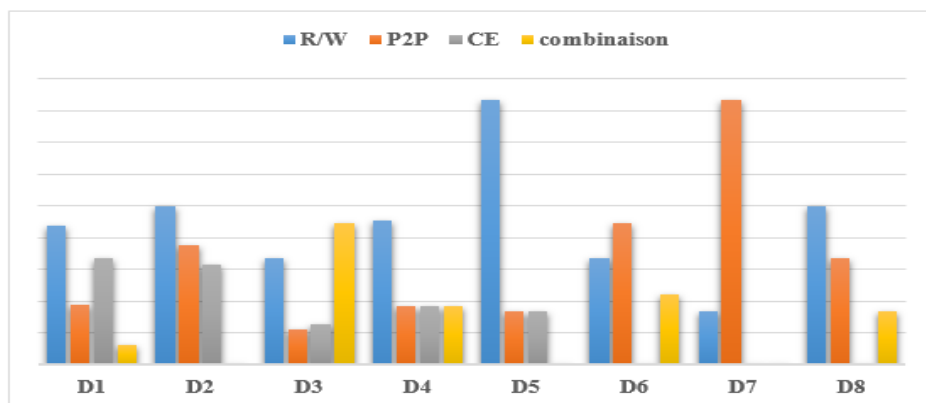


Figure 2.10 - Classification des applications NFC selon domaines et modes de communication

Chapitre 2. Etat de l'art sur les applications NFC

Après cette classification effectuée sur un ensemble d'articles publiées dans le domaine NFC nous pouvons clairement constater :

- Grande gamme d'articles publiés utilisent le mode R/W pour sa facilité d'implémentation ainsi la façon souple offerte pour interagir avec l'environnement ;
- Majorité des articles traitant le transfert et le partage des données préfèrent le mode P-2-P pour sa rapidité de transfert (le temps d'établissement de la connexion est de l'ordre de 0.1s) ;
- Peu d'articles utilisent le mode émulation de carte car la plupart des plateformes supportant NFC ne fournissent pas d'API complètes telle que l'API 2.3.3 d'Android.

A la fin de cette taxonomie, nous pouvons déduire les caractéristiques des applications NFC en respectant le mode de communication. Les applications s'appuyant sur le mode R/W n'utilisent pas un protocole de transfert complexe, elles sont fonctionnelles en mode passif c'est-à-dire que le système peut fonctionner en absence complète d'une source d'énergie puis elles peuvent être des cartes sans contact NFC ou un mobile NFC. Les applications utilisant le mode émulation de carte nécessitent l'exécution d'un protocole complexe, elles sont fonctionnelles que sur un mobile NFC car elles utilisent le SE comme un moyen de stockage des données. Enfin, nous ne devons pas manquer qu'une application avec le mode P-2-P utilise la mémoire de téléphone pour le stockage des données donc il faut faire vraiment gaffe pour la sécurité de ces données. Donc on peut conclure qu'il faut faire très attention lors du choix du type d'application et son mode de communication.

2.7.3. Conception des applications NFC

Le développement d'une application NFC n'est pas vraiment trivial. Un développeur peut voir ces applications comme étant une combinaison d'une interface graphique et une partie métier. Il faut tout d'abord noter que le développement d'une application pareille est confronté à deux styles de contraintes :

- Contraintes liées aux caractéristiques des téléphones mobiles telles que : la limitation d'espace mémoire, la déconnexion soudaine du réseau, ...
- Contraintes liées aux utilisateurs : parmi les sérieux problèmes rencontrés le manque de confiance des utilisateurs car beaucoup de personnes ont du mal à confier leur carte de crédit ou leur ticket à un téléphone portable.

Dans ce sillage, une pléthore de travaux introduisant l'utilisateur dans le processus de développement et l'évaluation de ces applications. Dans ce qui suit, nous présentons les méthodes de conception qui intègrent l'utilisateur comme acteur principal dans le processus de conception de ces applications et plus précisément dans le domaine NFC.

2.7.4. Fondements théoriques

L'implication des clients dans le processus de développement d'une application ou d'un produit a toujours un impact négatif sur les délais de commercialisation et les coûts de développement. En effet, cette inclusion a encouragé l'apparition de plusieurs méthodes potentielles dans le processus de développement aidant à réduire le coût, le temps de développement et de commercialisation par exemples : prototypage, conception participative, user lead approche [71] :

- **Prototypage** ; est l'une des premières approches intégrant le client dans le processus de développement. Son objectif est de bien cerner les contraintes et le contexte d'application afin de dresser la base d'évaluation des clients. Il est utilisé lorsque la création d'un système est complexe donc intégrer l'utilisateur dans ce processus nécessite la présence d'une première copie facile à corriger qui peut être elle-même le système demandé et cela permet de réduire le coût et le temps de commercialisation. Cette approche est largement utilisée pour la conception des applications NFC.
- **Approche compétitive** ; est l'une des approches les plus connues dans l'intégration des utilisateurs. La conception participative est l'ensemble des pratiques, des théories et des études relatives aux utilisateurs finaux comme étant des participants dans le développement de produits logiciels ou matériels. Elle sert à collaborer entre les développeurs et les utilisateurs finaux. Notamment le renforcement de la communication et de la compréhension, améliore l'incorporation de nouvelles idées, améliore les conditions du travail. Cette approche est largement utilisée pour la construction des systèmes interactifs dont l'objectif principal est la réalisation d'un système ayant une meilleure utilisabilité et une meilleure acceptabilité.
- **User lead** ; l'idée est de laisser l'utilisateur conduire le processus de conception [72]. En réalité, cette approche propose une boîte à outils pour les utilisateurs afin de pouvoir créer leurs conceptions préliminaires, simuler ou construire des prototypes, évaluer des fonctions, puis améliorer la conception du produit. En effet, cette approche sert

nécessairement à réduire le temps de commercialisation donc la réalisation d'un produit consistant en tenant compte des avis des utilisateurs.

L'objectif des approches basées utilisateur est d'arriver à concevoir et à évaluer un logiciel afin d'obtenir un produit ayant une meilleure utilisabilité et une meilleure acceptabilité. En revanche, il n'y aucune preuve que le produit commercialisé ne contient pas des erreurs influençant son bon déploiement. Alors introduire les méthodes formelles est primordial dans le processus de construction d'un système plus précisément les applications sous NFC.

2.7.5. Défis de développement

Une étude expérimentale réalisée à travers le monde par TES (Transaction Electronique Sécurisée), a porté sur des services mobiles sans contact dans plusieurs domaines sur toutes les catégories de clients, des deux sexes. Les résultats obtenus ont plaidé en faveur de la NFC : 90% des clients se sont montrés très satisfaits, 85% ont déclaré vouloir recommander le service à leur entourage et 60% ont utilisé régulièrement le service plusieurs fois par semaine durant la période expérimentée. Les clients finaux ont déclaré apprécier la rapidité, la facilité, la simplicité, la modernité et la sécurité de la technologie NFC.

Forts des résultats de cette étude, la plupart des grandes entreprises ont fini par adopter cette technologie pour améliorer la qualité des services fournis mais les développeurs de ces applications sont confrontés à plusieurs défis.

Le développement d'une application NFC nécessite trois conditions sine-qua-non :

- La compréhension de la technologie NFC (Mode de communication, les types d'attaque, etc.) ;
- La prise en charge des contraintes liées aux caractéristiques des téléphones mobiles telles que : la limitation d'espace mémoire, la déconnexion soudaine du réseau, etc.
- La connaissance sur les plateformes mobiles offrant des outils pour la construction d'une application NFC : Actuellement, la plupart des plateformes mobiles offrent les outils nécessaires pour le développement des applications NFC.

Certains outils comprennent un kit de développement logiciel : SDK Android [73] pour le téléphone mobile Android, le QT SDK pour le Symbian 3, le NFC SDK la série 40 pour les appareils Nokia 6212, etc.

2.8. Conclusion

NFC est l'une des technologies sans contact des plus réputées par sa rapidité, sa simplicité et sa sécurité. En revanche, comme toute technologie permettant une manipulation des données personnelles des scénarios improbables peuvent surgir ce qui donne lieu à : " des violations des vies privées" ou à des "types de fuites de données". Pour cela, des mesures de contrôle et des mécanismes de sécurité doivent être mis en place afin de minimiser les dégâts.

Les plateformes mobiles, à leur tour, devraient avoir un moyen pour gérer (NFC ou un autre type d'applications) demandant accès aux informations personnelles. La réticence des utilisateurs qui pèsent sur les applications NFC sont principalement liées à la sécurité.

En effet, introduire les utilisateurs dans le processus de développement et d'évaluation des applications pervasives et plus spécifiquement les applications NFC peut garantir que l'application répond aux exigences signalées au préalable mais cela ne garantit pas l'absence totale des erreurs. Par conséquent, faire appel aux méthodes rigoureuses (méthodes formelles) afin de développer ces applications est primordial.

Dans le chapitre suivant nous parlerons des méthodes formelles et leur utilisation dans le domaine des applications pervasives.

CHAPITRE 3

Formalisme Bigraphical Reactive System 'BRS'

Sommaire

3.1. Introduction

3.2. Modélisation formelle d'un système

3.2.1. Modélisation de système en utilisant les BRS

3.2.1.1. Définition de Bigraphe

3.2.2. Types de bigraphe

3.2.2.1. Bigraphe de base

3.2.2.2. Bigraphe d'ion discret

3.2.2.3. Bigraphe premier

3.2.2.4. Bigraphe discret

3.2.3. Algèbre et la forme normale discrète d'un bigraphe

3.2.4. Exemples d'un Bigraphe

3.2.5. Définition Bigraphical reactive system BRS

3.2.5.1. Composition

3.2.5.2. Transformation

3.3. Pourquoi les BRS pour la modélisation de systèmes sensibles au contexte ?

3.4. Travaux existants utilisant les BRS dans la modélisation des systèmes sensibles au contexte

3.5. Vérification formelle

3.5.1. Vérification de modèle (model-Checking)

3.5.2. Différents outils de vérification pour les BRS

3.6. Conclusion

« La résistance d’une chaîne est celle du maillon le plus faible »

3.1. Introduction

De nos jours, les systèmes deviennent plus complexes en offrant plusieurs fonctionnalités. Une modélisation formelle peut être une façon efficace pour la conception de ce genre de système mais il reste toujours un problème : comment s’assurer que le modèle fourni est pertinent ou non ?

Une méthode de vérification formelle offre une panoplie de métriques mathématiques garantissant l’absence d’erreurs dans un système.

Une méthode de vérification formelle présente l’ensemble des techniques mathématiques permettant une spécification, un développement ainsi qu’une vérification automatique de quelques propriétés. En effet, une large gamme de langages et des outils sont disponibles pour procéder à la vérification des propriétés à n’importe quel niveau de développement. Dès le départ les chercheurs ont eu la conscience de tester l’absence d’erreurs au sein de systèmes. Nous allons distinguer deux périodes importantes qui ont jalonné l’histoire de la vérification. La première période a été connue par l’utilisation des méthodes manuelles pour le test ce qui a rendu et rend la tâche ennuyeuse et très compliquée. La deuxième période a connu l’introduction d’outils automatiques pour réaliser la tâche de vérification.

Les applications sensibles au contexte sous NFC comme toute autre application doivent prouver l’absence d’un maillon faible aux yeux des utilisateurs.

Dans ce chapitre, nous décrirons d’abord la modélisation formelle d’un système informatique, puis nous présenterons le modèle formel BRS (Bigraphical Reactive System), ensuite nous survolerons les travaux qui ont utilisé les BRS pour la modélisation et la formalisation des systèmes sensibles au contexte enfin nous nous attarderons sur la description de la vérification formelle de modèle et les outils spécifiques de vérification pour le formalisme BRS et particulièrement l’outil BigMC.

3.2. Modélisation formelle d’un système

Dans la littérature, un modèle est une représentation abstraite de plus en plus fidèle permettant d’émettre le comportement d’un système donné. Un modèle pertinent est un modèle qui permet de bien cerner le domaine de problème en faisant abstraction aux détails qui ne participent pas à l’obtention des résultats tant attendus.

Chapitre 3. Formalisme Bigraphical Reactive System ‘BRS’

Une modélisation formelle est une phase très cruciale où à la fin nous produirons un modèle pertinent capable ensuite d’être validé en utilisant un outil formel.

Plusieurs modèles formels sont disponibles offrant aux développeurs la possibilité de décrire le comportement de leur système ; nous citons à titre d’exemple : les automates, les réseaux de Petri, les BRS, etc.

3.2.1. Modélisation de système en utilisant les BRS

3.2.1.1. Définition de Bigraphe

La notion de bigraphe a été inventée par Milner en 2001 [74] [75]. Milner a fourni un modèle graphique pour une représentation simultanée de connectivité et de localisation des entités (logique, physique). La combinaison entre un bigraphe et les règles d’interaction formant le Bigraphical Reactive System (BRS).

Formellement : un bigraphe concret est de la forme $G = (V, E, \text{Ctrl}, G^P, G^L) I \rightarrow J$ avec I : interface d’entrée $\langle \text{nombre sites}, \text{ensemble nomes d'entrée} \rangle$, elle permet à un autre bigraphe de s’intégrer au niveau des sites existants. L’interface de sortie $\langle \text{nombre racine}, \text{ensemble nomes de sortie} \rangle$, elle donne au bigraphe l’opportunité de se combiner avec d’autres bigraphes :

V : ensemble des nœuds ;

E : ensemble d’arcs ;

Ctrl : l’ensemble de signatures nommées aussi carte de contrôle, informellement, c’est l’ensemble des informations portant sur les portes d’un nœud pour augmenter la restriction (type, signe, ...) sur lui.

Formellement une signature est l’ensemble des éléments \mathcal{K} appelé contrôle. Pour chaque contrôle $K \in \mathcal{K}$, il existe une arité $ar(K)$.

G^P : Graphe de place ;

G^L : Graphe de lien.

Dans ce qui suit, nous donnons la définition des deux composants (G^P, G^L). Le tableau 3.1 expose la liste des symboles mathématiques [76] utilisés dans ce chapitre.

Symbole	Définition
\circ	Composition
Id	Identité
\otimes	Produit tensoriel
Id_S	Identité d'une fonction d'un ensemble S
$\emptyset S$	Fonction vide d'un ensemble S
$W \uplus T$	L'union de deux ensembles W et T, supposant ou connaissant au préalable qu'ils sont disjoints.
$f \uplus g$	L'union de deux fonctions f et g , supposant ou connaissant au préalable qu'elles sont disjointes.

Tableau 3.1 - Liste des symboles mathématiques [76]

- **Graphe de places**

Le graphe de places montre la distribution des entités géographiquement dans laquelle les places sont remplacées par des nœuds et des sites. La structure arborescente est utilisée pour exhiber au mieux la relation entre les nœuds et les sites de la même racine. Un graphe de places est un triplet, $G = (V; \text{Ctrl} ; \text{prnt}) : m \rightarrow n$, avec m est le poids d'entrée et n est le poids de sortie.

Formellement, le graphe de places est un ensemble fini de nœuds V , avec l'ensemble de signature $\text{Ctrl} : V \rightarrow \mathcal{K}$, et une carte source prnt (Parent map), $\text{prnt} : m \uplus V \rightarrow V \uplus n$.

- **Graphe de liens**

Est un hypergraphe représentant l'interconnexion entre entités. Un graphe de liens est un quadruplet $G = (V, E, \text{ctrl}, \text{link}) : X \rightarrow Y$, il a un ensemble fini de noms d'entrées X et des noms de sorties Y , des nœuds V , des arcs E , il contient aussi deux fonctions :

- Signature $\text{Ctrl} : V \rightarrow \kappa$

- Fonction carte de liens, où la somme disjointe $P \stackrel{\text{def}}{=} \sum_{v \in V} ar(\text{ctrl}(v))$, est l'ensemble de ports de G . Les noms d'entrées X et les ports P sont des points de G , les arcs E et les noms de sorties Y sont les liens.

Remarques

- Un graphe de liens est maigre s'il n'a pas d'arêtes libres ;
- Un nom externe est un lien ouvert ;
- Un bord est un lien fermé ;
- Un point (c'est-à-dire un nom interne ou un port) est ouvert si son lien est ouvert, sinon fermé ;
- Deux points distincts sont pairs s'ils sont dans le même lien.

Après avoir défini les concepts essentiels, nous pouvons maintenant donner la définition complète d'un bigraphe concret.

Un bigraphe concret est la combinaison entre les deux graphes (places-liens) qui ont les mêmes nœuds et la même signature, nous notons $G = (G^p, G^l)$, avec $G^p : m \rightarrow n$ est le graphe de places et $G^l : X \rightarrow Y$ est le graphe de liens.

3.2.2. Types de bigraphes

3.2.2.1. Bigraphe de base

Un bigraphe de base [77] est un bigraphe qui possède une face d'entrée originale c'est-à-dire sans sites et sans noms d'entrées. La forme d'un bigraphe de base est $G : \epsilon \rightarrow I$; avec $\epsilon \in \langle 0, \emptyset \rangle$.

