Trends and challenges in Formal Specification and Verification of Services Composition in Ambient Assisted Living Applications

Mohamed Hilia\textsuperscript{a}, Abdelghani Chibani\textsuperscript{a}, Karim Djouani\textsuperscript{b,a}

\textsuperscript{a}Signals Images & Intelligent Systems Laboratory, Paris-Est Creteil University, Vitry Sur Seine, France
\textsuperscript{b}F\textsuperscript{S}ATIE/TUT, Pretoria, South Africa

Abstract

Emerging Ambient Assisted Living (AAL) applications, as a part of AmI applications, deal essentially with healthcare related applications such as assistance to the elderly and handicapped persons, emergency services. Several approaches and techniques have been proposed, providing formal languages modeled with ontologies (e.g. OWL-S, WSMO) that describe in semantic way the environment. In this paper, relevant challenges of the current AAL application development, with a focus on the formal specification and verification are discussed. A formal system which enable to specify a semantic model represented by an upper ontology is presented. The innovative aspect of the proposed model concerns the use of a constructive description logic.

Keywords: Ambient Assisted Living, Ontology Specification and Modeling, Theorem Proving, Isabelle/HOL

1. Introduction

Ambient Intelligence (AmI) emerged as an initiative for building intelligent environment such as smart homes, smart hospital, etc. The AmI systems aim at providing elderly with specific services enabling a better, healthier and safer life in the preferred living environment. These services often consist of assisting users in their daily life activities by incorporating companion robots, deploying sensors, and actuators in the environment. AmI applications are mainly service-based applications. Basically, these services are provided with the growing number of the deployed devices in the environment. In the most cases, these services have different providers. Therefore, heterogeneity and distribution issues should be handled. Besides, building such applications is usually based on the integration of the provided services within the infrastructure to create high level services ensuring, on the one hand, the safety and the security of the users, and providing, on the other hand, new services responding to the user’s needs. Currently, Emerging Ambient Assisted Living (AAL) applications as a part of AmI area deal essentially with healthcare related applications such as assistance to the elderly and handicapped persons, emergency medical services and assistive living services in general. The complexity and the criticality of such applications leads to a real need for methodologies and tools that can improve the reliability of the final systems [1]. The AAL applications can be considered
as safety critical systems, [2], where the user safety is an important requirement [3]. In general, provided services in AmI environment are based on the integration of several critical services, such as rehabilitation services, as for instance the approach proposed in [4] that provides a multi-robot services integration. Indeed, a list of critical new challenges and trends raised from the point of view of heterogeneity handling, semantic and behavioral interoperability, formal verification of the composite services, behavioral correctness, dependability, safety and security of the users as well as dealing with the missing information, sharing the context with formal semantics between the heterogeneous component[5][6], reusable artifacts for implementing context-aware behavior, and closed-world assumptions reasoning [7]. Current research is focusing on how to semantically describe services (in a formal and expressive way), how to (automatically) compose them, how to discover them and how to ensure their correctness [8, 9].

Several approaches and techniques have been proposed to deal with most of the presented challenges by providing formal languages based ontologies (e.g. OWL-S, WSMO) leading to semantic descriptions of the environment, the list of the delivered services, their functionalities and their capabilities [10], the services composition, and the provisioning procedures [11]. The use of the ontologies has numerous advantages such as giving a formal semantic for the exchanged data, potentially providing a well-founded mechanism for the representation and sharing of such structured information [12], and also enabling reasoning to select the right services in the right context. As mentioned in [13], one of the most important requirement concerns the formal verification of these critical applications. Moreover, we have to ensure the correct design and formal analysis of the AAL applications. Unfortunately, the most existing modeling languages for building ambient intelligence systems such as in [4], present the lack of supporting any formal framework the system specification and verification of its correctness.

This paper is organized as follows. In section (2) the related work about proposed techniques and methods in the context of the AAL applications is discussed. Relevant challenges of the current AAL application development, with a focus on the formal specification and verification are presented in (Section 3). Section (5) presents the proposition of a framework for formal specification and semantic modeling represented by an upper ontology. The proposed model is based on a constructive description logic presented in section (4). The formal system is composed by formulas asserting the conceptual model and the state of the environment, and a set of inference rules expressed in natural deduction system. This inference system allows the logic consequences proof in this model. The formal system specification and the soundness of its natural deduction systems are shown in the interactive theorem prover Isabelle/HOL.

