# An Extension of Distributed Dynamic Description Logics for the Representation of Heterogeneous Mappings

Zhuxiao Wang
School of Control and Computer Engineering, State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, 102206 Beijing, China
Email: wangzx@ncepu.edu.cn

Jing Guo
National Computer Network Emergency Response Technical Team/Coordination Center of China, 100029 Beijing, China
Email: guojing.research@gmail.com

Fei Chen and Kehe Wu
School of Control and Computer Engineering, North China Electric Power University, 102206 Beijing, China
Email: {chenfei, wkh}@ncepu.edu.cn

Peng Wang
Institute of Information Engineering, Chinese Academy of Sciences, 100195 Beijing, China
Email: wangpeng@iie.ac.cn

***Abstract*—As a family of dynamic description logics, DDL(X) is constructed by embracing actions into the description logic X, where X represents well-studied description logics ranging from the ALC to the SHOIQ. To efficiently support automated interoperability between ontology-based information systems in distributed environments, we have to design an expressive mapping language to semantically understand resources from remote and heterogeneous systems. Distributed Dynamic Description Logics D3L(X) is a natural generalization of the DDL(X) framework, which is designed to model the distributed dynamically-changing knowledge repositories interconnected by semantic mappings and to accomplish reasoning in distributed, heterogeneous environments. In this paper, we propose an extension of Distributed Dynamic Description Logics D3L(X) and investigate the reasoning mechanisms in D3L(X).**

***Index Terms*—distributed reasoning, dynamic description logics, distributed dynamic description logics, tableau algorithms, semantic mappings**

## I. Introduction

Description Logics (DLs) are a family of formal knowledge representation languages which structure the knowledge about an application domain in terms of concepts (subsets of individuals in the domain) and roles (binary relations over the domain). Description Logics are playing a central role in knowledge representation, acting as the basis of the well known traditions of Frame-based systems, Semantic Networks and KL-ONE-like languages, Object-Oriented representations, Semantic data models, and Type systems [1-7].

By introducing a dynamic dimension into the description logics, Shi et al [8][9] propose a family of Dynamic Description Logics named DDL(X) for uniformly representing and reasoning about dynamic application domains [10][11], where X represents well-studied description logics ranging from the $\mathcal{ALC}$ to the $\mathcal{SHOIQ}$.

To efficiently support automated interoperability between ontology-based information systems in distributed environments, the problem of establishing semantic relations between heterogeneous components has to be dealt with. In many real cases[12], there is a compelling need for expressing mappings between the components of heterogeneous ontologies. For example to map a concept into an action or vice versa. Thereby, in this paper, we propose an extension of Distributed Dynamic Description Logics D3L(X) capable of capturing the dynamic behavior of the overall system. D3L(X) is a natural generalization of the DDL(X) framework [8][9], which is designed to model the distributed dynamically-changing knowledge repositories interconnected by semantic mappings and to accomplish reasoning in distributed, heterogeneous environments. Afterwards, we study the realizability, executability, and

Manuscript received May 31, 2012; revised June 25, 2012; accepted XX X, 2012.
Copyright credit, project number, corresponding author: Zhuxiao Wang.


doi:10.4304/jsw.8.1.243-250

projection problems on D3L(X)-actions. It is demonstrated that the three primary reasoning tasks on actions can be reduced to the satisfiability problem on formulas.

Our contributions in this paper are as follows: (i) we define the semantics and syntax of D3L(X) to formally capture the dynamic behavior of the overall system; (ii) use actions as modal operators in the construction of formulas, so that many reasoning tasks on actions can be reduced to the satisfiability problem of formulas and therefore are still decidable; and finally, (iii) analyze semantical mechanisms allowing for propagating the dynamic knowledge, i.e. how dynamic knowledge propagates through local reasoning engines.

In the following sections, we firstly present the syntax and semantics of Distributed Dynamic Description Logics D3L(X) in Section 2 and Section 3 respectively. We recall the basic definitions of D3L(X), and we provide an extension of D3L(X) to represent heterogeneous mappings. In Section 4, it is demonstrated that three primary reasoning tasks on actions can be reduced to the satisfiability problem on formulas. Furthermore, in Section 4 we study the main properties of the proposed D3L(X). Finally, we summarize the paper in Section 5.