3.2.2.2. Bigraphe d'ion discret

Est un bigraphe élémentaire $K_{\vec{x}} : 1 \rightarrow \langle 1, \{\vec{x}\} \rangle$ pour la séquence $\vec{x} = x_0 + \dots + x_k$ où κ présente l'arité du bigraphe K . La figure 3.1 montre l'ion discret de $K_{\vec{x}}$;

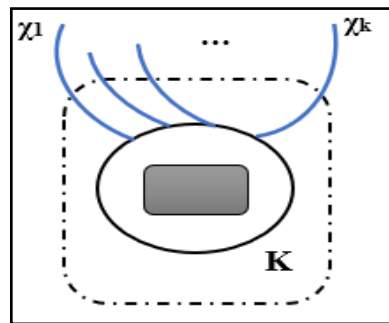


Figure 3.1 - Bigraphe d'ion discret $K_{\vec{x}}$ [77]

3.2.2.3. Bigraphe premier

On dit qu'une interface I est première si et seulement si elle a une largeur unitaire $\langle 1, X \rangle$ peut-être écrite sous la forme $\langle X \rangle$. Un bigraphe $G : I \rightarrow J$ si J est une interface première et I n'a plus de nom alors G est un bigraphe premier. La figure 3.2 montre un exemple d'un bigraphe premier ;

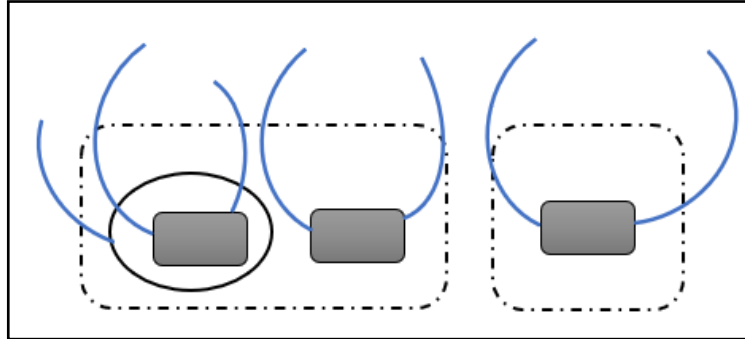


Figure 3.2 - Bigraphe premier [77]

3.2.2.4. Bigraphe discret

Un graph G est un bigraphe discret si et seulement si tous ces liens sont ouverts et s'il contient exactement un seul point c'est-à-dire un seul nom d'entrée ou un seul port P .

3.2.3. Algèbre et la forme normale discrète d'un bigraphe

Dans cette section, nous donnons la forme normale d'un bigraphe cette forme est nommée la forme normale discrète DNF. En réalité, Il existe deux formes dégénérées d'interface : interface de place $\langle m, \emptyset \rangle$ et une interface de lien $\langle 0, X \rangle$. La forme entièrement dégénérée est l'origine $\epsilon \stackrel{\text{def}}{=} \langle 0, \emptyset \rangle$.

La place est un bigraphe sans nœud $m \rightarrow n$, alors n'importe quel emplacement peut être exprimé en termes de trois types :

$1 : \epsilon \rightarrow 1$ Racine stérile ;

Fusionner : $2 \rightarrow 1$ Map deux sites à une racine ;

$\gamma_{m,n} : m+n \rightarrow n+m$ Echanger m avec n places.

Chapitre 3. Formalisme Bigraphical Reactive System 'BRS'

Le lien est un bigraphe $X \rightarrow Y$ où il est nécessairement sans nœuds. N'importe quelle liaison est définie selon ces deux types :

$/x : x \rightarrow \epsilon$ Clôture « Closure » ;

$y/X : X \rightarrow y$ Substitution $x \rightarrow y$ (pour tout $x \in X$).

La figure 3.3 montre la forme graphique d'une clôture et d'une substitution. La figure 3.4 résume les opérations de base sur un bigraphe avec exemple.

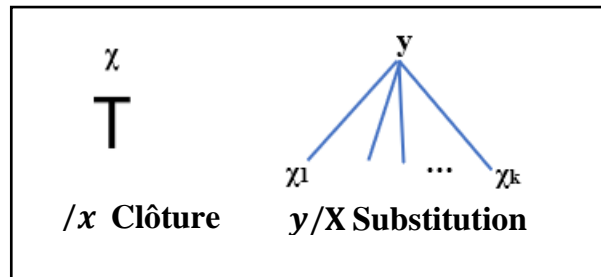


Figure 3.3 - Forme graphique d'une clôture et d'une substitution [77]

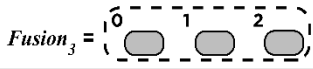
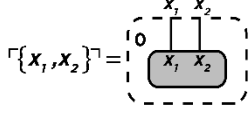
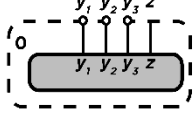
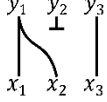
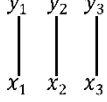
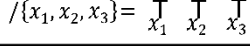
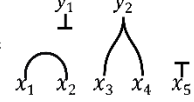
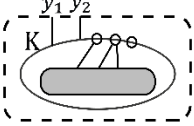
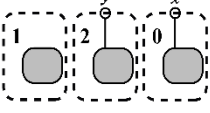
	Notation	Exemple
Fusion	$Fusion_n : n \rightarrow 1$	$Fusion_3 =$ 
Concrétion	$\lceil X \rceil : \langle X \rangle \rightarrow \langle X \rangle$	$\lceil \{x_1, x_2\} \rceil =$ 
Abstraction	$(Y)P : I \rightarrow \langle 1, [Y], Z \sqcup Y \rangle$	$((y_1, y_2))(\{y_3\}) \lceil \{y_1, y_2, y_3, z\} \rceil =$ 
Substitution	$\tilde{y}/\tilde{x} : X \rightarrow Y$ σ	$[y_1, y_2, y_3]/[\{x_1, x_2\}, \{\}, \{x_3\}] =$ 
Renommage	$\tilde{y}/\tilde{x} : X \rightarrow Y$ α, β	$[y_1, y_2, y_3]/[x_1, x_2, x_3] =$ 
Closure	$/X : X \rightarrow \{\}$	$/\{x_1, x_2, x_3\} =$ 
Wiring	$(id \otimes /Z) \sigma : X \rightarrow Y$ ω	$(id_{\{y_1, y_2\}} \otimes /\{z_1, z_2\})$ $[y_1, z_1, y_2, z_2]/$ $[\{\}, \{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}] =$ 
Ion	$K_{y(\tilde{x})} : (\{\tilde{x}\}) \rightarrow \langle \{\tilde{y}\} \rangle$	$K_{[y_1, y_2](\{x_1, x_2, x_3, \{\}\})} =$ 
Permutation	$\{i \mapsto j, \dots\} : \langle m, \tilde{X}, X \rangle \rightarrow \langle m, \pi(\tilde{X}), X \rangle$ $\pi_{\tilde{X}}$	$\{0 \mapsto 2, 1 \mapsto 0, 2 \mapsto 1\}_{[\{x\}, \emptyset, \{y\}]} =$ 

Figure 3.4 - Opérations de base sur un bigraphe [77]

3.2.4. Exemples d'un Bigraphe

Exemple 1

Dans un département deux bibliothèques sont disponibles. Chaque bibliothèque contient plusieurs salles de lecture, chacune de ces salles est doté d'un ordinateur contient la liste des livres disponibles dans la bibliothèque où les étudiants sont capables de les consultés via leurs téléphones portables. Les responsables souhaitent rendre les étudiants capables de partager leurs listes des ouvrages préférés avec les autres étudiants.

Chapitre 3. Formalisme Bigraphical Reactive System 'BRS'

Cet exemple peut être modélisé en tant que bigraphe comme suit :

Le bigraphe G contient deux racines décrivant la localisation des entités, quatre nœuds E, S, O, B. Où :

- E : Est l'ensemble des étudiants portant des téléphones mobiles interagissant et interconnectant ensemble ;
- S : Présente les salles contenant des ordinateurs ;
- O : Ordinateurs dans les toutes les salles au niveau de la bibliothèque B.

Chaque acteur peut être modélisé en tant que nœud. Pour que la liaison soit possible avec les autres nœuds ou région, le nœud doit avoir des portes. En effet, le formalisme bigraphe propose deux types de liens : un lien portant un nom et nous appelons ces liens *lien ouvert* reliant les différentes parties et un *lien fermé* reliant les trois étudiants aux ordinateurs, ou aux bibliothèques ce genre des liens ne les permet pas de les connecter avec d'autres composants dans l'environnement. Chaque nœud a une signature pour le contrôler et une arité montrant le nombre des portes. Dans cet exemple, les arités correspondant aux nœuds sont :

- E : 2 ;
- S : 0 ;
- O : 2 ;
- B : 1.

Chaque bigraphe a un poids correspond au nombre de racine ou de région, dans notre cas le poids de ce bigraphe G : 2. La figure 3.5 montre le bigraphe correspond à notre exemple.

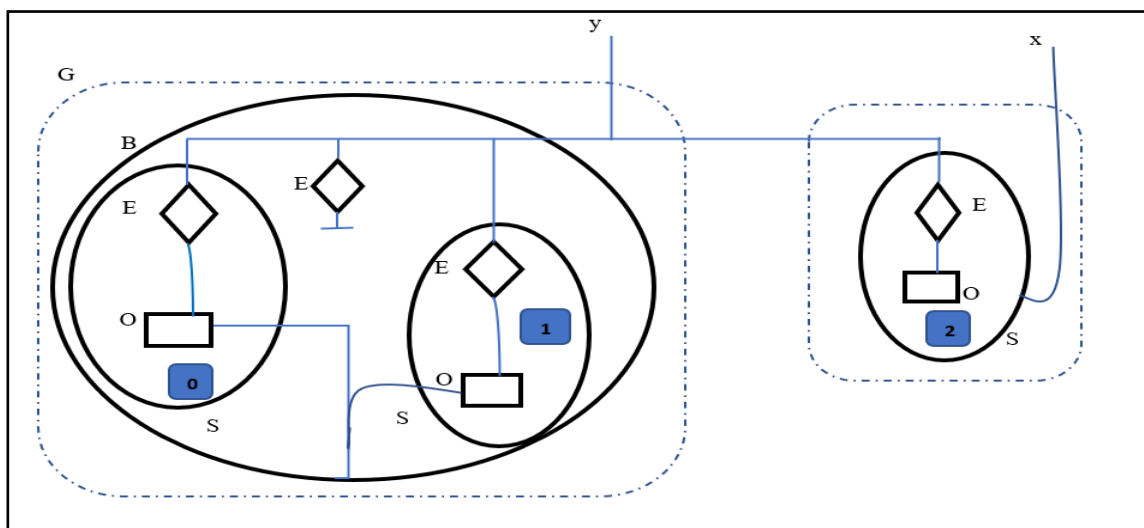


Figure 3.5 - Exemple d'un Bigraphe

Exemple 2

Soit le bigraphe $G = (V ; E ; ctrl ; G^P ; G^L) I \rightarrow J$ présenté dans la figure 4.6. Il permet de modéliser un ensemble de portables 'Ph' avec des utilisateurs 'U' situés dans deux salles 'S' séparées S1 et S2. L'ensemble de nœuds $V = \{U1, U2, U3, Ph1, Ph2, Ph3, S1, S2\}$, l'ensemble des arcs est $E = \{A0, A1, A2, A3, A4\}$ présente les connexions existantes entre les utilisateurs et les portables. L'ensemble de signature pour chaque nœud est :

$Ctrl = \{(U1: 2), (U2: 2), (U3: 2), (Ph1: 2), (Ph2: 2), (Ph3: 2), (S1: 2), (S2: 4)\}$, Ctrl affecte à chaque nœud son arité. Le nombre de sites $m=0$. G possède deux régions $n=2$ représentées par des rectangles numérotés (0,1) ainsi qu'un nom de sortie x , son interface d'entrées-sorties sont respectivement $I = \langle 0, \theta \rangle$ et $J = \langle 2, X \rangle$. La figure 4.6 montre le bigraphe G avec son bigraphe de places et de liens.

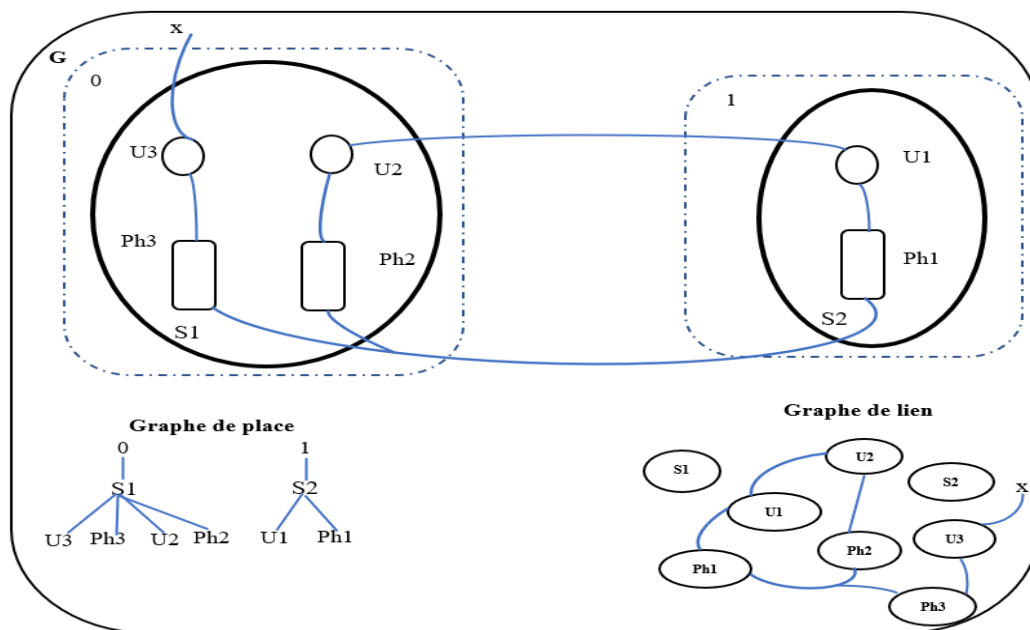


Figure 3.6 - Bigraphe G avec son graphe de places et de liens

3.2.5. Définition Bigraphical reactive system BRS

BRS est un méta-modèle expressif et flexible pour les systèmes ubiquitaires. La structure d'un système est représentée par un bigraphe montrant à la fois la localisation et la logique de connexions entre les entités puis le comportement est décrit à travers des règles de réactions. Deux opérations importantes dans un BRS : la composition et la transformation.

3.2.5.1.Composition

La composition de deux graphes $G = H \circ F$ consiste à remplacer les sites du graphe H par les racines de G. Plus formellement, $G_i = (V_i; \text{ctrl } i; \text{prnt}_i) : m_i \rightarrow m_{i+1}$ ($i = 0 ; 1$) soit un graphe de place avec $V_0 \cap V_1 = \emptyset$. Alors $G_0 \circ G_1 \stackrel{\text{def}}{=} (V; \text{ctrl}; \text{prnt})$ où $V = V_0 \sqcup V_1$, $\text{ctrl} = \text{ctrl } 0 \sqcup \text{ctrl } 1$, et $\text{prnt} = (\text{Id}_{V_0} \sqcup \text{prnt}_1) \circ (\text{prnt}_0 \sqcup \text{Id}_{V_1})$.

L'identité d'un graphe de places de m est $\text{id}_m \stackrel{\text{def}}{=}} (\emptyset, \emptyset_K, K, \text{Id}_m) : m \rightarrow m$. le produit tenseur de deux graphes de places $G : K \rightarrow \ell$ et $H : m \rightarrow m$ avec un ensemble de nœuds disjoints est : $G \otimes H : \kappa + m \rightarrow \ell + n$. Dans le paragraphe prochain, nous décrivons le graphe de liens. La figure 3.7 montre la composition entre les deux bigraphes H et F.

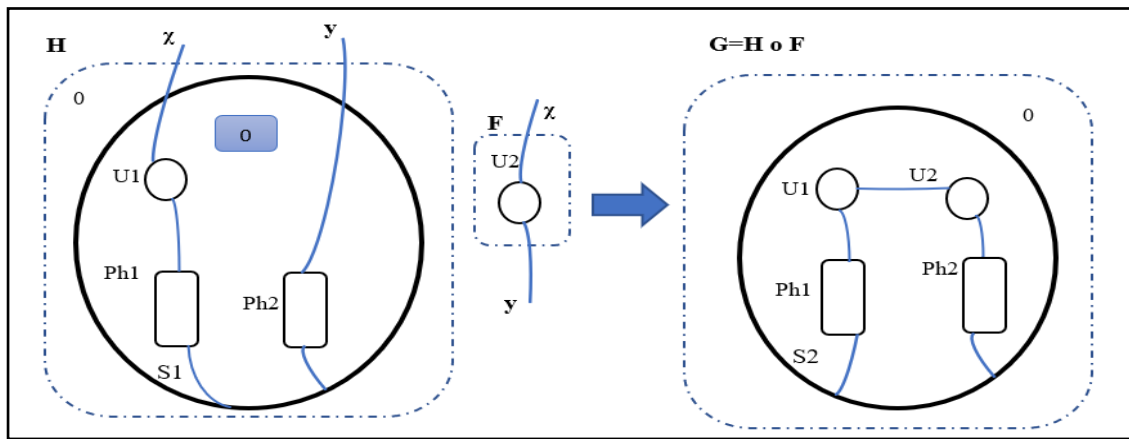


Figure 3.7 - Composition ($H \circ F$)

3.2.5.2.Transformation

Deux types de transformations sont possibles sur des bigraphes. La transformation de places représente l'arrivée ou le départ d'une entité. La transformation des liens, exprime la connexion ou la déconnexion d'un nœud à travers l'une de ses interfaces internes ou externes.