2. Related work

Several formal methods have been proposed to guarantee the correctness of the service composition. Most of them are based on the state transition semantics such as Petri-nets. In addition, very few contributions are based on the theorem proving. We present the proposed approaches according to their basics techniques, namely, Model checking and Theorem Proving. Both methods aim to verify the properties of a system specification. They were widely used in the literature in different disciplines to specify models and their required properties.

Model checking based approaches. Model checking based approaches are useful for the verification of some properties given by some formal model, generally considered as a state-transition system. The model checker system, depending on the approach used, checks all the possible execution cases. Despite the advantage of these methods that are fully automatic, they are not suitable for verifying large systems. The main limitation is that the transition system must be finite, it means that, the system should handle a finite domain variables. For this purpose, these methods are limited by the size of the state space, a large space generates combinatorial explosion phenomenon of the number of system status. In addition, systems specification-based state transition can lead to a loss of semantics in the encoding system.

Theorem Proving based approaches. The theorem proving techniques are not limited by the size of the state space. Large systems that cannot be handled using Model checking based approaches can still be verified by a theorem prover In [14] a method for automatic composition of Semantic Web services using Intuitionist
Linear Logic (ILL) theorem proving is presented. This approach represents Semantic Web services using DAML-S as an external presentation, while, internally, the services are presented by extra-logical axioms and proofs in LL. This approach uses a process algebra, an extension of the $\pi$-calculus, to formalize the service composition. The approach specifies services as DAML-S descriptions. These descriptions are transformed into ILL axioms. The composite service is represented as a theorem to be proven in LL theorem prover. Thus, the process model for a composite service can be generated directly from the complete proof, if the proof exist. In [15], the authors propose a similar approach, except for the web service initial specification. The authors use a Classical Linear Logic instead of the Intuitionist Linear Logic. They consider that the previous approach contains a number of potential inconsistencies. For instance, the process calculus being used is an extension of the $\pi$-calculus. However, no guarantees are given that the two calculi are equivalent or that the Bellin and Scott proofs are valid for the extended process calculus. Additionally, they use Intuitionist Linear Logic in a two-sided sequent calculus, which is also not guaranteed to be equivalent to the one-sided CLL approach of Bellin and Scott.

Both approaches are based on the proposed connection of the LL proof and $\pi$-calculus. The translation between rules of a fragment of LL and $\pi$-calculus is initiated by Abralsky. Abralsky gives the soundness and completeness for the processes calculus translation of the CLL based on one-side sequents. Rao make some modification for Abramsky’s translation to fit their presentation without guaranteed the soundness and the completeness of the new system. Papapanagiotou considers that the latter modification does not guarantee the soundness and the completeness, because of the extension of the target process calculus.

Bozzato proposed a formalization in Basic Constructive Description Logic $BCDL_0$ [16], and the proof of the correctness of the composition with respect to the requested service. This approach has motivated the work presented in this paper. No translation is needed from the initial specification into another formalism such as in [14]. The proposed methodology is based on a constructive description logic that support a computational interpretation of proofs. Another important advantage raised with the capability to specify both specification and the implementation by giving a structured witness for the proofs, and also their specification in a theorem prover to prove the correctness property.

3. Challenges and trends for Ambient Assisted Living Applications

A survey on main challenges and research problems in the context of the ambient intelligence systems is presented.

3.1. Semantic and Behavioral interoperability

Regarding ambient intelligence environments and intelligent applications processes development, interoperability is essential for successful communication and mutual understanding of the wide range actuators and sensors available in the environment in order to correctly interpret the information and perform the request actions. There are two types of interoperability, semantic interoperability and behavioral interoperability. The semantic interoperability concerns the structure and the representation of the exchanged data between the heterogeneous devices in the environment. The behavioral interoperability concerns particularly the preconditions and postconditions (i.e effect) of the various actions and services performed/achieved in the environment. These actions are mostly delivered by heterogeneous sensors and actuators in the environment.