## II. Distributed Dynamic Description Logics: the Syntax

In this section, we present the basic definitions of the Distributed Dynamic Description Logics D3L(X) formalism. From a theoretical perspective, the D3L(X) is based on the long tradition of logics for distributed systems, and based on extensions to Dynamic Description Logics introduced in [8][9]. If we do not consider the dynamic dimension of D3L(X), D3L(X) can be reduced to Distributed Description Logics [13][14]. Let I be a nonempty set of indexes, and $DDL_i$ be dynamic description logics for every $i \in I$. A sequence $\mathcal{D3L} = \{DDL_i\}_{i\in I}$ is then called a distributed dynamic description logic. We label each description $C$ in $DDL_i$ with its index i (written as i:$C$) to indicate that some description $C$ belongs to the language of the dynamic description logic $DDL_i$. Collections of bridge rules are used to express relations between the components of a Distributed Dynamic Description logic. In the following we use $C$ and $G$ as placeholders for concepts and $\alpha$ and $\beta$ as placeholders for actions.

**Definition 1**. A bridge rule from i to j is an expression defined as follows:

$i{:}C \xrightarrow{\sqsubseteq} j{:}G$ concept-into-concept bridge rule (1)

$i{:}C \xrightarrow{\sqsupseteq} j{:}G$ concept-onto-concept bridge rule (2)

$i{:}\alpha \xrightarrow{\sqsubseteq} j{:}\beta$ action-into-action bridge rule (3)

$i{:}\alpha \xrightarrow{\sqsupseteq} j{:}\beta$ action-onto-action bridge rule (4)

$i{:}C \xrightarrow{\sqsubseteq} j{:}\alpha$ concept-into-action bridge rule (5)

$i{:}C \xrightarrow{\sqsupseteq} j{:}\alpha$ concept-onto-action bridge rule (6)

$i{:}\alpha \xrightarrow{\sqsubseteq} j{:}C$ action-into-concept bridge rule (7)

$i{:}\alpha \xrightarrow{\sqsupseteq} j{:}C$ action-onto-concept bridge rule. (8)

where $C$ and $G$ are concepts of $DDL_i$ and $DDL_j$ respectively, and $\alpha$ and $\beta$ are actions of $DDL_i$ and $DDL_j$ respectively. Bridge rules (1)–(4) are called homogeneous bridge rules, and bridge rules (5)–(8) are called heterogeneous bridge rules.

Let $p$ be an individual of $DDL_i$ and $q$ individuals of $DDL_j$. An individual correspondence is an expression of the form

$i{:}p \longrightarrow j{:}q$ individual correspondence. (9)

Formulas of D3L(X) are formed according to the following syntax rule:

$$\varphi, \varphi' ::= C(p) \mid R(p, q) \mid <\pi> \varphi \mid [\pi]\varphi \mid \neg\varphi \mid \varphi \vee \varphi' \mid \varphi \wedge \varphi'$$

where $p, q \in N_I$ (the set of individual names), $C$ is a concept, $R$ is a role, and $\pi$ is an action. Formulas of the form $C(p)$, $R(p, q)$, $<\pi> \varphi$, $[\pi]\varphi$, $\neg\varphi$, $\varphi\vee\varphi'$ and $\varphi\wedge\varphi'$ are respectively called concept assertion, role assertion, diamond assertion, box assertion, negation formula, disjunction formula, and conjunction formula.

Actions of D3L(X) are formed according to the following syntax rule:

$$\pi, \pi' ::= \alpha \mid \varphi? \mid \pi \cup \pi' \mid \pi \cap \pi' \mid \pi ; \pi' \mid \pi^*$$

where $\alpha \in N_A$, and $\varphi$ is a formula. Actions of the form $\alpha$, $\varphi$?, $\pi \cup \pi'$, $\pi \cap \pi'$, $\pi ; \pi'$ and $\pi^*$ are respectively called atomic action, test action, choice action, conjunction action, sequence action and iteration actions.