L'aspect dynamique est défini par des règles de transformation, appelées Règles de réaction. Une règle de réaction est un couple de bigraphes : **Redex** représente les **préconditions d'une réaction** et **Reactum** est la post-condition indiquant comment la réaction va changer le modèle. Une règle de réaction est définie par un couple de bigraphes $(R, R') = (R : m \rightarrow J; R' : m' \rightarrow J; \mu)$ où R est appelé bigraphe Redex et R' est nommé le bigraphe Reactum et $\mu : m' \rightarrow m$ établit la correspondance entre le nombre de sites des deux bigraphes Redex et Reactum.

La figure 3.8. met en avant une règle de réaction.

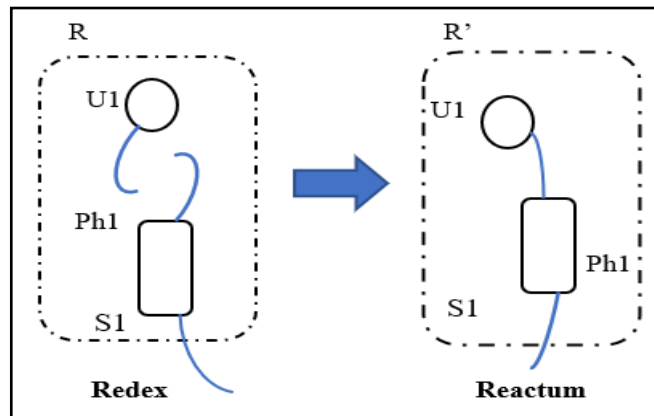


Figure 3.8 - Règle de réaction

3.3. BRS et la modélisation des systèmes sensibles au contexte

Les systèmes sensibles au contexte sont des systèmes complexes, prenant les sources de contexte en essayant de déduire un ensemble significatif de services. La création d'un modèle pertinent reste toujours un grand défi devant être surmonté par la communauté scientifique. A ce stade, plusieurs travaux ont été réalisés (voir chapitre 1). La construction d'un système sensible au contexte nécessite de :

- Connaître les différentes ressources de contexte ;
- Savoir le type de l'information contextuelle ;
- Avoir une idée sur la répartition de ces ressources ;
- Connaître la relation entre ces informations.

Les ressources de contexte peuvent être de différents types (environnement, capteurs, interactions, ...), elles peuvent être aussi réparties dans plusieurs endroits, ainsi connaître l'espace de ces ressources et leurs relations est extrêmement important pour la modélisation et la conception de ce système sensible au contexte.

Le formalisme BRS est un formalisme très puissant, il est capable de montrer à la fois la localisation des composants et les liens entre eux. On peut dire aussi que le BRS offre les moyens nécessaires pour présenter en même temps l'aspect structurel et l'aspect comportemental d'un système donné.

Cependant, on peut déduire qu'on peut s'appuyer sur le BRS pour déterminer : les ressources de contexte, ses répartitions ainsi que ses relations. En effet, plusieurs travaux ont utilisé ce formalisme afin de modéliser un système sensible au contexte. Dans ce qui suit, nous

montrons l'ensemble des travaux qui ont utilisé le BRS comme formalisme pour la modélisation des systèmes sensibles au contexte.

3.4. Travaux existants utilisent les BRS pour la sensibilité au contexte

L'introduction des méthodes formelles pour la modélisation des systèmes sensibles au contexte semble une tâche très ardue, ce qui explique le nombre limité de travaux publiés dans ce domaine. A cet égard, nous trouvons :

Le travail de [78], propose une modélisation formelle pour les systèmes sensibles au contexte en utilisant le formalisme BRS. Ils ont basé le travail sur la sensibilité à la localisation. Le modèle proposé représente l'union de trois BRS (C, A, P) où C montre le contexte, A est l'agent, P est un proxy. Lors de la modélisation, une condition doit être respectée : les trois BRS doivent être disjoints plus formellement : $C \cap A \cap P = \emptyset$

Dans ce travail, l'architecture proposée est une architecture à base de proxy où l'agent n'a pas accès directement au contexte. Ce travail reste une première tentative de modélisation des systèmes sensibles au contexte mais il n'a pas atteint la phase de vérification formelle.

L'équipe [79], tente de formaliser la structure et le comportement du système sensible du contexte en utilisant le formalisme BRS. Leur article apporte une carte d'équivalence entre les éléments de contexte et les composants de BRS pour rendre la formalisation de ces systèmes plus simple et plus pratique pour les concepteurs. Ce travail est très intéressant, mais il lui manque une partie de vérification.

Le travail de [80], présente une nouvelle vision des systèmes sensibles au contexte. Les auteurs constatent qu'un système sensible au contexte est la combinaison de deux parties : la partie contextuelle et la partie sensible au contexte, de sorte que les concepteurs doivent donner les parties souhaitables pour obtenir le modèle. Ils ont utilisé le formalisme bigraphe pour décrire l'idée et l'outil BigMC afin de vérifier certaines propriétés sur un exemple. Ce travail constitue un pas en avant dans la modélisation des systèmes complexes tels un système sensible au contexte mais nous trouvons une certaine ambiguïté pour comprendre le sens de la partie contextuelle (signalant l'absence totale du concept dans la littérature).

3.4.1. Synthèse

Après la présentation des travaux dans le domaine de la sensibilité au contexte en utilisant les BRS. Nous avons constaté que ces travaux :

- Suggèrent des modèles formels génériques pour le système sensibilité au contexte en ignorant la spécificité de certaines applications et certains domaines ;
- Utilisent des exemples de démonstrations basés seulement sur la sensibilité à la localisation bien que le contexte soit une notion plus large et plus étendue que la localisation (voir chapitre 1).

Les applications NFC sensibles au contexte sont des applications très spécifiques par ses domaines d’applications ou par ses modes de fonctionnement. Alors existe-il des modèles formels pour les applications NFC sensibles au contexte ? Pourquoi une modélisation formelle pour les applications NFC sensibles au contexte ?

Jusqu’à ce jour, aucun modèle formel n’a été présenté traitant des applications NFC et de leur modélisation. Par contre, la totalité des travaux expose une vérification d’un protocole de communication à base de NFC.

Après une phase de modélisation, un passage vers la vérification axée sur des moyens et des outils mathématiques est très important afin de détecter des erreurs ou/et des incohérences du modèle proposé. Dans ce qui suit, nous présentons la vérification de modèle et les outils de vérification des BRS disponibles.

3.5. Vérification formelle

La vérification formelle offre un ensemble de métriques mathématiques afin de s’assurer qu’un système donné satisfasse certaines propriétés. Elle consiste à comparer entre une représentation non ambiguë des propriétés et un modèle basé sur un langage formel. Deux approches de vérification formelle sont disponibles : vérification par *preuve formelle* qui est une méthode basée inférence ou démonstration mathématique très compliquée à appliquer sur des systèmes complexes, *vérification de modèle* (model-checking). Dans ce qui suit, on mettra en relief la vérification de modèle, les outils de vérification pour le BRS et l’outil BigMC. Ce dernier est un outil de vérification formelle pour les BRS à base de model checking.

3.5.1. Vérification de modèle (model-Checking)

C’est une approche reposant sur un algorithme d’énumération. Il consiste à déterminer toutes les situations possibles que peuvent rencontrer le système puis s’assurer qu’aucune d’entre-elles n’est en contradiction avec le comportement voulu. Ce type de vérification permet de détecter des failles, comme par exemple : le problème d’interblocage « deadlocks ». En effet, plusieurs types de model-checking existent dans la littérature par exemple le model checking temporel [81] est le premier type de model checking, où la spécification est exprimée dans une logique temporelle baptisée CTL (Computation Tree Logic) et la description à valider est décrite sous forme d’un système d’états de transitions. Enfin, pour vérifier une procédure de calculs des points fixes est appelée afin de déterminer automatiquement la conformité de la spécification et le système des états de transitions. Si l’outil s’arrête avec le résultat « True », cela veut dire que la spécification est conforme avec le système des états de transitions, dans le cas contraire un contre-exemple est délivré. L’inconvénient de cette approche est le problème d’explosion combinatoire.

Pour les modèles basés BRS plusieurs outils de vérification sont disponibles. Certains d’entre eux sont basés sur le model checking. Dans la section suivante on donnera un panorama sur l’ensemble des outils existants liés à la vérification de modèles BRS.

3.5.2. Différents outils de vérification pour les BRS

Le BRS est un formalisme très puissant appliqué à plusieurs domaines à l’instar de : la modélisation des systèmes sensibles au contexte, des services web, de la concurrence et de la mobilité, des structures basées agents, ... Pour cette raison, plusieurs laboratoires de recherche ont confectionné des outils et les ont mis à la disposition des chercheurs afin de leur faciliter la modélisation avec BRS. Le tableau 3.2 montre les différents outils BRS :

Chapitre 3. Formalisme Bigraphical Reactive System ‘BRS’

Nom	Caractéristique
BPLTool	a été l'une des premières implémentations et de simulation d'un bigraphe, il est écrit en langage SML [82], avec une phase de correction d'implémentation. Sans aucune notion de vérification des propriétés de système.
DBtk	C'est un outil de manipulation des bigraphes orientés. Ils sont une variante des bigraphes avec une structure de liaison orientée. Il n'offre pas une façon pour vérifier des propriétés de système. [83]
BigMC	Il a été créé par Perrone en 2012. C'est un outil de model checking pour BRS, est l'une des instanciations de modèle checker pour n'importe quel formalisme de BRS appliqué à tout domaine avec une grammaire spécifique. [84]
BigRed	C'est un outil graphique pour la modélisation d'un BRS avec une bibliothèque pour BigMC afin de donner la possibilité aux chercheurs de vérifier les propriétés des systèmes avec du model Checking. [85]
LibBig	C'est une bibliothèque Java, créée par le laboratoire de recherche MADS (<i>Models and Applications of Distributed Systems</i>), La bibliothèque a pour but d'offrir une implémentation pour les bigraphes et les BRS, mais elle offre aussi d'autres outils pour prendre en charge d'autres scénarios. Par exemple, les propriétés attachées peuvent être utilisées pour stocker des valeurs concrètes, telles que des adresses IP ou des autorisations d'utilisateurs [86].
BigraphER	Est un nouveau un environnement de modélisation et de raisonnement pour les bigraphes composé d'une bibliothèque OCaml. Cet outil permet la simulation et la visualisation des bigraphes et les bigraphes avec partage [87].

Tableau 3.2 - Outils de BRS

3.6. Conclusion

Dans ce chapitre, nous avons mis en relief l'importance de la modélisation formelle, puis nous avons abordé la notion de BRS dont nous avons expliqué sa structure générale, ses types, ...

Nous soulignons que le formalisme BRS est très puissant, il est capable de montrer à la fois la localisation des composants et les liens entre eux. Des pionniers l'ont introduit pour modéliser des systèmes sensibles au contexte. Ainsi, nous avons mis l'emphase sur des travaux publiés ayant utilisé les BRS pour la modélisation des systèmes sensibles au contexte. Nous

Chapitre 3. Formalisme Bigraphical Reactive System ‘BRS’

avons ensuite enchainé, avec la phase de vérification formelle. Un ensemble d’outils de vérification dédiés au formalisme BRS a été donné.

Après ce tour d’horizon, le temps est venu pour exposer notre approche qui consiste à proposer un modèle spécifique pour les applications NFC sensibles au contexte et basées sur les BRS, ce modèle est baptisé BigNFC. Comment créer ce modèle ? Quelle philosophie est-elle derrière sa confection ? Dans le chapitre suivant, nous allons essayer de déblayer voire de répondre à ces interrogations.

CHAPITRE 4

Conception et démonstration du modèle BigNFC

Sommaire

4.1. Introduction

4.2. Démarche de travail

4.3. Modélisation d'une application sensible au contexte sous NFC

4.4. Comment construire BigNFC ?

4.4.1. Equivalence entre le modèle BigNFC et les composants BRS

4.4.2. Passage vers le modèle de vérification avec BigMC

4.5. Exemple démonstratif

4.5.1. Spécification du système

4.5.2. Construction du modèle de validation en utilisant le BigNFC

4.5.2.1. Passage vers le modèle à base BRS

4.6. Conclusion

4.1. Introduction

La création d'un système sensible au contexte passe par plusieurs phases difficiles commençant toujours par des questions comme : comment capturer le contexte ? Par quel moyen va-t-il être capturé ? Sous quelle forme va-t-il être capturée ?

L'intégration de la NFC dans un environnement la rend plus interactive où elle peut jouer le rôle d'un collecteur de contexte. Cette technologie présente un atout remarquable de par sa simplicité et de sa sécurité quand bien même, elle rencontre plusieurs problèmes lors de son intégration. Les méthodes formelles reposent sur une grande rigueur et leurs utilisations sont quasiment incontournables. D'après ce qui a été présenté dans le chapitre précédent, l'utilisation des méthodes formelles dans le domaine de la NFC se réduit seulement à la vérification de protocole de communication, en négligeant la spécificité des applications à base de NFC.

Notre approche introduit l'aspect formel de modélisation des applications sensibles au contexte sous NFC en créant un modèle baptisé BigNFC, formé des termes bigraphes et NFC. Notre modèle est une représentation unique qui utilise la puissance du formalisme BRS afin de modéliser les composants ainsi que le comportement d'un système sensible au contexte sous NFC.

Dans ce chapitre, nous le débutons en premier par l'explication de la démarche de travail adoptée, nous présentons en second notre vision pour ce type d'applications, nous décrivons en troisième notre modèle formel BigNFC, nous montrons en quatrième l'équivalence entre les composants de notre modèle et les composants du formalisme BRS et nous appliquons en cinquième ce modèle sur un exemple démonstratif réel pour mesurer sa validité.

4.2. Démarche de travail

Dans cette thèse nous proposons une méthode de travail pour simplifier la tâche d'un concepteur souhaitant l'utilisation d'une méthode formelle pour la modélisation de son système sensible au contexte sous NFC. Notre démarche peut être synthétisée en trois étapes principales permettant de basculer d'une façon très simple et souple entre elles :

- **Etape 1** : Elle consiste à décrire une spécification d'un système sous NFC ;
- **Etape 2** : Elle s'occupe de la transformation de la spécification en un modèle de validation basé BRS en utilisant la carte/table de correspondance entre les composants BRS et les composants de l'application NFC ;

- **Étape 3** : Elle permet une autre transformation du modèle à base de BRS vers le modèle de vérification en respectant la grammaire BigMC, puis il faut vérifier la validité du modèle par rapport aux propriétés attendues. La figure 4.1 expose ces étapes.

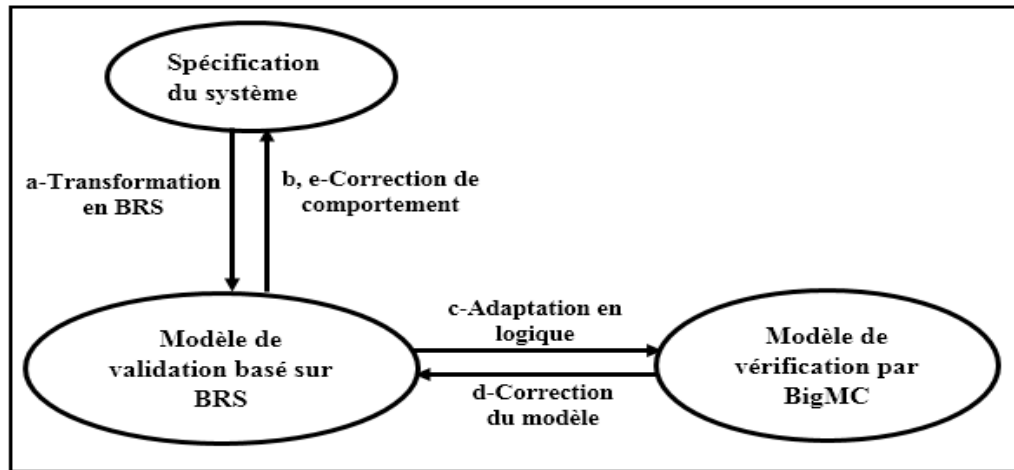


Figure 4.1 - Stratégie de travail

Remarque ?

La lettre ‘e’, indique la détection d’une erreur

Si une erreur est détectée, le concepteur doit faire un premier feedback vers le modèle de vérification c'est-à-dire qu’il faut voir sa transformation et reprendre la phase de vérification ;

Sinon un feedback bien profond doit être prévu au niveau du modèle de validation puis une reprise des différentes étapes avant vérification.

4.3. Modélisation d’une application sensible au contexte sous NFC

Dans la littérature, la plupart des articles connexes abordent souvent la vérification formelle sous l’angle sécuritaire pour tester l’aspect sécurité d’un protocole de communication à base NFC mais nous ne rencontrons pas un article traitant de la vérification comportementale de ces systèmes.

