Formal Modeling of Semantic Based Cooperative Ubicomp Services

Mohamed Hilia, Abdelghani Chibani, Yacine Amirat and Karim Djouani

LISSI Laboratory Paris-Est Créteil University, France

Problem Statement

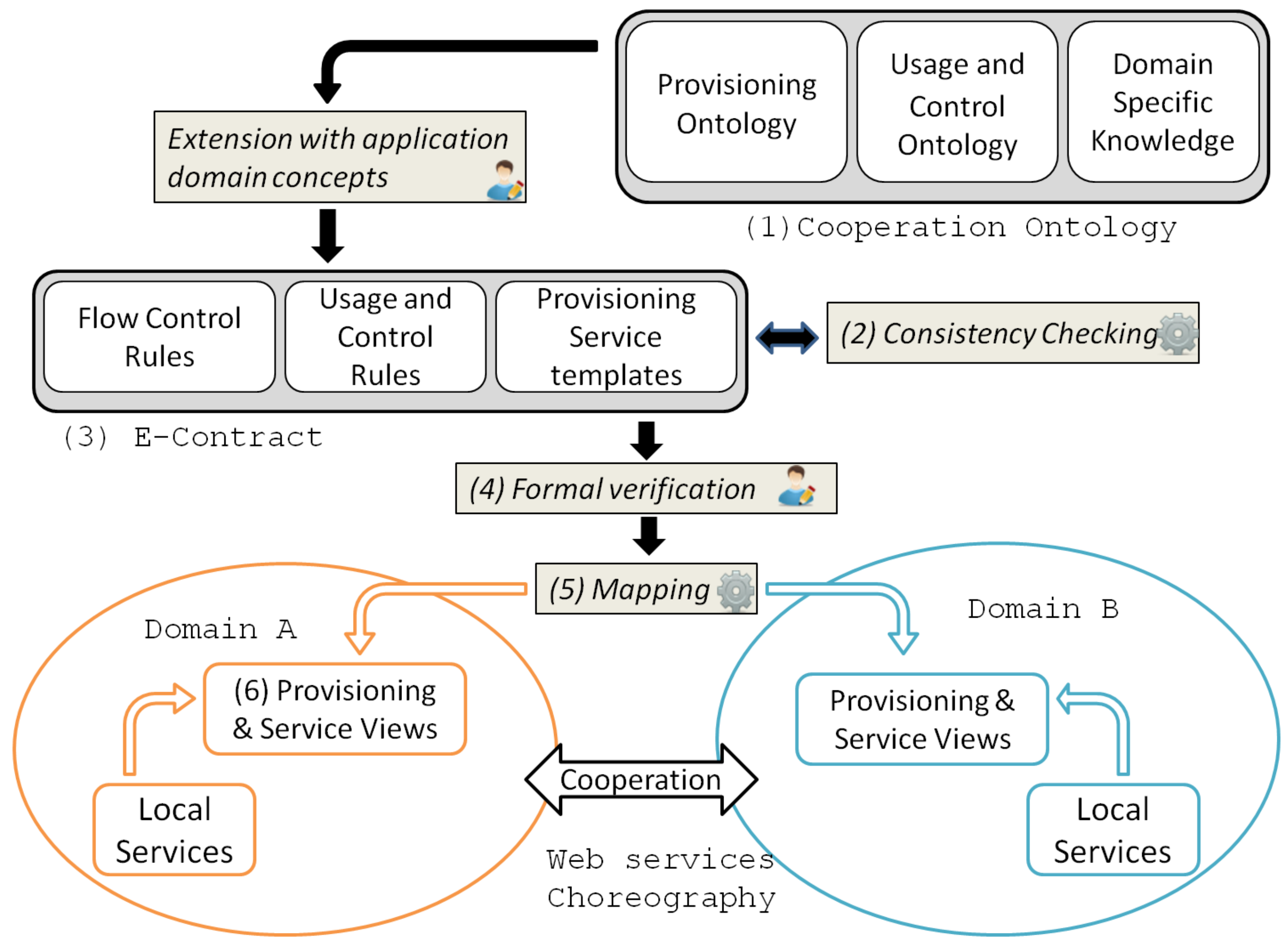
- Semantic service composition to build complex applications that run in multi domain environments
- The cooperation between heterogeneous service-oriented ubiquitous computing domains
- How to proof the *composability* of complex services composition schemas