A distributed TBox (DTBox) $\mathcal{DT} = <\{T_i\}_{i\in I}, \mathcal{P}>$ consists of a collection of T-boxes $\{T_i\}_{i\in I}$ and a set $\mathcal{P} = \{\mathcal{P}_{ij}\}_{i\neq j\in I}$ of concept bridge rules. A distributed ABox (DABox) $\mathcal{DA} = <\{A_i\}_{i\in I}, \mathcal{C}>$ consists of a collection of A-boxes $\{A_i\}_{i\in I}$ together with a set $\mathcal{C} = \{\mathcal{C}_{ij}\}_{i\neq j\in I}$ of individual correspondences. A distributed ActBox (DActBox) $\mathcal{DAct} = <\{Act_i\}_{i\in I}, \mathcal{H}>$ consists of a collection of ActBoxes $\{Act_i\}_{i\in I}$ and a set $\mathcal{H} = \{\mathcal{H}_{ij}\}_{i\neq j\in I}$ of action bridge rules or heterogeneous bridge rules. A distributed dynamic knowledge base is a triple $\mathcal{K} = (\mathcal{DT}, \mathcal{DA}, \mathcal{DAct})$.

## III. Distributed Dynamic Description Logics: the Semantics

The semantics of a Distributed Dynamic Knowledge Base (DDKB) is formally defined as follows.

**Definition 2**. A distributed model $\mathcal{M}$ of a DDKB $\mathcal{K} = (\mathcal{DT}, \mathcal{DA}, \mathcal{DAct})$ is a tuple $< \{M_i = (W_i, T_i, \Delta^{I_i}, I_i)\}_{i\in I}, \{r_{ij}\}_{i\neq j\in I}, \{state_{ij}\}_{i\neq j\in I}, \{sc_{ij}\}_{i\neq j\in I}, \{cs_{ij}\}_{i\neq j\in I} >$, where, Each $M_i$ is a local model for the corresponding $DDL_i$ on local domains $\Delta^{I_i}$;

$W_i$ is a set of states;

$T_i : N_A \rightarrow 2^{W\times W}$ is a function mapping action names into binary relations on $W_i$;

$\Delta^{I_i}$ is a non-empty domain;

$I_i$ is a function which associates with each state $w \in W_i$ a description logic interpretation $I_i(w) = < \Delta^{I_i}, \bullet^{I_i(w)} >$, where the mapping $\bullet^{I_i(w)}$ assigns each concept to a subset of $\Delta^{I_i}$, each role to a subset of $\Delta^{I_i} \times \Delta^{I_i}$, and each individual to an element of $\Delta^{I_i}$.

A domain relation $r_{ij}$ from $\Delta^{I_i}$ to $\Delta^{I_j}$ is defined as a subset of $\Delta^{I_i} \times \Delta^{I_j}$. Given a point $d \in \Delta^{I_i}$ and a subset $D \subseteq \Delta^{I_i}$, we set

$$r_{ij}(d) = \{d' \in \Delta^{I_j} \mid (d, d') \in r_{ij}\},\ r_{ij}(D) = \bigcup_{d\in D} r_{ij}(d). \tag{10}$$

A state relation $state_{ij}$ from $W_i$ to $W_j$ is defined as a subset of $W_i \times W_j$. Given a point $w \in W_i$ and a subset $T_i(\alpha) \subseteq (W_i \times W_i)$, we set

$$state_{ij}(w) = \{w' \in W_j \mid (w, w') \in state_{ij}\}, \tag{11}$$

$$state_{ij}(T_i(\alpha)) = \bigcup_{(w,w')\in T_i(\alpha)} state_{ij}(w) \times state_{ij}(w'). \tag{12}$$

A concept-action relation $cs_{ij}$ from $\Delta^{I_i}$ to $W_j$ is a subset of $\Delta^{I_i} \times W_j \times W_j$. A action-concept relation $sc_{ij}$ from $W_i$ to $\Delta^{I_j}$ is a subset of $W_i \times W_i \times \Delta^{I_j}$.