L’objectif principal d’un système sensible au contexte est d’obtenir et d’exploiter le contexte. Nous pensons que le contexte s’incarne dans trois formes différentes et ordonnées selon la nature de traitement.

- **Forme 1** : consiste à percevoir le contexte brut à partir de différentes ressources par exemple l’environnement, ...

Chapitre 4. Conception et démonstration du modèle BigNFC

- **Forme 2** : comporte l'aspect conversion de contexte brut vers un contexte unifié c'est-à-dire utiliser une méthode de représentation de contexte ;
- **Forme 3** : est la phase d'inférence où le contexte unifié se transforme en un contexte haut niveau, autrement dit un contexte exploitable.

La Figure 4.2 présente un schéma simplifié exhibant l'évolution de la forme de contexte et les phases de traitement.

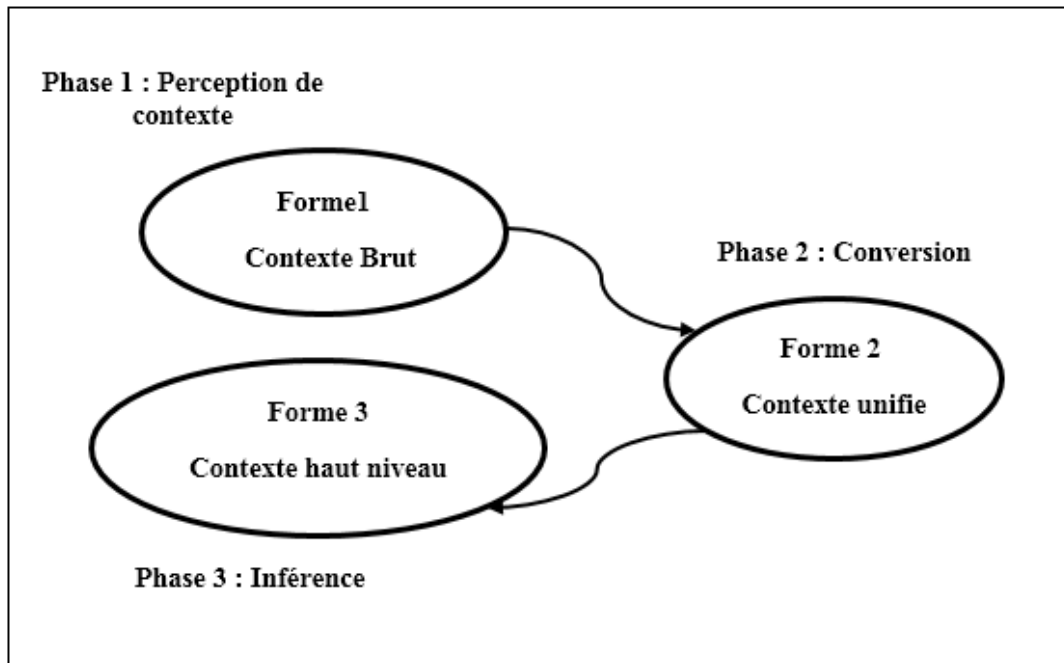


Figure 4.2 - Schéma simplifié des formes et des phases de traitement

Un système sensible au contexte sous NFC présente des spécificités liées à cette technologie de communication.

- **Percevoir (acquérir) du contexte brut** ; cette phase est assurée par l'un des modes de communication :
 - Tags NFC selon un mode R/W ;
 - Téléphone NFC avec un mode P-2-P ;
 - Téléphone NFC avec un mode EC.

Cette phase va être initiée par une touche par exemple une touche de tag NFC, pour être en mesure de lire les informations stockées sur le tag, ces informations sont censées représenter le contexte brut ;

- **Recevoir les informations contextuelles brutes NFC** ; le téléphone renvoie ces informations à la plateforme en vue de leur traitement (si le système reçoit des informations hétérogènes, une phase de filtrage est indispensable) ;
- **Traiter le contexte** ; après une phase de perception, la phase de traitement de contexte commencera par une conversion du contexte brut vers un contexte unifié (modèle compréhensible par le système) ;
- **Inférer le contexte de haut niveau** ; basé sur le contexte unifié le système va inférer le contexte en utilisant un moteur d'inférence ;
- **Délivrer un service personnalisé** ; après une phase d'inférence, un service personnalisé peut être restitué en un temps raisonnable.

4.4. Comment construire BigNFC ?

BigNFC est un modèle formel unique dédié aux applications sensibles au contexte sous NFC. Ce modèle utilise le formalisme BRS afin de décrire formellement les composants d'une application NFC.

Techniquement, un système sensible au contexte sous NFC peut être vu comme étant l'union de deux zones : une zone délivre le contexte et une zone le traite. De ce fait, nous pensons qu'une architecture producteur/consommateur présente bien la solution afin de séparer ces deux zones. La zone de production de contexte brut représente la zone physique contenant toutes les ressources de contexte, et la zone de consommation correspond aux différents niveaux de la plateforme intelligente.

BigNFC permet de modéliser un système sensible au contexte en faisant abstraction de quelques détails pour réduire la complexité.

- **Zone de production (zone physique)**, capable de délivrer un ensemble d'informations contextuelles. Dans une zone physique nous pouvons distinguer deux types de ressources de contexte :

Informations utilisateur ; ce sont des informations qui tournent autour de ses caractéristiques personnelles, sa vie sociale, sa localisation, ...

Objets intelligents ; représentent l'ensemble des gadgets installés dans l'environnement, capables d'acquérir et de communiquer des informations contextuelles tels que des capteurs, des technologies de communication (NFC, RFID, etc.).

Chapitre 4. Conception et démonstration du modèle BigNFC

- **Zone de consommation (plateforme intelligente)**, traite les informations contextuelles qui ont été fournies par la zone de production afin d'offrir des services adaptatifs aux utilisateurs. Elle contient au moins trois couches :

Couche de perception : Elle s'occupe de l'acquisition des informations contextuelles de l'environnement intelligent. Dans le système contextuel sous NFC, la perception d'un contexte brut provient du téléphone NFC de l'utilisateur après une touche réalisée sur les objets NFC. On peut étendre le rôle de cette couche au filtrage des données aussi s'il existe des ressources de perception hétérogènes.

Couche agent intelligent : Elle est assurée par un agent cognitif, qui est capable de décider d'une meilleure adaptabilité des services aux utilisateurs. D'abord, il compare les informations avec ses connaissances, puis, il déduit la situation en s'appuyant sur les règles d'inférence, ensuite, il met en exergue la disponibilité du service. Dans le système contextuel sous NFC, la détection d'informations contextuelles en tant que préférence obtenue à partir d'une touche réalisée par l'utilisateur dans un environnement intelligent, c'est-à-dire que la touche correspond à une détection de préférence ;

Couche de délivrance de service : Elle permet à une application sensible au contexte de fournir un service adapté à l'utilisateur. La figure 4.3 montre l'architecture générale d'un système sensible au contexte sous NFC.

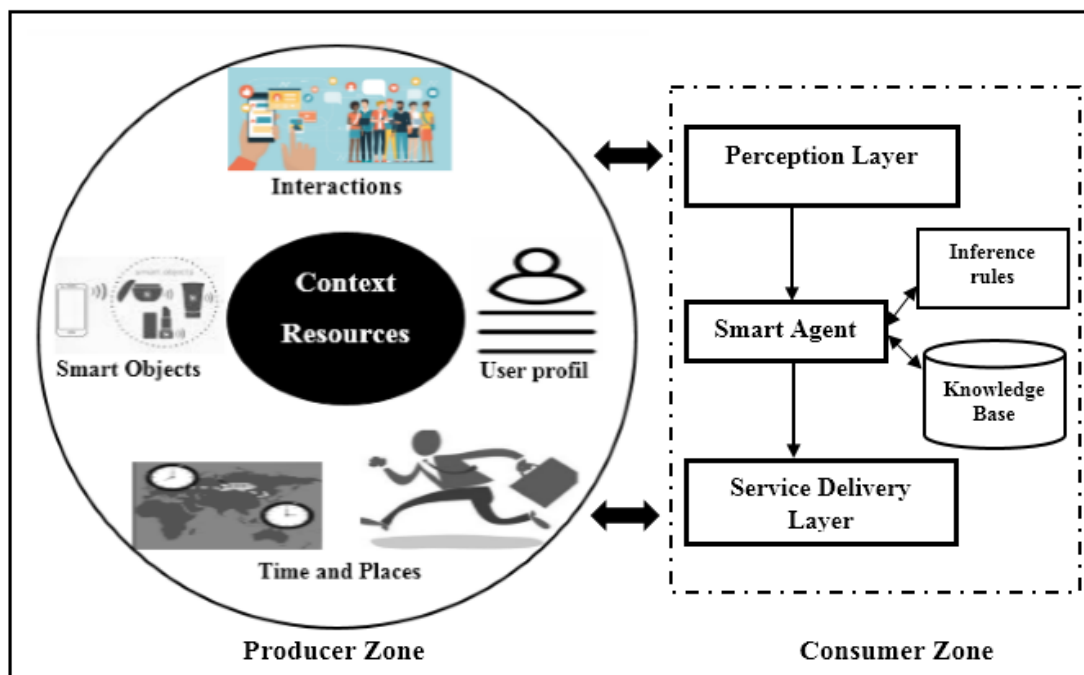


Figure 4.3 - Architecture générale d'un système sensible au contexte sous NFC

Chapitre 4. Conception et démonstration du modèle BigNFC

La construction du modèle BigNFC doit passer par la réalisation d'une équivalence entre BigNFC et le formalisme BRS. Dans ce qui suit nous présentons cette équivalence.

4.4.1. Equivalence entre le modèle BigNFC et les composants BRS

BRS est un formalisme mathématique très pratique, il offre un ensemble d'éléments (voir chapitre 3) permettant de présenter à la fois la connectivité et la localisation des entités avec deux représentations différentes : l'une sous une forme graphique et l'autre faisant appel à une grammaire spécifique. En sus, il permet de valider formellement l'idée en utilisant des outils mathématiques. Techniquement, nous utilisons le formalisme BRS pour établir :

- **Graphe de places** ; décrit la relation d'appartenance et d'inclusion entre les éléments de l'architecture ainsi que les informations contextuelles ;
- **Graphe de liens** ; montre la dépendance entre les informations et les transferts ;
- **Règles de réactions** ; utilisées pour définir le comportement souhaité du système.

Une carte d'équivalence a été réalisée pour aborder le mappage entre le modèle BigNFC et les composants BRS. Cette carte est puissante puisqu'elle peut être programmable, ce qui rend le traitement de cette étape plus simple et plus facile. Le tableau 4.1 dresse une correspondance entre le modèle BigNFC et les composants BRS.

BigNFC	Composants de BRS
Zone physique, plateforme intelligente.	Racine
Utilisateur, objets NFC, informations sur l'objet, relation entre objets, services, niveaux de la plateforme intelligente, informations contextuelles,...	Nœuds
Liens de communication, liens d'interaction	Arcs, arête.
Autres applications, capteurs, temps de transaction, ...	Sites
Comportements, perception, ...	Règles de réaction

Tableau 4.1 - Equivalence entre le modèle conceptuel et les composants BRS

Chapitre 4. Conception et démonstration du modèle BigNFC

D'une manière globale, un système sensible au contexte sous NFC peut être écrit sous cette forme en utilisant la grammaire BRS :

```
Zone_de_production [point de connexion].(objet_NFC[communication] )|$0) ||  
Zone_de_consommation[point_d'entrée, point_de_sortie].(niveaux[x,y] )|$1)
```

4.4.2. Passage vers le modèle de vérification avec BigMC

Après avoir obtenu le modèle de validation, un passage vers un modèle de vérification est indispensable. Ce passage exige une transformation de plus ou moins fidèle. La création du programme BigMC en respectant sa grammaire (voir chapitre5) devrait commencer avant le lancement d'une étape de vérification.

La bonne assimilation du modèle BigNFC ainsi notre démarche requiert une étape d'illustration. Dans ce qui suit nous montrons comment pouvons-nous utiliser le modèle BigNFC sur un exemple démonstratif.

4.5. Exemple démonstratif

4.5.1. Spécification du système

La plupart du temps dans l'état de l'art, l'ensemble des exemples traitant de la sensibilité au contexte prennent la sensibilité à la localisation comme un champ d'application en négligeant que ce concept est plus vaste et plus complexe que la localisation. De ce fait, nous validons notre modèle BigNFC, en le construisant autour d'un exemple sur la recommandation intelligente en utilisant la NFC comme un moyen pour la détection des préférences.

NFC permet d'obtenir des informations contextuelles d'une façon intuitive en utilisant ses trois modes de communication (se reporter chapitre 2 pour plus de détails). Dans notre exemple, un utilisateur est capable d'obtenir des informations sur un livre puis avoir une recommandation intelligente. Dans la zone de production, un poster intelligent contient un tag NFC (mode lecture / écriture), contenant l'ensemble des informations du contexte brut, ce sont les informations pertinentes qui caractérisent le livre à l'instar : le titre, l'auteur, le type, ... Lorsque l'utilisateur touche un poster intelligent, un ensemble d'informations importantes s'affichent sur le téléphone de l'utilisateur NFC. Ensuite, le processus de recommandation intelligent sera exécuté automatiquement si le livre est le type approprié et ayant une note élevée. Une recommandation intelligente consiste à détecter la similarité entre les utilisateurs pour pouvoir adapter et fournir le service recommandé.

Chapitre 4. Conception et démonstration du modèle BigNFC

Pour ce faire, l'application doit savoir : quel type de livre l'utilisateur préfère-t-il ? Les préférences sont déduites après l'action touche sur le poster intelligent.

4.5.2. Construction du modèle de validation en utilisant le BigNFC

En appliquant le modèle BigNFC, la structure est la suivante : La zone de production « zone physique » représente l'ensemble des éléments de l'environnement qui entourent l'utilisateur, nous citons par exemple le poster intelligent, le tag NFC, ... De plus, la zone de consommation « la plateforme intelligente » comprend différentes couches de la plateforme. Il est à noter justement que nous faisons abstraction de certaines informations pour simplification et la lisibilité de la tâche de modélisation.

4.5.2.1. Passage vers le modèle à base BRS

Le passage vers le modèle BRS nécessite l'utilisation d'une carte d'équivalence entre les composants de système et les éléments BRS. Dans un premier temps, nous donnons séparément la carte d'équivalence entre la structure et les éléments du formalisme BRS. Puis nous dessinons distinctement le bigraphe des deux niveaux et enfin, nous écrivons les deux formules modèles selon la grammaire donnée par Milner (voir chapitre4) La description du comportement nécessite l'écriture de plusieurs règles d'interaction, nous avons choisi quatre d'entre-elles (touche d'un tag, perception du contexte brut, ...).

Zone de production (zone physique)

La zone de production représente l'ensemble des éléments constituant l'environnement y compris les objets NFC, ces éléments sont en mesure de fournir un ensemble d'informations caractérisant une situation autour d'un utilisateur. Le tableau 4.2 présente le mappage entre les éléments de la zone de production et les composants du bigraphe.

Chapitre 4. Conception et démonstration du modèle BigNFC

Zone de production	Composant de bigraphe
Zone physique (environnement)	Racine
Utilisateur, tag NFC, poster intelligent, livre, type de livre, téléphone NFC,...	Nœuds
Liens de communication, liens d'interaction.	Arcs, arête
Autres applications, capteurs, temps de transaction, ...	Sites

Tableau 4.2 - Equivalence entre la zone de production et les composants de bigraphe

En se référant sur cette carte d'équivalence, nous sommes en mesure de dessiner le bigraphe correspondant à notre spécification. La figure 4.4 décrit l'état initial de la zone de production par un bigraphe.

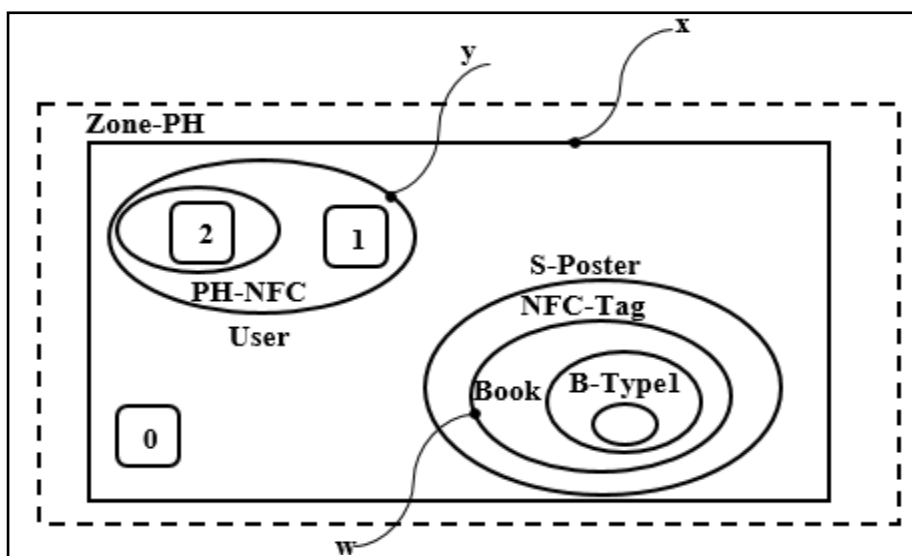


Figure 4.4 - Etat initial de la zone de production

Nous interprétons l'état initial de la zone de production en utilisant une formule respectant la grammaire du bigraphe.