3.2. Context management

The performed actions and services depend on the specific situation of the environment, the users situation and intention, and the state of the system. Dealing with these contextual information, and sharing the context requires more formal semantics investigations to enable context-aware applications and an efficient management of the context between the heterogeneous components [5][6].
3.3. Service composition description languages

All are service-based applications. Describing these services in a formal machine understandable way, enables the automation of several tasks such as the service discovery, the service composition, and the selection of the appropriate services in a given context and by considering QoS parameters. Recently, several approaches were proposed in the state of the art to semantically describe and semantic web ontologies such as WSDL-S, DAML-S, WSMO, SWSL, SAWSDL [17, 18]. The most adopted language for building semantic description of web services is OWL-S [19], the successor of DAML-S [20]. For instance, OWL-S is used in [4] to describe the ambient intelligence systems semantic web services description and composition.

3.4. Formal verification

In an ambient environment, the main objective is to ensure service delivery and users safety and in particular, in healthcare domain, where the safety of users is the ultimate priority. This motivates the need for formal validation techniques to ensure not only building good applications, but also the correctness. The proposed languages to describe services present a lack of giving the formal semantic and lack of the formal analysis dealing with form do not support formal approach for the verification of system correctness. As a consequence, effort should be made to integrate formal model into these languages or proposing new ones.

3.5. Reusable artifact

Reuse of the existing artifact is required to accelerate the development process. This aspect has been discussed in [13], the authors show that it is a fundamental propriety in a development methodology of an intelligent applications especially in the context of AAL applications. The proposed methodologies should found ways in order to increase the reuse not only for the development processes, the implemented artifact, but also for the design and how the application behaves when running.

4. Towards Constructive Description Logics

In this section, we propose a formal system based on the constructive description logic $\mathcal{BCDL}_0$ [21]. This formal system is used to specify in semantic and formal way the AAL ontology. The benefits of this formal system is twofold. The first is that semantic representation enables the reasoning on the ontologies entities and formally derives new knowledge. The second is that this constructive logics, provides an interpretation to realize proofs in such system. The syntax and semantics of the Basic Constructive Description Logic $\mathcal{BCDL}_0$ presented in [16], is discussed bellow.

4.1. Basic Constructive Description Logic, $\mathcal{BCDL}_0$

Description logics (DL) are a family of knowledge representation languages [22]. The main objective of LDs is to formally represent knowledge and to reason effectively to minimize response times. LDs are mainly based on three basic entities: concepts, roles and individuals. A concept represents a set of objects, a role represents a relationship between two objects, and individuals are the objects. These entities are organized in the following sets. NC the set of the concept names, NR is the set of the role names, and NI is the set of the individual names, and VAR the set of the individual variable names. These entities are represented by the respect of the grammar depicted in the table (Table 1).

| C, D ::= A | ¬ C | C D | C D | ∃ R. C | ∀ R. C |
| K ::= ⊥ | t : C | A ⊑ C | (s, t) : R |

Where A is an atomic concept, s,t ∈ NI ∪ VAR. C, D ∈ NC and R ∈ NR. A concept is built from atomic concepts, denoted by A. These concepts represent the entities of the application domain. The formulation shown in the table (Table 1) defines also the types of formulas K generated by elements of the logic (i.e. concepts and constructors).
4.2. Constructive interpretation

The constructive semantics are given by mean of a structured mathematical object, associated to the presented $K$ formulas. This structure represent the witness justifying the truth of the associated formula in a classical model. The underlying object is named information term, and it is defined according to a subset $N$ of $N$, noted $IT_N(K)$. Let consider $N \subseteq NI$, $\mathbb{L}_N$ is the language represented by the generated formulas (Table 2), where the individual variable names are in $N$. The constructive interpretation of $BCD\mathbb{L}_0$ is based on information terms. Formally, if $N \subseteq NI$ and $K$ is a formula without individual variables names (closed formulas), The list of information terms $IT_N(K)$ are defined by induction on the structure of $K$ as follows:

| $TT \ N(K)$ | $\{ it \} \iff K \ is \ a \ closed \ formula$ |
| $TT_N(c : C_1 \cap C_2)$ | $\{ (\alpha, \beta) \ | \ \alpha \in TT_N(c : C_1) \ and \ \beta \in TT_N(c : C_2) \}$ |
| $TT_N(c : C_1 \cup C_2)$ | $\{ (k, \alpha) \ | \ k \in 1,2 \ and \ \alpha \in TT_N(c : C_1) \}$ |
| $TT_N(c : \exists R.C)$ | $\{ (d, \alpha) \ | \ d \in N \ and \ \alpha \in TT_N(d : C) \}$ |
| $TT_N(c : \forall R.C)$ | $\{ \phi : N \rightarrow \bigcup_{d \in N} TT_N(d : C) \ | \ \phi(d) \in TT_N(d : C) \}$ |
| $TT_N(A \subseteq C)$ | $\{ \phi : N \rightarrow \bigcup_{d \in N} TT_N(d : C) \ | \ \phi(d) \in TT_N(d : C) \}$ |

This formalization associates to a closed formula a type noted by tt. This object can be considered as a reference to a Java class object or an entry stored in database in a database management system.