We use $cs_{ij}(d)$ to denote $\{< w, w' > \in W_j \times W_j \mid < d, w, w' > \in cs_{ij}\}$; for any subset D of $\Delta^{I_i}$, we use $cs_{ij}(D)$ to denote $\bigcup_{d\in D} cs_{ij}(d)$. We use $sc_{ij}(< w, w' >)$ to denote $\{ d \in \Delta^{I_j} \mid < w, w', d > \in sc_{ij}\}$; for any subset S of $W_i \times W_i$, we use $sc_{ij}(S)$ to denote $\bigcup_{< w,w' > \in S} sc_{ij}(< w, w' >)$.

A concept-action relation $cs_{ij}$ represents a possible way of mapping elements of $\Delta^{I_i}$ into pairs of states in $W_j$, seen from j's perspective. For instance, $\Delta^{I_i}$ and $W_j$ are the representation of a web service system in which customers are able to buy/return books online with credit cards (see Fig. 1). A concept-action relation $cs_{ij}$ could be the function mapping bill numbers into the corresponding buyBook actions. For instance, by setting

$cs_{ij}(\text{BillNumberOfKingLear}^{I_i}) = \{(w, w') \in T_j(\text{buyBook})\}$

we can represent the fact that the bill number of KingLear is associated with pairs of states ($w$, $w'$) such that the execution of atomic action buyBook is interpreted as binary relations on states. Vice-versa a action-concept relation $sc_{ij}$ represents a possible way of mapping a pair of $W_i$ into the corresponding element in $\Delta^{I_j}$.

With respect to any state $w \in W_i$, a distributed model $\mathcal{M}$ is said to d-satisfy (written $(\mathcal{M},w) \vDash_d$) concept bridge rules and individual correspondences according to the following clauses:

$(\mathcal{M},w) \vDash_d i{:}C \xrightarrow{\sqsubseteq} j{:}G$ iff $r_{ij}(C^{I_i(w)}) \subseteq \bigcap_{w' \in state_{ij}(w)} G^{I_j(w')}$,

concept into-bridge rule

$(\mathcal{M},w) \vDash_d i{:}C \xrightarrow{\sqsupseteq} j{:}G$ iff $r_{ij}(C^{I_i(w)}) \supseteq \bigcup_{w' \in state_{ij}(w)} G^{I_j(w')}$,

concept onto-bridge rule

$(\mathcal{M},w) \vDash_d i{:}p \longrightarrow j{:}q$ iff $\bigcup_{w' \in state_{ij}(w)} q^{I_j(w')} \subseteq r_{ij}(p^{I_i(w)})$. individual correspondence

Secondly, the satisfaction of an action bridge rule $br$ in $\mathcal{M}$, written as $\mathcal{M} \vDash_d br$, is defined as follows:

$\mathcal{M} \vDash_d i{:}\alpha \xrightarrow{\sqsubseteq} j{:}\beta$ iff $state_{ij}(T_i(\alpha)) \subseteq T_j(\beta)$,

action into-bridge rule

$\mathcal{M} \vDash_d i{:}\alpha \xrightarrow{\sqsupseteq} j{:}\beta$ iff $state_{ij}(T_i(\alpha)) \supseteq T_j(\beta)$.

action onto-bridge rule

Thirdly, the concept-action relation $cs_{ij}$ satisfies a concept to action bridge rule w.r.t., $M_i$ and $M_j$, in symbols $< M_i, cs_{ij}, M_j > \vDash$ br, according with the following