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

Zone de consommation (plateforme intelligente)

La zone de consommation comporte les différents niveaux composant une plateforme intelligente capable d'intercepter, de déduire et de fournir un service personnalisé à l'utilisateur. Le tableau 4.3 établit la carte d'équivalence entre la zone de consommation et les éléments du bigraphe.

Chapitre 4. Conception et démonstration du modèle BigNFC

Zone de consommation	Composant de bigraphe
Plateforme intelligente	Racine
Couche de perception, couche délivre le service, service, relation entre entités, des informations sur une entité, ...	Nœuds
Liens de communication, liens d'interaction.	Arcs, arête
Autres couches, des éléments d'abstraction, ...	Sites

Tableau 4.3 - Equivalence entre la zone de consommation et les éléments de bigraphe

A partir de cette carte d'équivalence, nous serons capables de dessiner le bigraphe montrant la zone de consommation dans son état initial. La Figure 4.5 illustre le bigraphe correspondant à l'état initial de la zone de consommation.

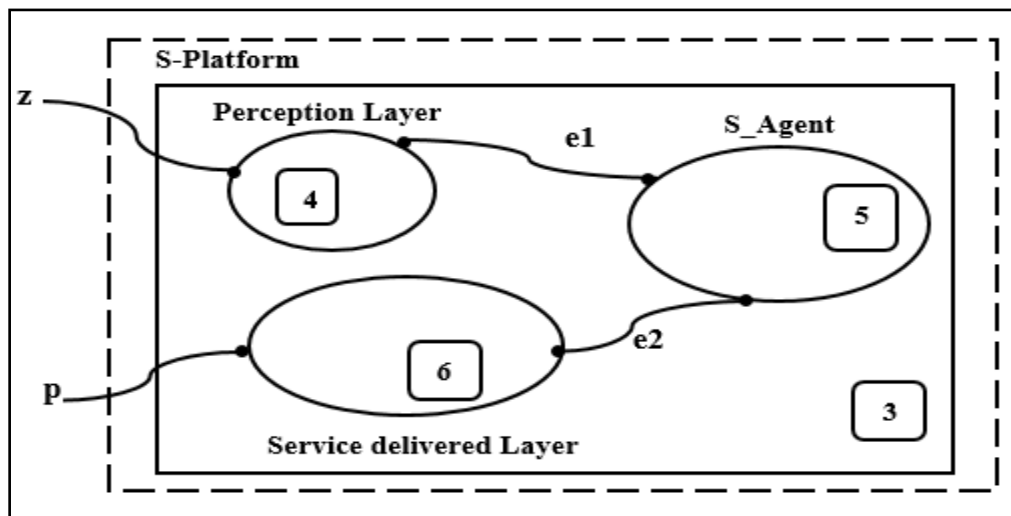


Figure 4.5 - Bigraphe de l'état initial de la zone de consommation

En utilisant la grammaire de bigraphe, nous obtenons la formule qui montre l'état initial de la zone de consommation :

S-Platform. (\$3 | Perception-Layer [z, e1]. \$4 | S-Agent. [e1, e2]. d5 | Service delivered-Layer. \$6)

Enfin, la concaténation des deux formules (zone de production/ zone de consommation) permet d'obtenir la formule globale de l'exemple démonstratif.

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).

Règles d'interactions

La représentation d'un système peut être effectuée en prenant en compte deux aspects : l'aspect statique, qui montre l'ensemble des composants formant le système et l'aspect dynamique, permettant de montrer son comportement. Le formalisme BRS utilise les règles d'interactions pour montrer la dynamique d'un système.

Première règle d'interaction : action touchez tag

Dans un système sensible au contexte sous NFC, l'action touchez tag, est une action majeure pour démarrer la phase de traitement. Lorsque l'utilisateur touche le tag NFC, cela permet de transférer le contexte brut vers le téléphone de l'utilisateur NFC en usant du mode lecteur. Dans notre cas, le contexte brut (type de livre) correspond aux données du livre (titre, noms des auteurs et type de livre, ...). Rappelons que :

- En un seul geste, le téléphone NFC ne peut lire qu'un et un seul tag NFC ;
- Le tag sélectionné, implique une préférence pour un type de livre ;
- La totalité de cette action s'effectue au niveau de la zone de production.

La figure 4.6 montre la règle touchez tag.

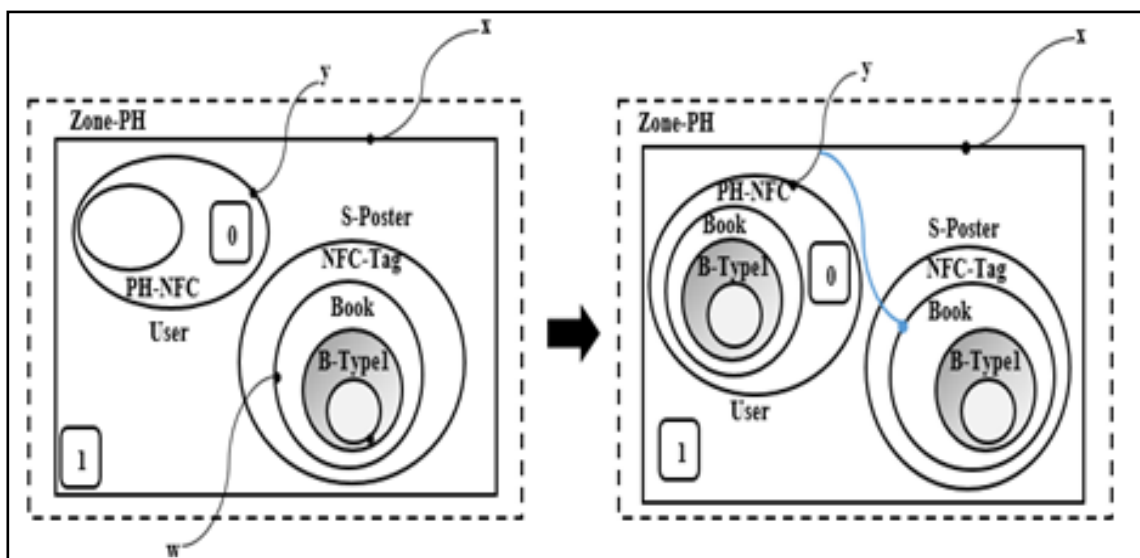


Figure 4.6 - Règle d'interaction : action touchez tag

Deuxième règle d'interaction : interception du contexte brut

Au niveau de cette règle, nous essayons d'exprimer la communication entre la zone de production fournissant le contexte brut et la zone de consommation qui le traite (plateforme intelligente). Cette communication doit garantir le transfert du contexte brut depuis le téléphone

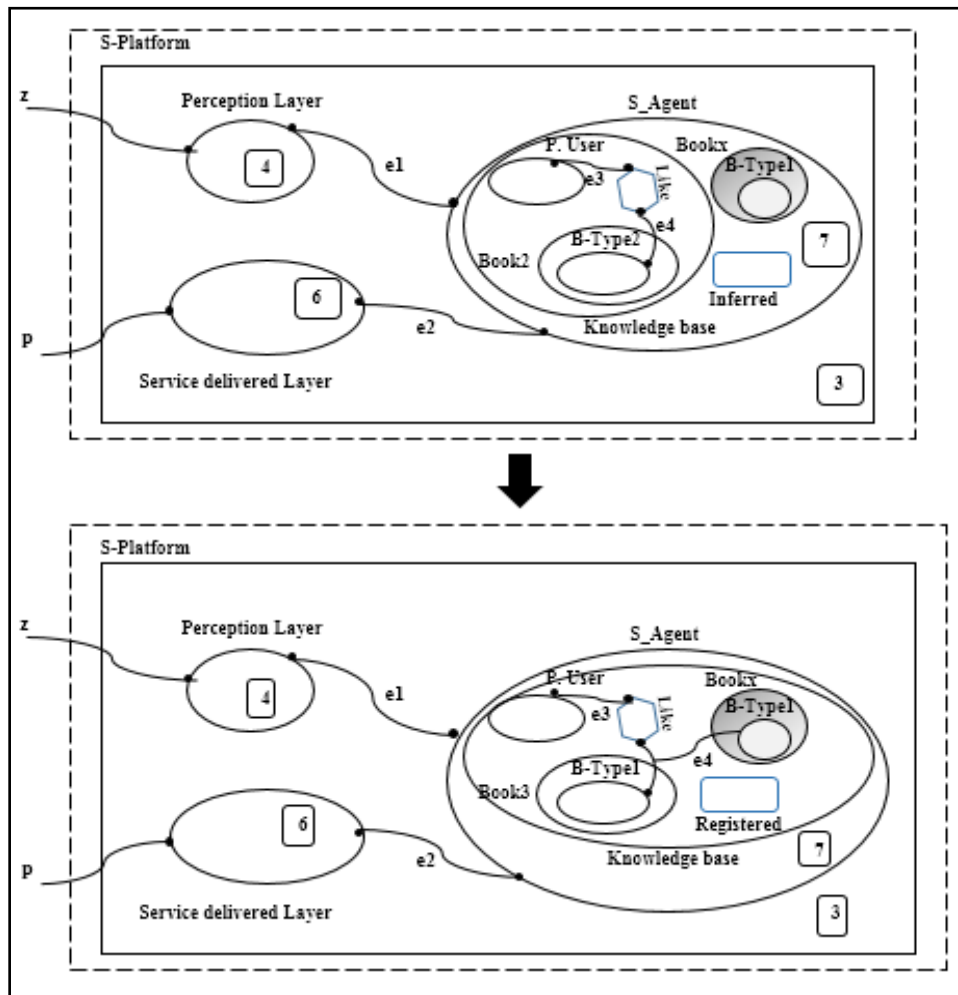


Figure 4.8 - Règle d'interaction : déduction des préférences

Quatrième règle d'interaction : livres recommandés

Après un long traitement accompli sur le contexte brut et ses différents passages entre les couches. La zone de consommation (plateforme intelligente) peut enfin délivrer la liste des livres pouvant être dans le centre d'intérêt de l'utilisateur. Pour ce faire, elle doit réaliser un pont de communication avec le téléphone de l'utilisateur plus précisément ; ainsi la couche délivrant le service doit ouvrir un canal de communication avec le téléphone de l'utilisateur afin d'envoyer la liste. La figure 4.9 relate la réalisation d'un point de communication entre la couche livraison et le téléphone de l'utilisateur.

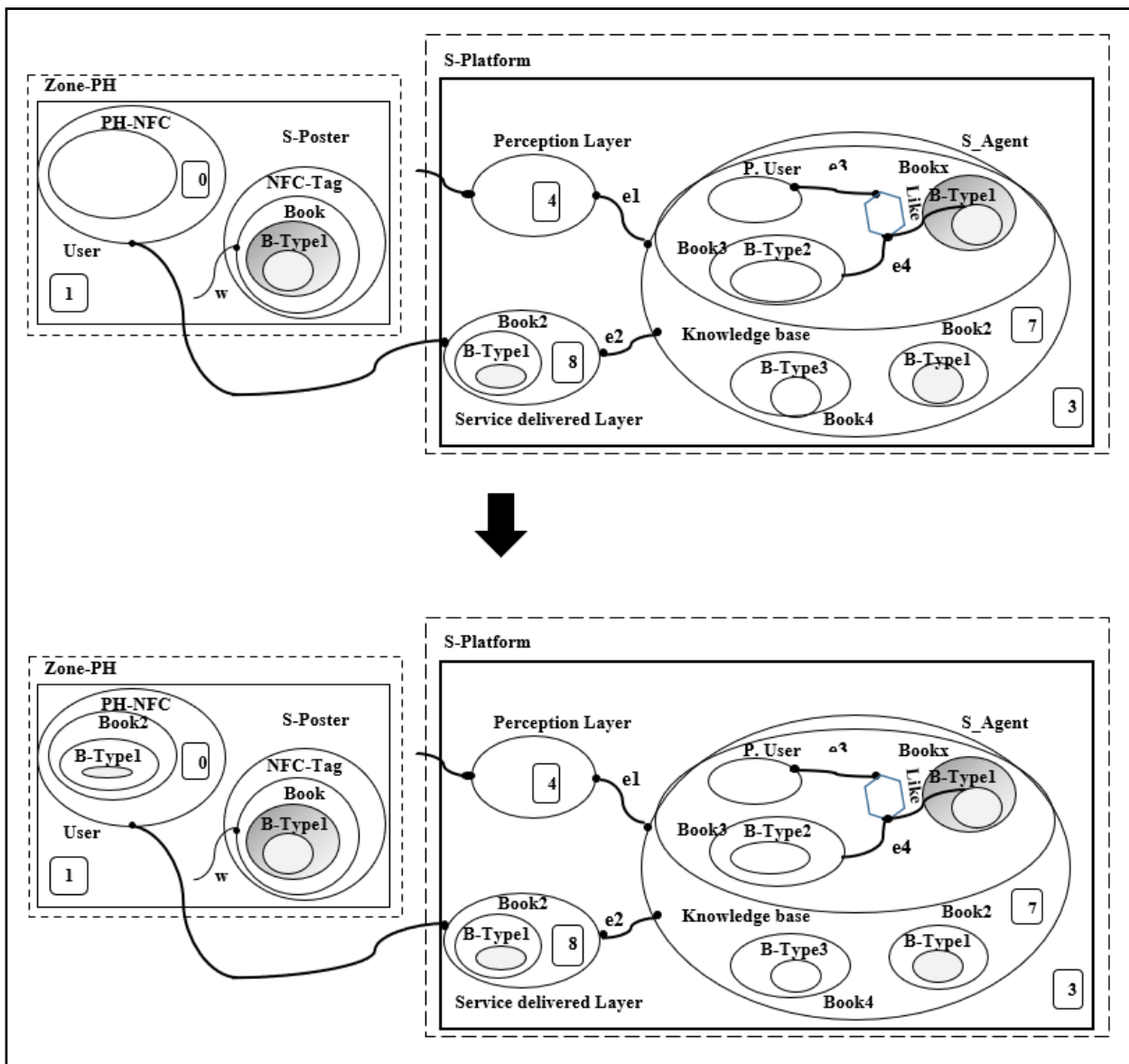


Figure 4.9 - Règle d'interaction : livres recommandés

Après cette description et cette explication de l'exemple démonstratif, nous passons à la phase de validation en utilisant l'outil BigMC.

4.6. Conclusion

Dans ce chapitre, nous avons présenté d'abord, notre approche de travail. Cette démarche peut être résumée en trois étapes : la première demande une spécification du système sous NFC ; la deuxième c'est une phase de transformation de la spécification vers un modèle de validation basé BRS ; la troisième comporte le passage du modèle de validation vers un modèle de vérification en respectant la grammaire BigMC.

Chapitre 4. Conception et démonstration du modèle BigNFC

Puis, nous avons expliqué notre visualisation des applications sensibles au contexte sous NFC où nous pensons que le concepteur de ce type d'application doit déterminer à prime abord le rôle à jouer par la NFC dans son environnement.

Ensuite, nous avons répondu avec plus de détails à la question la plus évidente faisant l'objet de ce chapitre à savoir : Comment construire le modèle BigNFC ? Dans la réponse, nous avons essayé mis en exergue : les concepts, les cartes d'équivalence entre la spécification et le modèle de validation BRS, les étapes, etc.

Enfin, nous avons proposé une démonstration du modèle BigNFC sur un exemple, sur lequel, nous tentons de montrer pas à pas son applicabilité.

Suite à cette exploration du modèle sur un exemple, il est opportun de passer à une nouvelle étape dont l'objectif est l'application de l'outil BigMC sur lui en cherchant à vérifier un ensemble de propriétés. En effet, le prochain chapitre sera l'avènement de tout ce qui a précédé et constituera une validation de nos propos.

CHAPITRE 5

Vérification formelle du modèle BigNFC

Sommaire

5.1. Introduction

5.2. Approche de vérification

5.3. C'est quoi le langage BigMC ?

5.3.1. Structure d'un programme BigMC

5.3.2. Termes du langage BigMC

5.3.2.1. Contrôles

5.3.2.2. Noms

5.3.2.3. Règles de réaction

5.3.2.4. Définition de modèle

5.3.2.5. Définition des propriétés

5.4. Validation de l'approche

5.4.1. Transformation de l'exemple démonstratif en programme BigMC

5.4.1.1. Programme BigMC de la zone de production

5.4.1.2. Programme BigMC de la zone de consommation

5.4.1.3. Programme BigMC des règles d'interaction

5.4.1.4. Programme BigMC du modèle général

5.4.1.5. Définition des propriétés avec BigMC

5.5. Résultats de la vérification formelle par BigMC

5.6. Discussion

5.7. Conclusion

5.1. Introduction

La validation est une phase cruciale dans la vie d'un modèle où cette dernière permet la détection des anomalies offrant la possibilité d'agir pour une correction rapide, efficace et à moindre coût. En effet, il existe plusieurs façons pour faire valider un modèle par exemple la validation par simulation, la validation par les tests, ...