Approach Overview

- Semantic formalization of multi domain service composition based on Constructive Description Logic (BCDL0) :
  - Cooperation ontology
  - E-Contract (Services enactments)
- Functional and non-functional view-based modeling of services
- Implementation of the formal proof of "Service composition soundness" by using Isabelle/HOL theorem proving tool

- Hybrid composition approach that preserves the autonomy and confidentiality of the local domains.

Multi Domain Semantic Services Composition Framework



Composition Methodology

1. Semantic Description of the cooperation
2. Consistency Checking
3. Local Process Views Creation
4. BCDL0 specification of views composition
5. Encoding & Soundness proof in Isabelle-HOL
6. Grounding : Mapping Views to Concrete Services, Services choreography , WS-Management and SPML standards

Why BCDL0 ?

- BCDL0 has been used by Bozzato et al [1] to formalize information services composition using three composition operators
- BCDL0, is a subsystem of the constructive description logic BCDL that is used to make constructive interpretation of ALC formulas.
- ***BCDL0 : Syntax*** :

$$C, D := A \; |\neg C \; |C \sqcup D \; |C \sqcap D \; |\exists R.\, C \; |\forall R.\, C$$

Where $C,\ D \in \mathrm{NC}$ and $R \in \mathrm{NR}$. The generated formulas are as follows :

$$K := \quad \bot \,|\, t : C \,|\, A \sqsubseteq C \,| \\ (s\,,\ t) : R \text{ where } \mathrm{s,t} \in \mathrm{NI} \cup \mathrm{Var}$$

- ***BCDL0 : Semantics*** **:** BCDL0 is based on the notion of information term [1]. Intuitively, an information term for a closed formula K is a structured object that provides a justification for the validity of K in a classical model. The information term is defined by induction on K :

$$\begin{array}{ll}
IT_{\mathcal{N}}(K) & = \{tt\}\,, \text{iff K is a } \textit{closed formula} \\
IT_{\mathcal{N}}(c : C_1 \sqcap C_2) & = \{(\alpha, \beta) | \alpha \in IT_{\mathcal{N}}(c : C_1) \text{ and } \beta \in IT_{\mathcal{N}}(c : C_2)\} \\
IT_{\mathcal{N}}(c : C_1 \sqcup C_2) & = \{(k, \alpha) | k \in 1, 2 \; and \; \alpha \in IT_{\mathcal{N}}(c : C_k)\} \\
IT_{\mathcal{N}}(c : \exists\, R.C) & = \{(d, \alpha) | d \in \mathcal{N} \; and \; \alpha \in IT_{\mathcal{N}}(d : C)\} \\
IT_{\mathcal{N}}(c : \forall\, R.C) & = \left\{\phi : \mathcal{N} \rightarrow \bigcup_{d \in \mathcal{N}} IT_{\mathcal{N}}(d : C) |\ \phi(d) \in IT_{\mathcal{N}}(d : C)\right\} \\
IT_{\mathcal{N}}(A \sqsubseteq C) & = \left\{\phi : \mathcal{N} \rightarrow \bigcup_{d \in \mathcal{N}} IT_{\mathcal{N}}(d : C) |\ \phi(d) \in IT_{\mathcal{N}}(d : C)\right\}
\end{array}$$

Multi-domain Cooperation Ontology (sample)

$$\begin{array}{l}
Add \sqsubseteq ProvisioningMessage \\
AutomaticTask \sqsubseteq \neg ManualTask \\
Lookup \sqsubseteq ProvisioningMessage \\
ManualTask \sqsubseteq \neg AutomaticTask \\
ProvisioningAction \sqsubseteq Action \\
ProvisioningTask \sqsubseteq \exists\, executedBy\ Domain \\
Search \sqsubseteq ProvisioningMessage \\
\exists\, executedBy\ Thing \sqsubseteq Task \\
\exists\, hasProvisioningMessage\ Thing \sqsubseteq ProvisioningAction \\
\top \sqsubseteq \forall\, hasPerformer\ Role \\
\top \sqsubseteq \forall\, hasProvisioningMessage\ ProvisioningMessage \\
\top \sqsubseteq \forall\, hasState\ TaskState
\end{array}$$