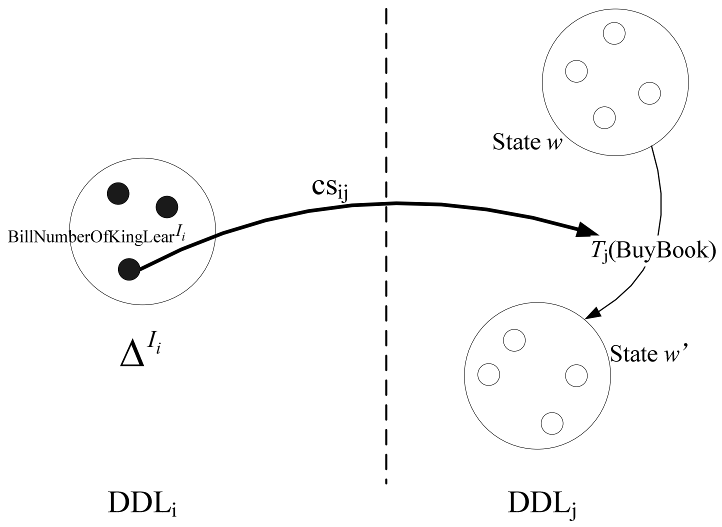


Figure 1. Concept-Action relation.

definition:

$<M_i, cs_{ij}, M_j> \vDash_d$ i:$C \xrightarrow{\sqsubseteq}$ j:$\alpha$ iff $cs_{ij}(C^{I_i}) \subseteq T_j(\alpha)$

$<M_i, cs_{ij}, M_j> \vDash_d$ i:$C \xrightarrow{\sqsupseteq}$ j:$\alpha$ iff $cs_{ij}(C^{I_i}) \supseteq T_j(\alpha)$

where $C$ is a concept expression of i and $\alpha$ an action expression of j.

Fourthly, the action-concept relation $sc_{ij}$ satisfies a action to concept bridge rule w.r.t., $M_i$ and $M_j$, in symbols $<M_i, sc_{ij}, M_j> \vDash$ br, according with the following definition:

$<M_i, sc_{ij}, M_j> \vDash_d$ i:$\alpha \xrightarrow{\sqsubseteq}$ j:$C$ iff $sc_{ij}(T_i(\alpha)) \subseteq C^{I_i}$

$<M_i, sc_{ij}, M_j> \vDash_d$ i:$\alpha \xrightarrow{\sqsupseteq}$ j:$C$ iff $sc_{ij}(T_i(\alpha)) \supseteq C^{I_i}$

where $C$ is a concept expression of j and $\alpha$ an action expression of i.

The $\mathcal{M} \vDash_d$ is standard for formulas of the component Dynamic Description Logics [9]. With respect to any state $w \in W_i$, the truth-relation $(M_i,w) \vDash \varphi$ for a formula i:$\varphi$ is defined inductively as follows:

$(M_i,w) \vDash$i:$C(p)$ iff $p^{I_i(w)} \in C^{I_i(w)}$;

$(M_i,w) \vDash$i:$R(p, q)$ iff $(p^{I_i(w)}, q^{I_i(w)}) \in R^{I_i(w)}$;

$(M_i,w) \vDash$i:$<\pi>\varphi$ iff $\exists w' \in W_i.((w, w') \in T_i(\pi)$ and $(M_i,w') \vDash$ i:$\varphi)$;

$(M_i,w) \vDash$i: $[\pi]\varphi$ iff $\forall w' \in W_i.((w, w') \in T_i(\pi)$ implies $(M_i,w') \vDash$ i:$\varphi)$;

$(M_i,w) \vDash$i:$\neg\varphi$ iff it is not the case that $(M_i,w) \vDash$ i:$\varphi$;

$(M_i,w) \vDash$i:$\varphi \vee \psi$ iff $(M_i,w) \vDash$ i:$\varphi$ or $(M_i,w) \vDash$ i:$\psi$;

$(M_i,w) \vDash$i:$\varphi \wedge \psi$ iff $(M_i,w) \vDash$ i:$\varphi$ and $(M_i,w) \vDash$ i:$\psi$;

Finally, each action i:$\pi$ will be interpreted as a binary relation $T_i(\pi) \subseteq W_i \times W_i$ according to the following inductive definitions:

$T_i(\varphi?) = \{(w, w) \mid w \in W_i$ and $(M_i, w) \vDash$ i:$\varphi\}$;

$T_i(\pi \cup \pi') = T_i(\pi) \cup T_i(\pi')$;

$T_i(\pi \cap \pi') = T_i(\pi) \cap T_i(\pi')$;

$T_i(\pi ; \pi') = \{(w,w') \mid \exists u.(w, u) \in T_i(\pi)$ and $(u, w') \in T_i(\pi')\}$;

$T_i(\pi^*)$ = reflexive and transitive closure of $T_i(\pi)$.