La vérification formelle reste un moyen puissant pour la détection des anomalies surtout si nous parlons de domaines critiques tels que la surveillance d'un réacteur nucléaire, le contrôle de la température d'un four dans une industrie lourde, etc. Décidément, son rôle a été élargi pour atteindre tous les domaines informatiques. Malgré sa puissance, la vérification pose toujours des ennuis à cause de sa complexité et son fondement purement mathématique. Dans cette thèse, la phase de validation est assurée par un outil baptisé BigMC, où il est le premier outil proposant la vérification formelle du formalisme BRS, nous pensons que la compréhension de notre travail passe par une explication primordiale du langage BigMC ainsi que la structure des programmes.

Dans ce chapitre, nous expliquons d'abord le processus de notre approche de vérification après nous essayons de décrire en détail le langage BigMC, puis nous passons à la présentation de la structure générale d'un programme BigMC, ensuite nous proposons une transformation de notre modèle BigNFC vers un programme BigMC, enfin nous continuons la transformation et la validation de notre exemple (abordé au chapitre 4) en s'assurant de sa satisfaction de certaines propriétés par exemple l'absence d'interblocage, ...

5.2. Approche de vérification

Dans cette thèse, nous optons pour une approche de vérification qui s'articule sur trois phases :

Première phase : Elle consiste à traduire le modèle décrit en BRS en un programme BigMC en respectant sa grammaire (description détaillée dans 5.2.2) ;

Deuxième phase : c'est le cœur de cette approche. Elle consiste à faire passer le programme BigMC (issue de la phase précédente) et la spécification des propriétés au moteur de vérification de l'outil BigMC ;

Troisième phase : à l'issue de cette phase, le moteur de l'outil rend son verdict après un certain nombre d'étapes définies au préalable. Deux cas sont possibles, une terminaison avec :

- {

Premier cas : succès si la propriété est vérifiable ;

Deuxième cas : échec si la propriété n'est pas vérifiable (contre-exemple offert).

La figure 5.1 montre le processus de vérification de notre approche.

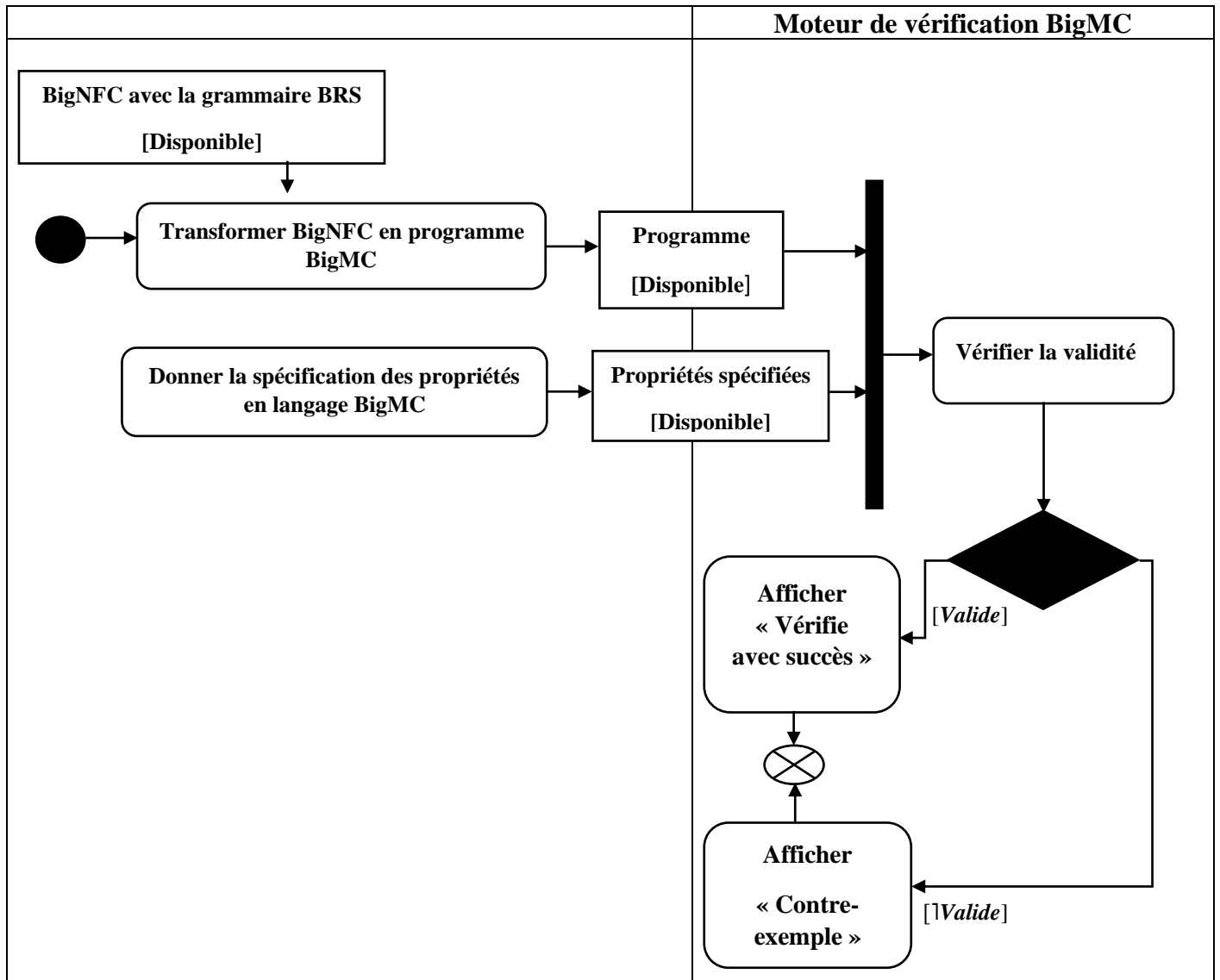


Figure 5.1 - Processus de vérification de notre approche

5.3. C'est quoi le langage BigMC ?

BigMC correspond à l'abréviation de *BiGraph Model Checker* [88]. C'est un vérificateur de modèle conçu pour vérifier le modèle BRS, ce formalisme a trouvé un grand succès pour la modélisation d'une variété de systèmes par exemple les systèmes biologiques, des systèmes sensibles au contexte, des processus métier, ... L'utilisation de BigMC permet de s'assurer qu'une spécification est correcte pour un modèle de BRS en explorant tous les états possibles du système.

Le problème avec ce style d'outils reste toujours l'explosion combinatoire de l'espace des états qui limite la taille du système à vérifier. En effet, l'utilisation de BigMC nécessite la transformation du modèle en un programme structuré. Dans ce qui suit, nous détaillons la structure générale d'un programme BigMC, ses termes, etc.

5.3.1. Structure d'un programme BigMC

La structure d'un programme BigMC comporte quatre parties :

Première partie est consacrée à la définition des noms des nœuds formant le modèle avec leurs contrôles et le nombre de ports d'entrée/sortie pour chaque nœud ;

Deuxième partie est réservée à la description d'un ensemble de règles de réaction de la forme $A \rightarrow B$ où A est le redex et B est le réactum.

Troisième partie est dédiée à la représentation du modèle dans une grammaire spécifique ;

Quatrième et dernière partie est dévolue au rajout de quelques propriétés ensuite les vérifier sur le modèle décrit. La description de la propriété commence toujours par le mot clé *%property*. Le langage BigMC propose des prédicats prédéfinis pour exprimer les propriétés à l'instar : *\$empty()*, qui est équivalent à la propriété taille vide, *\$terminal()*, ce prédicat est appliqué sur l'ensemble des états marqués terminaux. Si la propriété échoue, un contre-exemple est donné. Un programme BigMC se termine par le mot clé *%check* pour débiter la phase de vérification. La figure 5.2 montre la structure générale d'un programme BigMC.

```
#Partie 1 (définition des nœuds)
Control nom_de_nœud : arité ;
#Partie 2 (définition des règles d'interaction)
A → B;
#Partie 3 (définition du modèle)
nom_de_racine1[nom_de_porte].(nom_de_noeud.$0)| nom_de_racine2.($1) ;
#Partie 4 (définition des propriétés)
%property nom_propriété expression_propriété ;
%check ;
```

Figure 5.2 - Structure générale d'un programme BigMC

5.3.2. Termes du langage BigMC

Chaque langage a plusieurs termes, pour une bonne compréhension de notre travail, nous exposons les différents termes de ce langage.

5.3.2.1. Contrôles

Le formalisme BRS est un moyen permettant de définir des caractéristiques supplémentaires pour un nom (racine ou un nœud), dans le langage BigMC. Ces caractéristiques constituent le *contrôle*. Il existe deux types de contrôle : un contrôle actif et un contrôle passif.

5.3.2.2. Noms

BigMC, ne fait pas de distinction entre la racine et les nœuds, il les appelle des *noms*. On peut préciser que le nom est un nom externe en rajoutant le mot clé *%outer*. La figure 5.3 présente la grammaire de BigMC :

```

T ::= K.T
T ::= T | T
T ::= T || T
T ::= $n
T ::= K
T ::= nil

K ::= k[names]
K ::= k

names ::= n , names
names ::= n

n ::= [a-zA-Z] [a-zA-Z0-9]*
n ::= -
    
```

Figure 5.3 - Grammaire de BigMC

5.3.2.3. Règles de réaction

Une règle de réaction est écrite sous la forme $T \rightarrow T$; où le premier membre de la règle représente le redex et le deuxième membre représente le reactum dans le formalisme BRS. Si la règle de réaction est complexe, il est raisonnable d'être très attentif lors de sa formulation de la décomposer dans la mesure du possible en plusieurs actions simples par exemple la règle d'action *touche NFC* peut être décomposée en trois règles de réaction élémentaire : connexion

entre les deux appareils NFC, transfert de données et déconnexion entre les deux appareils NFC ;

5.3.2.4. Définition de modèle

Dans le langage BigMC, un modèle s'écrit sous la forme T ; Le modèle à vérifier doit se présenter à la fin du fichier avant le mot clé *%check*, et après toutes les définitions dont il dépend.

5.3.2.5. Définition des propriétés

Après avoir défini le modèle, il est temps de passer à la phase de vérification qui doit être débutée par la définition des propriétés. Dans le langage BigMC, une propriété doit être présentée sous la forme suivante :

%property nom_{propriété} expression_{propriété} ; le nom de la propriété est une concaténation alphanumérique où il est souhaitable qu'il soit significatif car si le vérificateur trouve une violation (propriété fautive), il va le signaler en indiquant de façon claire la propriété concernée. Un fichier '.bgm' peut contenir plusieurs types de propriétés où l'ordre d'apparition est très important. Il est vraiment judicieux de placer en premier les propriétés qui vont potentiellement échouer.

Le langage BigMC propose une variété de prédicats prédéfinis afin de permettre à l'ensemble des utilisateurs de les manipuler plus facilement lors de la rédaction d'une propriété. Dans ce qui suit, nous présentons d'une manière non exhaustive l'ensemble des prédicats prédéfinis :

- *\$pred()* représente le prédécesseur de l'état actuel, autrement dit, qu'il existe une étape de réaction précédente à l'état actuel ;
- *\$this ()* indique l'état actuel ;
- *\$succ()* permet de déterminer l'ensemble des états successeurs, c'est-à-dire ceux qui sont atteignables par une étape de réaction à partir de l'état courant ;
- *\$terminal ()* ; Le prédicat en question ne s'applique qu'aux états marqués terminaux, c'est-à-dire qu'ils ne conduisent à aucun autre état et qu'aucune réaction ne peut leur être appliquée. *\$terminal* $\rightarrow p()$ retourne vrai s'il est appliqué à des états non terminaux, sinon il retourne $p()$.

Après avoir terminé succinctement l'explication et la définition du langage BigMC, nous passons à présent à l'explication de son application sur notre approche. Dans cette partie, nous tentons d'expliquer le passage de l'exemple démonstratif décrit en BRS (voir le chapitre 4) vers le langage BigMC.

5.4. Validation de l'approche

La validation est une phase primordiale, dans ce travail nous adoptons une stratégie de validation construite autour du langage formel BigMC. Ainsi, nous passons à la phase de transformation du modèle décrit précédemment en un programme BigMC

Pour rappeler le travail qui a été fait, nous exposons le modèle en respectant la grammaire BRS.

Le modèle BigNFC est un nouveau modèle formel dédié aux applications sensibles au contexte sous NFC basé sur le formalisme BRS. L'idée de sa construction a été inspirée de l'architecture producteur/ consommateur où la zone de production responsable sur la livraison des informations contextuelles et la zone de consommation va traiter ces informations afin de fournir un service adapté aux besoins des utilisateurs.

Dans ce travail, nous traitons les gadgets sous NFC (objets NFC) comme étant des collecteurs et des fournisseurs du contexte détecté à partir de l'environnement de l'utilisateur en déterminant sa localisation par exemple, ou déduire ses préférences en touchant un tag intégré dans un poster, ... pour la bonne construction de ce modèle la détermination du mode de communication dès le départ est importante. BigNFC peut être vu comme une concaténation des deux zones (zone de production de contexte et zones de consommation (plateforme intelligente)) ; plus techniquement nous utilisons le formalisme BRS où le graphe de place décrit la relation d'inclusion entre les éléments de l'architecture et les informations contextuelles ; le graphe de liens montre la dépendance entre l'information et son transfert ; les règles de réactions montrent l'aspect dynamique (comportemental) du système.

BigNFC la zone de production et la zone de consommation sont vus comme étant des racines les gadgets NFC comme étant des nœuds avec une porte de communication avec la zone de consommation. Dans ce qui suit nous présentons le modèle BigNFC en accord avec la grammaire BRS.

```
Zone_de_production [point_de_connexion].(objet_NFC[communication])|$0) ||  
Zone_de_consommation[point_d'entrée, point_de_sortie].(niveaux[x,y] )|$1)
```

5.4.1. Transformation de l'exemple démonstratif en programme BigMC

La phase de transformation du modèle vers un programme BigMC est une phase très importante où nous essayons de détecter les anomalies existantes en appliquant un ensemble de propriétés sur le cas à expérimenter.

En effet, nous allons essayer de respecter la structure générale d'un programme BigMC c'est-à-dire de prime abord ; nous déterminons les nœuds ainsi que leurs contrôles et le nombre de ports d'entrée / sortie constituant le cas étudié ; de plus, nous exposons les règles d'interaction exprimant l'aspect dynamique. Nous présentons le modèle général à vérifier et nous définissons l'ensemble des propriétés sur lesquelles le modèle va être validé.

Dans la présentation du programme BigMC correspondant, nous privilégions la présentation des programmes des deux zones (physique, intelligente) séparément pour offrir une meilleure lisibilité.

5.4.1.1. Programme BigMC de la zone de production

La zone de production (physique) est peut-être construite autour de six nœuds et trois noms ou connecteurs dans un langage plus simple. La figure 5.4 montre la transformation de la zone physique en programme BigMC.

```
%active ZonePH : 1;      %name x;  
%passive SPoster : 0;    %name y;  
%active PHNFC : 0;      %name w;  
%passive TagNFC : 1;  
%passive BooKx: 0;  
%active User: 1;  
%passive Type1: 1;  
ZonePH[-].((User[-].PHNFC)|(SPoster.TagNFC[-.BooKx.(Type1[-])));
```

Figure 5.4 - Programme BigMC de la zone de production

5.4.1.2. Programme BigMC de la zone de consommation

La zone de consommation (smartplatform) comporte quatre nœuds et trois contrôleurs. La figure 5.5 dresse le programme BigMC de cette zone.

```
%active DelivredLayer: 2;
%active SPlatform: 0;
%active PerceptionLayer: 2;
%active SAgent: 2;
%name e1;
%name e2;
%name
SPlatform.(PerceptionLayer[-,e1]|SAgent[e1,e2]|DelivredLayer[e2,-]);
```

Figure 5.5 - Programme BigMC de la zone de consommation

5.4.1.3. Programme BigMC des règles d'interaction

Dans cette partie, nous exposons l'ensemble des règles d'interaction en utilisant le langage BigMC. Dans un premier temps, nous intéressons à six règles d'interaction où :

- La première présente l'action touching entre téléphone et le poster intelligent ;
- La deuxième dévoile une présentation unique sur le mode de communication entre le téléphone et le tag NFC ;
- La troisième exprime la déconnexion entre le téléphone et le poster intelligent ;
- La quatrième montre le port de communication entre le téléphone portable et le niveau de perception le contexte brut ;
- La cinquième montre la liaison et le transfert des informations entre la zone physique et la plateforme intelligente ;
- La sixième présente la communication entre les différents niveaux de la plateforme intelligente ;

Règle 1 ;

ZonePH[-].(User[-].PHNFC|SPoster.TagNFC[-].\$1) ->
 ZonePH[-].(User[w].PHNFC.(\$0)|SPoster.TagNFC[w].\$1);

Règle 2 ;

ZonePH[-].(User[w].(PHNFC.\$0)|SPoster.TagNFC[w].\$1) ->
 ZonePH[-].(User[w].PHNFC.(\$1)|SPoster.TagNFC[w].\$1);

Règle 3 ;

ZonePH[-].(User[w].PHNFC.(\$1)|SPoster.TagNFC[w].\$1) ->
 ZonePH[-].(User[w].PHNFC.\$1|\$3);

Règle 4 ;

ZonePH[-].(User[w].PHNFC.\$1|\$3)|SPlatform.(PerceptionLayer[w,e1].(\$0)| \$4) ->
 ZonePH[-].(User[w].PHNFC.\$1|\$3)|SPlatform.(PerceptionLayer[w,e1].(\$1)|SAgent[e1,e2] \$5);

Règle 5 ;

ZonePH[-].(User[e2].PHNFC.\$1|\$3)|SPlatform.(PerceptionLayer[e2,x].(\$0)| \$4)->
 ZonePH[-].(User[e2].PHNFC.\$1|\$3)|SPlatform.(PerceptionLayer[e2,x].(\$1)|SAgent[x,y] \$5);

Règle 6 ;

SPlatform.(PerceptionLayer[w,e1]|SAgent[e1,e2].(Bookx.\$6|\$7)|\$5) ->
 SPlatform.(PerceptionLayer[e1,e2]|SAgent[e1,e2].(beliveslike.(\$6)|\$8)|\$5);

En effet, nous considérons que les six premières règles sont des règles basiques montrant le fonctionnement global du système. De ce fait, nous complétons notre travail en rajoutant d'autres règles plus complexes et imbriquées en faisant des abstractions sur quelques détails :

- La septième règle présente passage de l'information entre les différents niveaux de la plateforme intelligente ;
- La huitième exprime la phase de déduction ;
- La neuvième montre la mise à jour du profil utilisateur.