Composition Rules Formalization in BCDL0

$$p\,(x) :: P \Longrightarrow Q$$

Service Specification

P : Preconditions

Q : Effects

$$\frac{p\,(x) :: P \Longrightarrow Q}{\begin{array}{l} \varPi_1 : p_1\,(x) :: P_1 \Longrightarrow Q_1 \\ \varPi_2 : p_2\,(x) :: P_2 \Longrightarrow Q_2 \\ \dots \\ \varPi_n : p_n\,(x) :: P_n \Longrightarrow Q_n \end{array}}\ rule$$

Flow Control Rule template, with :

- *rule* : rule name
- *x* : input Parameter
- *Pi* : list of the services

$$\frac{p\,(x) :: P \Rightarrow Q}{\begin{array}{l} \varPi_1 : p_1\,(x) :: P_1 \Longrightarrow Q_1 \\ \varPi_2 : p_2\,(x) :: P_2 \Longrightarrow Q_2 \\ \vdots \\ \varPi_n : p_n\,(x) :: P_n \Longrightarrow Q_n \end{array}}\ Sequence \qquad AC = \begin{cases} T, x : P \,|_{\overline{\mathcal{BCDL}_0}}\, x : P_1 \\ T, x : Q_{k-1} \,|_{\overline{\mathcal{BCDL}_0}}\, x : P_k, \text{for } k \in \{2, \dots, n\} \\ T, x : Q_n \,|_{\overline{\mathcal{BCDL}_0}}\, x : Q \end{cases}$$

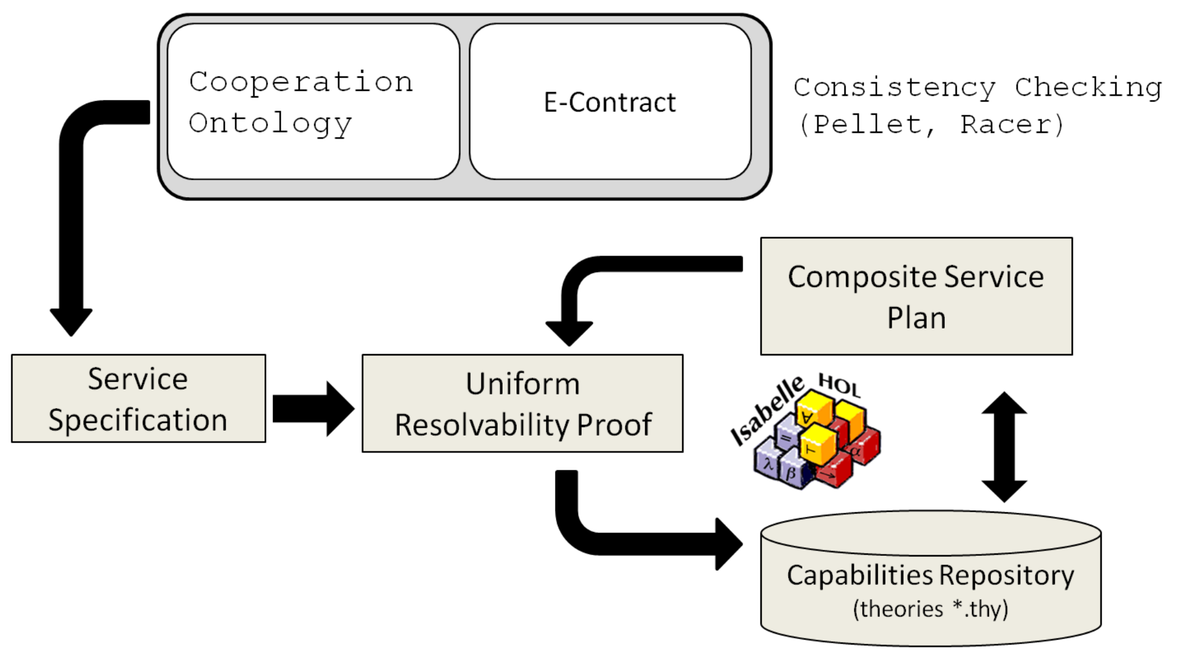
$$\frac{p\,(x) :: P \Rightarrow Q}{\begin{array}{l} \varPi_1 : p_1\,(x) :: P_1 \Longrightarrow Q_1 \\ \varPi_2 : p_2\,(x) :: P_2 \Longrightarrow Q_2 \\ \vdots \\ \varPi_n : p_n\,(x) :: P_n \Longrightarrow Q_n \end{array}}\ Parallel\ Split \qquad AC = \begin{cases} T, x : P \,|_{\overline{\mathcal{BCDL}_0}}\, x : P_k, \text{for } \ k \in \{1, \dots, n\} \\ T, x : Q_1 \sqcap \dots \sqcap Q_n \,|_{\overline{\mathcal{BCDL}_0}}\, x : Q \end{cases}$$