5. Formal Semantic Model Specification and Theorem Proving

In this section, we present a formal system to semantically describe the various knowledge and services in an ambient environment. The proposed model $S = (L, R)$ consists of the conceptual model $L$, and a set of inference rules $R$. $L$ represent the application domain knowledge represented by a list of the previously described concepts and formulas. $R$ is the set of the inference rules based on the natural deduction system and expressed in [21] by means of explicit context representation. A context is denoted by $\Gamma$. The context contains the assumptions and the proved formulas. Proof of a formula $K$ from a set of assumptions $\Gamma$ is denoted by $\Gamma \vdash K$. The proof is a sequence of application of natural deductions rules shown in [21].

Illustrative example : . Let consider the case presented in (Fig.1). It represents the knowledge base of an ambient intelligence environment. The formalization of this knowledge is formalized in the table (Table 3).

<table>
<thead>
<tr>
<th>Table 3. Part of the AAL Ontology specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TBox$</td>
</tr>
<tr>
<td>(Ax1) : NotificationAction $\subseteq \exists isPerformedBy.Robot$</td>
</tr>
<tr>
<td>(Ax2) : Robot $\subseteq \exists isSituatedOn.Location$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$ABox$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ax1) : kompai : Robot</td>
</tr>
<tr>
<td>(ax2) : alert1 : NotificationAction</td>
</tr>
<tr>
<td>(ax3) : alert2 : NotificationAction</td>
</tr>
<tr>
<td>(ax4) : kitchen : Location</td>
</tr>
<tr>
<td>(ax5) : room : Location</td>
</tr>
<tr>
<td>(ax6) : (alert2, kompai) : isPerformedBy</td>
</tr>
<tr>
<td>(ax7) : (kompai, room) : isSituatedOn</td>
</tr>
</tbody>
</table>

Let consider the formula which associates a robot to a location in the smart home, such as $K = Robot \cap \exists isLocatedOn.Location$. Let assume that $NI$ is the set of all individuals in the knowledge base. The computation of the IT Interpretation of $K$ is the following. Let $N$ be the names of individuals in the model defined in table (Table 3). An example of an element $IT_N(Ax1)$ is a function $\phi$ which associates to each element $c$ of $N$ to an element $\gamma$ such as :

$$IT_N(Ax1) = \{ \phi : N \rightarrow \bigcup_{d \in N} TT_N(d : C) \ | \ \phi(d) \in TT_N(d : C) \}$$
\[ \gamma \in \Pi_{\lambda}(c : \text{Robot} \sqcap \exists \text{isLocatedOn}.\text{Location}) \]

\[ \Pi_{\lambda}(K) = \Pi_{\lambda}(c : \text{Robot} \sqcap \exists \text{isLocatedOn}.\text{Location}) \]

\[ = (\Pi_{\lambda}(c : (\text{Robot})), \Pi_{\lambda}(c : \exists \text{isLocatedOn}.\text{Location})) \]

\[ = (tt, (d, tt)) \]

The state of the system is represented by ABox. The information terms \((tt(\text{room}, tt))\) express the fact that \(c\) is the robot and its location in the \textit{room}. These formulas, also called the axioms establish that the robot represented by an atomic concept \textit{Robot} is associated with a position represented by the atomic concept \textit{Location} via the role \textit{isSituatedOn}. The robot can also perform actions such as notification, in our semantic model robot provides various services, such as notification, moving, localization of an object or a person, etc. The notification action is represented by the concept \textit{NotificationAction}, this feature is associated with the robot by means of the role \textit{isPerformedBy}. The \textit{Abox} gives the environment objects with the individual names. Axioms express that in the environment there are two notification actions triggered by the robot \textit{kompai} \(^1\) in two different places, namely, \textit{room} and \textit{kitchen}. Each notification corresponds to a location of the robot \textit{kompai}. For example, the formula in this knowledge base robot \textit{kompai} is not in the kitchen, for this purpose, the formula \((\text{kompai}, \text{kitchen}) : \text{isSituatedOn}\) is not valid in this interpretation, and writing \(\Delta_{\lambda} \not\models (\text{kompai}, \text{kitchen}) : \text{isSituatedOn}\) because no formula justifies this. Let consider the following formula that states that a robot can perform a notification where it is in specific location. We can construct the proof of this formula expressed as follows:

\[ \pi :: \mathcal{T} \vdash (\text{NotificationAction} \sqsubseteq \exists \text{isPerformedBy}.(\text{Robot} \sqcap \exists \text{isLocatedOn}.\text{Location})) \]