Règle 7 ;

SPlatform.(PerceptionLayer[e2,x].\$1|SAgent[x,y].(\$0)|\$5) ->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(\$1|\$7)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].\$1|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(\$1|\$7)|\$5);

Règle 8 ;

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(\$1|\$7)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|\$7)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|\$7)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$10)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$10)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$6)|\$5);

Règle 9 ;

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$6)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$6|\$8)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$6|\$8)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.\$6|KnowldgeB)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.(\$6)|KnowldgeB)|\$5)->
 SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.(\$6)|KnowldgeB.\$9)|\$5);

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.(\$6)|KnowldgeB.Puser.(LikeB.Book
 x.\$6)|Book.(Book3.Type1|Book2.Type2))|\$5)->

SPlatform.(PerceptionLayer[e2,x]|SAgent[x,y].(Bookx.\$6|beliveslike.(\$6)|KnowldgeB.Puser.(LikeB.Book
 x.\$6)|Book.(Book3.Type1|Book2

5.4.1.4. Programme BigMC du modèle général

Le programme BigMC du modèle général concatène deux zones ; une zone de production du contexte et une zone de consommation. La formule ci-dessous présente le modèle général utilisant la grammaire BigMC.

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

Dans ce qui suit, nous présentons la définition des deux propriétés à vérifier afin de garantir la validité et la non violation de ces propriétés par le modèle.

5.4.1.5. Définition des propriétés avec BigMC

Dans cette partie, nous donnons la définition de l'ensemble des propriétés dont l'objectif principal est de garantir la sûreté de système en prouvant l'absence d'interblocage en utilisant le prédicat prédéfini donné par le programme BigMC. Dans ce qui suit, nous donnons la définition de la propriété absence d'interblocage.

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

Une deuxième propriété qui a été prévue afin de garantir la validité de la règle cinq par le biais du prédicat prédéfini matches. Dans ce qui suit nous décrivons la propriété.

```
# Property  
%property workable !matches(ZonePH[-  
].(User[e2].PHNFC.$1|$3)|SPlatform.(PerceptionLayer[e2,x].($1|Bookn|$2)|SAgent[x,y]|$5));
```

5.5. Résultats de la vérification formelle par BigMC

Dans notre travail, nous vérifions la sûreté de notre modèle en prouvant la satisfaction des deux propriétés (voir 5.4.1.5). En effet, le programme BigMC donne à l'utilisateur la possibilité de spécifier le nombre de pas nécessaires pour vérifier la propriété sur le modèle afin d'éviter le problème d'explosion combinatoire.

L'application de la première propriété sur l'ensemble du modèle montre l'existence d'une bonne communication entre la zone de production et la zone de consommation. La figure 5.6 illustre le rapport global de résultat de la vérification obtenue après l'application de la première propriété (absence d'interblocage) sur une partie du modèle après vingt pas de simulation.

```

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

```

Figure 5.6 - Résultat de vérification de la propriété absence d’interblocage sur une partie du modèle

Sur la même propriété nous avons testé la fluidité de communication entre les différents niveaux de la zone de consommation après quatre-vingt pas. La figure 5.7 exhibe le résultat de la vérification après quatre-vingt pas.

```

bigmc> %check;
1: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAgent[x,y].nil | DelivredLayer[y,-].nil))
2: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,y].nil | DelivredLayer[y,-].nil))
4: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAgent[x,-].nil))
5: (SPlatform.(PerceptionLayer[-,x].BookX.Type1[-].nil | SAgent[x,-].nil) |
.....
40: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,-].(BookX.nil | BookX.nil) | DelivredLayer[-,-].nil))
41: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,-].(BookX.nil | beliveslike.nil) | DelivredLayer[-,-].nil))
50: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,-].(BookX.nil | beliveslike.Type1[-].nil) | DelivredLayer[-,-].nil))
51: (SPlatform.(SAgent[x,-].(BookX.Type1[-].nil | beliveslike.nil) | DelivredLayer[-,-].nil
| PerceptionLayer[-,x].nil) | ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil))
52: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,-].(BookX.Type1[-].nil | beliveslike.Type1[-].nil) | DelivredLayer[-,-].nil))
.....
70: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,-].(BookX.nil | beliveslike.nil | KnowledgeB.Puser.nil) |
DelivredLayer[-,-].Book3.nil))
79: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(SAgent[x,-].(beliveslike.nil
| BookX.Type1[-].nil) | DelivredLayer[-,-].Book3.nil | PerceptionLayer[-,x].nil))
80: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil |
SAgent[x,-].(BookX.Type1[-].nil | beliveslike.Type1[-].nil) |
DelivredLayer[-,-].Book3.nil))
[mc::step] Interrupted! Reached maximum steps: 80
[mc::report] [q: 39 / g: 119] @ 80

```

Figure 5.7 - Résultat de vérification après quatre-vingt pas

Chapitre 5. Vérification formelle du modèle BigNFC

Dans le même contexte, nous vérifions la propriété absence d'interblocage mais cette fois-ci sur la totalité du modèle. La figure 5.8 montre le résultat de la vérification de la propriété absence d'interblocage sur le modèle entier après cent pas.

```
bigmc> %check;
1: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAgent[x,y].nil | DelivredLayer[y,-].nil))
2: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil
SAgent[x,y].nil | DelivredLayer[y,-].nil))
4: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAgent[x,-].nil))
5: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(SAgent[x,-].nil
PerceptionLayer[-,x].BookX.Type1[-].nil))
6: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil
SAgent[x,-].nil))
10: (SPlatform.(SAgent[x,-].(BookX.nil | BookX.Type1[-].nil) | PerceptionLayer[-,x].nil)
ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil))
.....
99: (SPlatform.(SAgent[x,-].(BookX.nil | beliveslike.nil | KnowledgeB.Puser.LikeB.BookX.nil)
|delivredLayer[-,-].Book3.nil | PerceptionLayer[-,x].nil)
zonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | ZonePH[-].(User[-].PHNFC.nil))
100: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | ZonePH[-].(User[-].PHNFC.nil)
Platform.(SAgent[x,-].(BookX.nil | beliveslike.nil | KnowledgeB.Puser.LikeB.BookX.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 5.8 - Résultat de vérification de la propriété absence d'interblocage sur le modèle entier (cent pas)

A la fin, nous montrons le rapport global sur le résultat de vérification de la cinquième propriété sur le modèle entier. La figure 5.9 présente le résultat de cette vérification après vingt pas.

```
bigmc> %check;
1: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].nil | SAgent[x,y].nil | DelivredLayer[y,-].nil))
2: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) | SPlatform.(PerceptionLayer[-,x].nil
SAgent[x,y].nil | DelivredLayer[y,-].nil))
4: (ZonePH[-].(User[-].PHNFC.nil | SPoster.TagNFC[-].BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].() | SAgent[x,-].nil))
5: (ZonePH[-].(User[-].PHNFC.BookX.Type1[-].nil) |
SPlatform.(PerceptionLayer[-,x].(BookX.Type1[-].nil) | SAgent[x,-].nil))
.....
12: (ZonePH[-].(User[-].PHNFC.(Bookn.nil | BookX.Type1[-].nil | Bookn.nil | Bookn.nil)) |
SPlatform.(PerceptionLayer[-,x].(Bookn.nil | BookX.Type1[-].nil | Bookn.nil | Bookn.nil
Bookn.nil) | SAgent[x,-].nil))
13: (ZonePH[-].(User[-].PHNFC.(Bookn.nil | BookX.Type1[-].nil | Bookn.nil | Bookn.nil)) |
SPlatform.(PerceptionLayer[-,x].(Bookn.nil | BookX.Type1[-].nil | Bookn.nil | Bookn.nil)
SAgent[x,-].nil))
14: (ZonePH[-].(User[-].PHNFC.(Bookn.nil | BookX.Type1[-].nil | Bookn.nil | Bookn.nil
Bookn.nil)) | SPlatform.(PerceptionLayer[-,x].(Bookn.nil | BookX.Type1[-].nil | Bookn.nil
Bookn.nil | Bookn.nil | Bookn.nil) | SAgent[x,-].nil))
.....
20: (ZonePH[-].(User[-].PHNFC.(BookX.Type1[-].nil | Bookn.nil | Bookn.nil | Bookn.nil
Bookn.nil | Bookn.nil | Bookn.nil | Bookn.nil)) | SPlatform.(PerceptionLayer[-,x].(Bookn.nil
BookX.Type1[-].nil | Bookn.nil | Bookn.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 5.9 - Résultat vérification de deuxième propriété sur le modèle entier (vingt pas)

Prochainement, nous expliquons les résultats obtenus en les comparant avec les résultats des autres travaux.

5.6. Discussion

Dans cette thèse, nous proposons un modèle formel unique baptisé BigNFC dédié aux applications NFC sensibles au contexte fondé sur le célèbre formalisme BRS de Milner. En réalité, plusieurs facteurs sont entrés en jeu afin de choisir le BRS comme un formalisme d'interprétation :

Premièrement, BRS permet de montrer à la fois l'aspect géographique des entités d'un système ainsi que leurs liaisons. Deuxièmement, ce formalisme a d'autres extensions par exemple le bigraphe stochastique. Troisièmement, plusieurs travaux dans la littérature ont montré l'équivalence du bigraphe avec d'autres formalismes tels que les automates, les Rdp, ...

Dans ce travail, nous avons au départ adopté une stratégie de travail comportant trois étapes (spécification d'un système sous NFC, passage vers le modèle BigNFC, interprétation du modèle via la grammaire BigMC en vue de la vérification). Comme première étape de construction du modèle, une carte d'équivalence établissant une correspondance simple entre les composants BRS et BigNFC ; dont l'objectif est de simplifier la tâche de conception. Le modèle BigNFC est construit autour de l'idée de décomposition, dans laquelle la structure producteur / consommateur est privilégiée afin de séparer la zone de production délivrant le contexte de celle qui la traite. Plus techniquement nous avons utilisé les composants de bigraphe pour définir l'aspect statique puis nous avons donné un ensemble de règles d'interaction afin de définir l'aspect dynamique. Une seconde étape a été consacrée à l'application du modèle sur un exemple de la vie courante partant d'une spécification et allant jusqu'à l'étape de validation sur deux propriétés.

Après avoir appliqué toutes les étapes (voir chapitre 4), nous avons pu prouver la simplicité, la clarté ainsi que la cohérence dans le travail proposé. Ce parcours prouve la validité de notre modèle BigNFC sur lequel nous avons pu vérifier avec succès les deux propriétés : absence d'interblocage testée dans un premier temps partiellement puis entièrement sur le modèle ensuite nous avons proposé une deuxième propriété appliquée par la cinquième règle sur la totalité du modèle après 100 pas. En effet, nous pouvons interpréter l'absence d'interblocage par la bonne fluidité du passage des informations contextuelles entre les différents niveaux des zones. La deuxième propriété montre clairement la bonne communication entre les différents niveaux de la zone intelligente. Pour une bonne assimilation de l'importance et la puissance de notre travail, nous proposons une comparaison avec les autres

Chapitre 5. Vérification formelle du modèle BigNFC

travaux dans le même champ de travail (voir chapitre 3) ; dans un premier temps, les travaux présentés dans [78],[79] focalisent essentiellement sur la puissance du formalisme bigraphe pour modéliser les applications sensibles au contexte, où ils ont proposé un modèle général exploitant le bigraphe qui a été appliqué sur un exemple démonstratif sensible à la localisation. En revanche, aucun de ces travaux n'est arrivé à la phase de vérification. Le tableau 5.1 montre une comparaison entre BigNFC et les deux travaux connexes [78],[79] en respectant quatre critères.

	BigNFC	[78]	[79]
Phase de spécification	Oui	Oui	Oui
Phase de vérification	Utilisation un outil de modèle checker	Non	Non
Exemple basé sur	Les préférences d'utilisateur	Localisation d'utilisateur	Localisation d'utilisateur
Vérification	Oui	Non	Non

Tableau 5.1 - Comparaison entre BigNFC et [78],[79]

Comparé avec [80], il est l'unique travail de littérature présentant des résultats de vérification sur la propriété prédéfinie absence d'interblocage, testée sur vingt étapes. Notre modèle a été vérifié entièrement sur deux propriétés. Pour synthétiser, nous dressons le tableau 5.2 mettant en exergue une comparaison entre les deux travaux via quatre critères.

	BigNFC	[80]
Modèle avec BRS	Un modèle spécifique dédié aux applications sensibles au contexte sous NFC.	Modèle général pour les applications sensible au contexte.
Sensibilité au contexte sur l'exemple démonstratif	Préférences utilisateur.	Localisation d'utilisateur.
L'outil de vérification utilisé	BigMC	BigMC
Propriétés Vérifiées	Deux propriétés	Une propriété
Nombre maximal d'étapes vérifiées	Cent étapes	Vingt étapes

Tableau 5.2 - Comparaison entre BigMC et [80]

Dans cette thèse, nous avons essayé d'aller vers une perspective unique où nous avons introduit la notion du formel à ce domaine par la proposition d'un modèle formel afin de gagner

plus la confiance des utilisateurs puis généraliser cette technique à plusieurs secteurs. En effet, ce travail présente quelques faiblesses par exemple la présentation de la distance de séparation des composants NFC, ...

5.7. Conclusion

L'objectif principal de ce chapitre est de montrer la validité et la cohérence de notre modèle. De ce fait, nous avons présenté le langage BigMC pour offrir une meilleure assimilation du travail puis nous avons passé à l'étape de validation de l'approche par une démonstration claire des étapes de passage du modèle vers un programme BigMC ensuite nous avons montré les résultats de la vérification réalisée sur l'ensemble du modèle en utilisant le langage BigMC et à la fin nous avons discuté les résultats obtenus suivis d'une comparaison entre notre travail et les travaux proches de littérature. A notre connaissance, notre travail reste une nouvelle vision pour la conception et la vérification des applications sensibles au contexte sous NFC en proposant une démarche simple et claire.

Conclusion générale

Sommaire

1. Bilan général de la thèse
 2. Perspectives et travaux futurs
-

1. Bilan général de la thèse

Une application dite sensible au contexte sous NFC est toute application capable d'adapter son comportement selon des informations d'une nature spécifique nommées contexte en utilisant les objets NFC comme étant des collecteurs de ces informations.

La plupart des travaux proposés dans la littérature négligent la spécificité des applications et encore moins les applications sous NFC. En effet, nous avons eu la conscience de l'importance de ce point, d'où vient l'idée de cette thèse avec l'incorporation de la vérification formelle pour donner à ce travail plus de rigueur.

Dans ce travail, nous avons proposé un nouveau modèle formel baptisé BigNFC, basé sur le formalisme BRS pour les applications sensibles au contexte sous NFC. L'idée de sa construction est inspirée de la structure producteur/consommateur, par la séparation des deux zones : la zone délivrant le contexte brut de celle qui l'exécute. Pour réaliser notre idée nous avons utilisé le formalisme BRS, où l'aspect statique qui est un ensemble de composants et d'éléments, est modélisé sous forme de bigraphe, et l'aspect dynamique (comportements) sous forme de règles d'interaction. A cet égard, nous présentons une carte d'équivalence entre les éléments BRS et le système sensible au contexte sous NFC. La puissance de cette table de correspondance réside dans l'aptitude du concepteur de la programmer et de pouvoir générer automatiquement du le modèle cible. Nous avons appliqué après notre modèle sur un exemple. Afin de vérifier les propriétés sur le modèle, nous avons recouru à l'outil BigMC, qui donne une preuve automatique de correction, par l'application des propriétés de sûreté sur un exemple et les résultats obtenus sont très flatteurs après 100 étapes.