$$\frac{p\,(x) :: P \Longrightarrow Q}{\begin{array}{l} \varPi_1 : p_1\,(x) :: P_1 \Longrightarrow Q_1 \\ \varPi_2 : p_2\,(x) :: P_2 \Longrightarrow Q_2 \\ \vdots \\ \varPi_n : p_n\,(x) :: P_n \Longrightarrow Q_n \end{array}}\ Synchronization \qquad AC = \begin{cases} T, x : P_1 \sqcap \dots \sqcap P_n \,|_{\overline{\mathcal{BCDL}_0}}\, x : P \\ T, x : Q_k \,|_{\overline{\mathcal{BCDL}_0}}\, x : Q, \ \text{for } \ k \in \{1, \dots, n\} \end{cases}$$

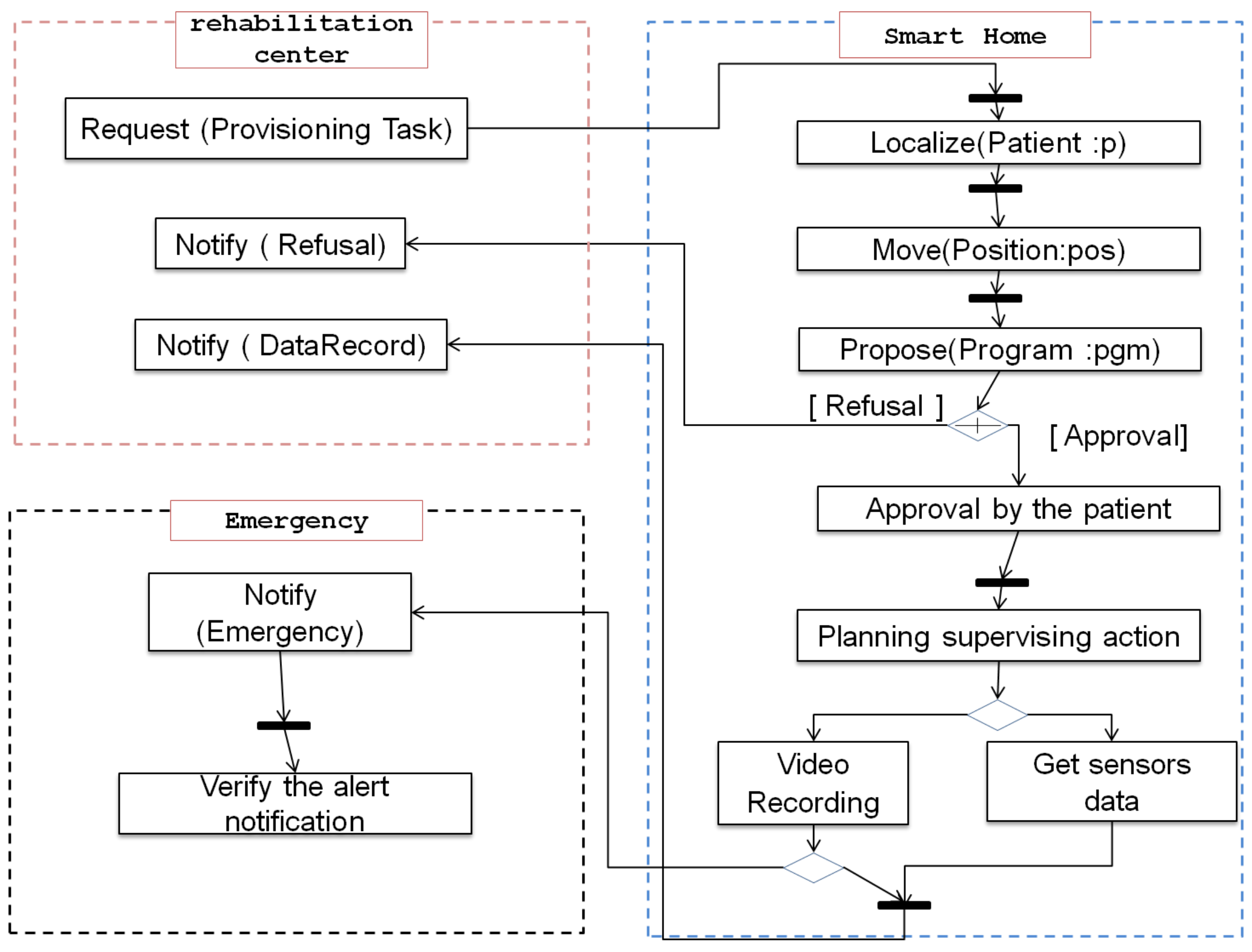
$$\frac{p\,(x) :: P \Longrightarrow Q}{\begin{array}{l} \varPi_1 : p_1\,(x) :: P_1 \Longrightarrow Q_1 \\ \varPi_2 : p_2\,(x) :: P_2 \Longrightarrow Q_2 \\ \vdots \\ \varPi_n : p_n\,(x) :: P_n \Longrightarrow Q_n \end{array}}\ Exclusive\ Choice \qquad AC = \begin{cases} T, x : P \,|_{\overline{\mathcal{BCDL}_0}}\, x : P_1 \sqcup \dots \sqcup P_n \\ T, x : Q_k \,|_{\overline{\mathcal{BCDL}_0}}\, x : Q, \text{for } \ k \in \{1, \dots, n\} \end{cases}$$

$$\frac{p\,(x) :: P \Longrightarrow Q}{\begin{array}{l} \varPi_1 : p_1\,(x) :: P_1 \Longrightarrow Q_1 \\ \varPi_2 : p_2\,(x) :: P_2 \Longrightarrow Q_2 \\ \vdots \\ \varPi_n : p_n\,(x) :: P_n \Longrightarrow Q_n \end{array}}\ Simple\ Merge \qquad AC = \begin{cases} T, x : P \,|_{\overline{\mathcal{BCDL}_0}}\, x : P_k, \ \text{for } k \in \{1, \dots, n\} \\ T, x : Q_1 \sqcup \dots \sqcup Q_n \,|_{\overline{\mathcal{BCDL}_0}}\, x : Q \end{cases}$$

Consistency Checking and Soundness Proof



Multi domain healthcare services composition



Outcomes and ongoing works

- Extension of BCDL0 composition rules with synchronization and simple merge rules.
- Composition Views that handle services provisioning issues.
- Ongoing Work : (i) Implementation of ubicomp healthcare scenario involving smart home and hospital (ii) Prove automation in Isabelle /HOL

References

[1] L. Bozzato and M. Ferrari. "A note on semantic web services specification and composition in constructive description logics". *Journal of Syntax And Semantics*, 2010.

[2] M. Hilia, A. Chibani, K. Djouani, and Y. Amirat. "Semantic Service Composition Framework for Multidomain Ubiquitous Computing Applications". In prooceeding Of the *International Conference on Service Oriented Computing*, 2012

Contact

Mohamed Hilia, mohamed.hilia@gmail.com