6. Formal Specification within Isabelle/HOL

Isabelle/HOL represent a formal proven environment to develop formal systems. Isabelle/HOL provides a rich collection of library theories like sets, seq, relations, and various arithmetic theories. Several automated proofs procedures like simp, auto, and the arithmetic types such as integers have been done. Among the advantages of the implementation of Isabelle/HOL, we may notably mention the fact that the prover treats simultaneously process specification and semantic aspects. In addition, the theorem proving tool provides a meta-proven logic for defining specification languages and techniques to take advantage of evidence

\(^1\)Kompai is a robot manufactured by RobotSoft
and proof simplification. The constructive semantics is based on the notion of type of information introduced in [23], this notion is shown in [16] as information terms. The semantics of this logic is isomorphic to the type theory λ-calculus proposed by Alain Church. The formulas of the logic are considered types. More precisely, the type of a formula \( K \) can be seen as a characteristic of information needed to justify its truth value in the classical model. This semantic information terms gives computational interpretation of natural deduction proof system \( \mathcal{BCDL}_0 \). In the following we present the syntax implementation of the proposed formal system in Isabelle/HOL theorem prover. The basic entities are defined using the \texttt{datatype} constructors. The formal system is presented as a theory that uses the main theory \texttt{Main.thy} which contains the primitive data types integers, boolean and all the basic theories in arithmetic.

theory \texttt{BCDL} imports \texttt{Main} begin

datatype 'nr role = AtomR 'nr

datatype 'ni Individu = AtomN 'ni

datatype ('nr,'nc) Concept = AtomC 'nc

| NotC "('nr,'nc) Concept" ("\not\_")
| OrC "('nr,'nc) Concept" "('nr,'nc) Concept" ("\_\langle\cup\rangle\_")
| AndC "('nr,'nc) Concept" "('nr,'nc) Concept" ("\_\langle\cap\rangle\_")
| SomC "('nr) role" "('nr,'nc) Concept"("\exists\langle_\_\rangle\_")
| AllC "('nr) role" "('nr,'nc) Concept"

datatype ('nr,'nc,'ni) Kformulas = Bottom

|RoleF "'ni * 'ni " 'nr role" ("\_\_\langle_\_\rangle\_")

|ConceptF "'ni" "('nr,'nc) Concept" ("\_\_\langle_\_\rangle\_")
|AConceptF "'nc" "('nr, 'nc) Concept" ("\_\langle\subseteq\rangle\_")

fun is_atomic_formula ::"('nr,'nc,'ni) Kformulas \Rightarrow bool"
where
" is_atomic_formula (Bottom) = True"
| " is_atomic_formula (ConceptF (n) (AtomC a))= True"
| " is_atomic_formula (RoleF ( n, c) (AtomR r)) = True"
| " is_atomic_formula (x) = False"
end

Bozzato defines a language for the specification of the ontology as it has not been formalized and not specified in a theorem prover tool. As the language used to define Web service semantics and no evidence of its correctness was provided. The use of this language in modeling an ontology for describing the environment and services has never been discussed or proposed in the prior state of the art [24]. To our knowledge this is the first attempt to formalize the logic and its use in the context of ambient intelligence applications. The language proposed by Bozzato is used in the context of Aml and the evidence proved on Isabelle/HOL. Note that no computation was provided to the information terms algorithm. This is the first attempt to our knowledge for studying of the CDL in the field of systems Aml.

7. Conclusion

In this paper, the challenges in AmI development trends were discussed. By considering these applications as critical, we conclude that more investigation must be done on the formal specification and verification. A model to describe this kind of critical systems and to prove the correctness is proposed. It is based on the constructive description logics and its evidence proved in Isabelle/Hol. An integrate framework for building a reliable AAL applications is under development.

References