2. Perspectives et travaux futurs

N'importe quel travail scientifique ouvre de nouvelles perspectives pour le faire évoluer dans le temps. Un travail parfait n'existe pas. Pour cette raison, dans un futur proche nous envisageons d'atteindre les objectifs suivants :

- Faire évoluer le modèle BigNFC : Pour la simple raison est que dans la vie réelle, l'environnement est dynamique, c'est pourquoi un modèle probabiliste est plus approprié et par conséquent nous tentons de faire l'extension du modèle BigNFC à un modèle probabiliste ;

Conclusion générale

- Ajouter de règles plus complexes, avec la création de situations de synchronisation et de vérification d'un modèle alternatif par un outil probabiliste ;
- Explorer les pistes des spécificités des modes d'interaction notamment celles de la carte d'émulation ;
- Programmer la table d'équivalence pour rendre le passage entre un système sensible au contexte sous NFC et le modèle BigNFC se fait d'une façon plus naturelle et de manière automatique.

Bibliographie

- [1] R. MILNER, «Axioms for bigraphical structures,» *Mathematical Structures in Computer Science*, vol. 15, n° 16, pp. 1005--1032, 2005.
- [2] V. LOMNE, . T. ROCHE et al, «A Side Journey to Titan,» chez *30th USENIX Security Symposium*, 2021.
- [3] Martin, «Serrure connectée de NETATMO», [En ligne]. Disponible : <https://www.robots-et-compagnie.com/serrure-connectee-netatmo/>. [consulté le 18 mai 2020].
- [4] C. Hall, «Qu'est-ce qu'Apple CarKey et comment va-t-il révolutionner la conduite automobile ?», [En ligne]. Disponible : <https://www.pocket-lint.com/fr-fr/vehicules/actualites/apple/152710-quest-ce-qu-apple-carkey-et-comment-va-t-il-revolutionner-la-conduite>. [consulté le 6 juin 2022]
- [5] A. NABET et R. BOUDOUR, «BigNFC: Novel Formal model for NFC based context-aware applications,» *Informatica*, vol. 45, n°15, pp. 705-712, 2021.
- [6] B. SHILIT, N. ADAMS et R. WANT, «Context-aware computing applications,» chez *First workshop on mobile computing systems and applications*, 1994.
- [7] J. PASCOSE , K. THOMSON et H. RODRIGUES, «Context-awareness in the wild: an investigation into the existing uses of context in everyday life,» chez *The Move to Meaningful Internet Systems*, 2007.
- [8] A. K. DEY, D. SALBER, G. ABOWD et al, «Combining context-awareness with wearable computing,» chez *Digest of Papers. Third International Symposium on Wearable Computers*, 1999.
- [9] P. J. Brown, «The stick-e document: a framework for creating context-aware applications,» *Electronic Publishing-Chichester*, vol. 8, pp. 259--272, 1995.
- [10] C. RAYAN et P. ROSSI, «Software, performance and resource utilisation metrics for context-aware mobile applications,» chez *11th IEEE International Software Metrics Symposium*, 2005.
- [11] M. PLEC, R. ANTHONY, P. WARD et J. HAWTHORNE, «Towards context-Awareness and self-Configuration of embedded systems: Resource usage and real-time characteristics,» chez *Proceedings of 7th Workshop on Advanced Control and Diagnostics*, 2009.
- [12] J. PASCOSE , N. RAYAN et D. MORSE, «Using while moving: HCI issues in fieldwork environments,» *ACM Transactions on Computer-Human Interaction*, vol. 7, n°13, pp. 417--437, 2000.
- [13] A. K. DEY, «Understanding and using context,» *Personal and ubiquitous computing*, vol. 5, n° 11, pp. 4--7, 2001.
- [14] T. CHAARI, F. LAFOREST et A. CELENTANO, «Design of context-aware applications based on web services,» 2004.

- [15] M. MIRAOUI, C. TADJ et C. BENAMAR, «Context modeling and context-aware service adaptation for pervasive computing systems,» *International Journal of Computer and Information Science and Engineering*, vol. 2, n° 13, pp. 148--157, 2008.
- [16] N. S. RAYAN, J. PASCOE et D. R. MORSE, «Enhanced reality fieldwork: the context-aware archaeological assistant,» chez *Computer applications in archaeology*, 1998.
- [17] T. RODDEN, K. CHEVERST, K. DAVIES et A. DIX, «Exploiting context in HCI design for mobile systems,» chez *Workshop on human computer interaction with mobile devices*, 1998.
- [18] E. NOT et D. PETRELLI, «Blending customisation, context-awareness and adaptivity for personalised tangible interaction in cultural heritage,» *International Journal of Human-Computer Studies*, vol. 114, n° 12018, pp. 3--19, 2018.
- [19] G. CHEN et . D. KOTZ, «A Survey of Context-Aware Mobile Computing Research,» 2000.
- [20] J. GWIZDKA, «What's in the context,» chez *Computer Human Interaction*, 2000.
- [21] T. HOFER, W. SCHWINGER, M. PICHLER et al, «Context-awareness on mobile devices-the hydrogen approach,» chez *36th annual Hawaii international conference on system sciences*, Hawaii, 2003.
- [22] P. M. KIRSCH et C. SOUVEYET, «Is Group-Awareness Context-Awareness?,» chez *International Conference on Collaboration and Technology*, 2018.
- [23] M. A. REZZAQUE, S. DOBSON et al, «Categorization and modelling of quality in context information,» *American Scientific Publishers*, vol. 2006, n° 18, p. 38, 2006.
- [24] T. A. CHERFIA, «Un Framework Basé Bigraphes pour la Conception et l'Analyse des Systèmes Sensibles au Contexte», Constantine, Université Constantine 2- Abdelhamid Mehri, 2016.
- [25] C.-H. LIN, P.-H. HO et H.-C. LIN, «Framework for NFC-based intelligent agents: a context-awareness enabler for social internet of things,» *International Journal of Distributed Sensor Networks*, vol. 10, n° 12, pp. 978--951, 2014.
- [26] K. HENRICKSEN et J. INDULSKA, «Developing context-aware pervasive computing applications: Models and approach,» *Pervasive and mobile computing*, vol. 2, n° 11, pp. 37--64, 2006.
- [27] Y. GHAYAM, *La sensibilite au contexte dans un environnement mobile*, Rabat, Informatique, Chapitre 1 pp. 20--25, 2011.
- [28] . W. . N. SCHILIT, *A system architecture for context-aware mobile computing*, Columbia: Columbia University, 1995.
- [29] X. LI, M. ECKERT et al, «Context aware middleware architectures: Survey and challenges,» *Sensors*, vol. 15, n° 18, pp. 20570--20607, 2015.
- [30] A. DEY et J. HÄKKILÄ, «Context-awareness and mobile devices,» chez *Handbook of research on user interface design and evaluation for mobile technology*, 2008.

- [31] M. MIRAOU, C. TADJ et C. B. AMAR, «Architectural survey of context-aware systems in pervasive computing environment,» *Ubiquitous Computing and Communication Journal*, vol. 3, n° 13, pp. 1--9, 2008.
- [32] E. ROHN, «Predicting context aware computing performance,» *Ubiquity*, vol. 2003, n° 1February, pp. 1--17, 2003.
- [33] B. SCHILIT, N. ADAMS et R. WANT, «Context-aware computing applications,» chez *first workshop on mobile computing systems and applications*, 1994.
- [34] J. BAUER, R.-D. KUTSCHE et al, «Identification and modeling of contexts for different information scenarios in air traffic,» *Technische Universität Berlin, Diplomarbeit*, vol. 62, 2003.
- [35] K. HENRICKSEN et J. INDULSKA, «Developing context-aware pervasive computing applications: Models and approach,» *Pervasive and mobile computing*, vol. 2, n° 11, pp. 37--64, 2006.
- [36] K. HENRICKSEN, J. INDULSKA et al, «Modelling context information with ORM,» chez *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems*, 2005.
- [37] . K. HENRICKSEN et J. INDULSKA, «Developing context-aware pervasive computing applications: Models and approach,» *Pervasive and mobile computing*, vol. 2, n° 11, pp. 37--64, 2006.
- [38] R. ROBINSON, K. HENRICKSEN et al, «XCML: A runtime representation for the Context Modelling Language,» chez *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2007.
- [39] R. DE VIRGILIO et R. TORLONE, «Management of heterogeneous profiles in context-aware adaptive information system», chez *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems*, 2005.
- [40] Q. Z. SHENG et B. BENATALLAH, «ContextUML: a UML-based modeling language for model-driven development of context-aware web services,» chez *International Conference on Mobile Business*, 2005.
- [41] N. GODARD et A. VAGNER , «Environnement sensible au contexte pour l'exécution de widgets sur mobile,» *IHM'10*, p. 23, 2010.
- [42] R. WANT, A. HOPPER, V. FALCAO et al, «The active badge location systems,» *ACM Transactions on Information Systems*, vol. 4, pp. 42--47, 1997.
- [43] M. O. BALITANAS et T. KIM, «Architecture for Automatic Management of ParcTab Ubiquitous Computing,» *Architecture*, vol. 15, 2010.
- [44] M. VAN SETTEN, S. PAKRAEV et al, «Context-aware recommendations in the mobile tourist application COMPASS,» chez *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2004.
- [45] nfcforum, «The technology of the futur: Near Field Communication,» [En ligne]. Disponible : <https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/NFCChips/nfcforum.html>. [consulté le 5 mars 2022]

- [46] A. Leslie, «Disruptive views BillingViews», [En ligne]. Disponible : <https://disruptiveviews.com/nfc-survey-where-does-nfc-belong/>. [consulté le : 7 Mai 2015].
- [47] 01 net, «Cartes bancaires NFC : une faille permet de voler des millions d'euros», [En ligne]. Disponible : <https://www.01net.com/actualites/cartes-bancaires-nfc-une-faille-permet-de-voler-des-millions-d-euros-630996.html>. [consulté le : Novembre 2015].
- [48] V. COSKUN, K. OK et B. OZDENIZCI, Near field communication (NFC): From theory to practice, John Wiley & Sons Ltd Edition, 2012, p. 390.
- [49] S. A. AHSON et M. ILYAS, RFID handbook: applications, technology, security, and privacy, CRC press, 2017, p. 675.
- [50] G. MADLMAYR, J. LANGER, C. KANTNER et al, «NFC Devices: Security & Privacy,» chez *Third International Conference on Availability, Reliability and Security*, 2008.
- [51] CarrefourNet, «La RFID», [En ligne]. Disponible : <http://www.carrefour.net/fr/articles.html?t=59>. [consulté le 7 avril 2015].
- [52] G. MALDLMAYR, J. LANGER et al, «NFC devices: Security and privacy,» chez *Third International Conference on Availability, Reliability and Security*, 2008.
- [53] M. M. SINGH, K. ADZMAN et R. HASSAN, «Near Field Communication (NFC) technology security vulnerabilities and countermeasures,» *International Journal of Engineering & Technology*, vol. 7, n° 14.31, pp. 298--305, 2018.
- [54] E. HASELSTEINER et K. BREITFUß, «Security in near field communication (NFC),» chez *Workshop on RFID security*, 2006.
- [55] S. AKTER, S. CHELLAPPAN, T. CHAKRABORTY et al, «Man-in-the-middle attack on contactless payment over NFC communications: design, implementation, experiments and detection,» *IEEE Transactions on Dependable and Secure Computing*, vol. 18, n° 16, pp. 3012--3023, 2020.
- [56] O. ITALIS et S. PIERRE, Les systèmes de paiement mobile à l'ère de la Covid-19: sécurité, vie privée et confiance numérique, London: ISTE Ltd., 2021.
- [57] . P.-H. THEVENON, *Sécurisation de la couche physique des communications sans contact de type RFID et NFC*, Grenoble, Université de Grenoble: Université de Grenoble, 2011.
- [58] U. BIADER CEIPIDOR, C. M. MEDAGLIA, . A. MORONI et al, «NFC: Integration between RFID and Mobile, state of the art and future developments,» *RFIDays*, 2008.
- [59] V. DUTOT, «Factors influencing near field communication (NFC) adoption: An extended TAM approaches,» *The Journal of High Technology Management Research*, vol. 26, n° 11, pp. 45--57, 2015.
- [60] V. COSKUN, B. OZDENIZCI et K. OK, «The survey on near field communication,» *Sensors*, vol. 15, n° 16, pp. 13348--13405, 2015.
- [61] S. C. ALLIANCE, «Mobile/NFC Security Fundamentals: Secure Elements,» 2013.

- [62] G. MADLMAYR, J. LANGER et J. SCHARINGER, «Managing an NFC Ecosystem,» chez *7th International Conference on Mobile Business*, 2008.
- [63] kimovil, «Les meilleurs smartphones avec NFC», [En ligne]. Disponible: <https://www.kimovil.com/fr/meilleurs-mobiles-avec-nfc>. [consulté janvier 2020]
- [64] W. SUPARTA, «Application of Near Field Communication Technology for Mobile Airline,» *Journal of Computer Science*, pp. 1235-1243, 2012.
- [65] D. NELSON, M. QIAO et A. CARPENTER, «Security of the near field communication protocol: an overview,» chez *CCSC: Rocky Mountain Conference*, 2013.
- [66] V. COŞKUN, K. OK et B. ÖZDENIZCI KÖSE, «Chapter 4 : NFC Operating Modes,» chez *Near Field Communication: From Theory to Practice.*, John Wiley and Sons, 2011.
- [67] R. MEINDL, «NFCIP-1 security standard protects near field communication,» chez *4th ETSI Security Workshop*, 2009.
- [68] W. ALI, W. ELKILANI et M. HADHOUD, «Simulation of NFCIP-1 Protocol over NS-2,» chez *The 7th International Conference on Informatics and Systems (INFOS)*, 2010.
- [69] . B. OZDENIZCI, . M. ALSADI, K. OK et al, «Classification of NFC applications in diverse service domains,» *International Journal of Computer and Communication Engineering*, vol. 2, n° 15, p. 614, 2013.
- [70] S. PAMPATTIWAR, «Literature survey on NFC, applications and controller,» *International Journal of Scientific and Engineering Research*, vol. 3, n° 12, pp. 1--4, 2012.
- [71] F. RESATSCH, «Designing an Ubiquitous Computing Application Development and Evaluation Process Model,» chez *Ubiquitous Computing: Developing and Evaluating Near Field Communication Applications*, Springer Science and Business Media, pp. 87--98, 2010.
- [72] N. FRANKE, E. VON HIPPEL et M. SCHREIER, «Finding commercially attractive user innovations: A test of lead-user theory,» *Journal of product innovation management*, vol. 23, n° 14, pp. 301--315, 2006.
- [73] M. MURPHY, *L'art du développement Android*, Paris: Pearson Education France, 2010.
- [74] R. MILNER, «Pure bigraphs: Structure and dynamics,» *Information and computation*, vol. 204, n° 11, pp. 60--122, 2006.
- [75] O. H. JENSEN et R. MILNER, «Bigraphs and transitions,» chez *Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages*, 2003.
- [76] R. MILNER, «Axioms for bigraphical structure,» *Mathematical Structures in Computer Science*, pp. 1005 --1032, Décembre 2005.
- [77] . R. MILNER, «Pure bigraphs: Structure and dynamics,» *Information and computation*, vol. 204, n° 11, pp. 60--122, 2006.
- [78] L. BIRKEDAL , S. DEBOIS, E. ELSBORG et al, «Bigraphical models of context-aware systems,» chez *International Conference on Foundations of Software Science and Computation Structures*, 2006.

- [79] J.-S. WANG, D. XU et Z. LEI, «Formalizing the structure and behaviour of context-aware systems in bigraphs,» *International Symposium on Software and Network Engineering*, vol. 2011, n° 11, pp. 89--94, 2011.
- [80] T. A. CHERFIA, F. BELALA et K. BARKAOUI, «A bigraph-based framework for specification and analysis of context-aware systems,» *International Journal of Critical Computer-Based Systems*, vol. 6, n° 14, pp. 322--342, 2016.
- [81] A. VAKILI et N. A. DAY, «Temporal logic model checking in Alloy,» chez *International Conference on Abstract State Machines, Alloy, B, VDM, and Z*, 2012.
- [82] E. Højsgaard et al, «Bigraphical languages and their simulation,» Citeseer, 2011.
- [83] G. BACCI, D. GROHMANN et M. MICULAN, «DBtk: a toolkit for directed bigraphs,» chez *International Conference on Algebra and Coalgebra in Computer Science*, 2009.
- [84] Q.-Q. SHI, «An Improvement for BigMC,» chez *International Conference on Machinery, Materials and Information Technology Applications*, 2017.
- [85] S. KHOSHMANESH, «Safety and Security specifications in an AR Context-Aware Systems: A model based on Bigraphical Reactive System (BRS),» *OSF Preprints*, pp. 1--6, 2018.
- [86] A. GASSARA, I. B. RODRIGUEZ, M. JMAIEL et al, «Executing bigraphical reactive systems,» *Discrete Applied Mathematics*, vol. 253, pp. 73--92, 2019.
- [87] M. SEVEGNANI et M. CALDER, «BigraphER: rewriting and analysis engine for bigraphs,» chez *International Conference on Computer Aided Verification*, 2016.
- [88] G. PERRONE, S. DEBOIS et T. T. HILDEBRANDT, «A model checker for bigraphs,» chez *Proceedings of the 27th annual ACM symposium on applied computing*, 2012.