For i:$\alpha$ and i:$\beta$ (possibly complex) actions, i:$\alpha \sqsubseteq \beta$ is called a general action inclusion, and a finite set of general action inclusions is called a ActBoxes. An interpretation $T_i$ satisfies a general action inclusion i:$\alpha \sqsubseteq \beta$ if $T_i(\alpha) \subseteq T_i(\beta)$.

A distributed model $\mathcal{M}$ satisfies the elements of a DTBox $\mathcal{DT}$ according to the following clauses:

1. $\mathcal{M} \vDash_d$ i: A⊑B, if $M_i \vDash$ A⊑B

2. $\mathcal{M} \vDash_d T_i$, if $\mathcal{M} \vDash_d$ i: A⊑B for all A⊑B in $T_i$

3. $\mathcal{M} \vDash_d \mathcal{P}_{ij}$, if $\mathcal{M}$ satisfies all the homogeneous concept bridge rules in $\mathcal{P}_{ij}$

4. $\mathcal{M} \vDash_d \mathcal{DT}$, if for every i, j $\in I$, $\mathcal{M} \vDash_d T_i$ and $\mathcal{M} \vDash_d \mathcal{P}_{ij}$

As usual, $\mathcal{DT} \vDash_d$ i:C⊑D means that for every distributed model $\mathcal{M}$, $\mathcal{M} \vDash_d \mathcal{DT}$ implies $\mathcal{M} \vDash_d$ i:C⊑D.

Concerning the assertional part, a distributed model $\mathcal{M}$ is said to satisfy the elements of a distributed ABox $\mathcal{DA}$ if

1. $M_i \vDash \varphi$ for all formulas $\varphi$ in $A_i$

2. $\mathcal{M} \vDash_d$ i:$p \longrightarrow$ j:$q$, if $\bigcup_{w' \in state_{ij}(w)} q^{I_j(w')} \subseteq r_{ij}(p^{I_i(w)})$

3. $\mathcal{M} \vDash_d \mathcal{DA}$, if for every i,j $\in I$, $\mathcal{M} \vDash_d A_i$ and $\mathcal{M} \vDash_d \mathcal{C}_{ij}$

As usual, $\mathcal{DA} \vDash_d$ i:$\varphi$ if for every $\mathcal{M}$, $\mathcal{M} \vDash_d \mathcal{DA}$ implies $\mathcal{M} \vDash_d$ i:$\varphi$.

Finally, a distributed model $\mathcal{M}$ satisfies the elements of a distributed ActBox $\mathcal{DAct}$ according to the following clauses:

1. $\mathcal{M} \vDash_d$ i: $\alpha \sqsubseteq \beta$, if $M_i \vDash \alpha \sqsubseteq \beta$

2. $\mathcal{M} \vDash_d Act_i$, if $\mathcal{M} \vDash_d$ i: $\alpha \sqsubseteq \beta$ for all $\alpha \sqsubseteq \beta$ in $Act_i$

3. $\mathcal{M} \vDash_d \mathcal{H}_{ij}$, if

-$\mathcal{M}$ satisfies all the action bridge rules in $\mathcal{H}_{ij}$

-$<M_i, cs_{ij}, M_j>$ satisfies all the concept-to-action bridge rules in $\mathcal{H}_{ij}$

-$<M_i, sc_{ij}, M_j>$ satisfies all the action-to-concept bridge rules in $\mathcal{H}_{ij}$

4. $\mathcal{M} \vDash_d \mathcal{DAct}$, if for every i, j $\in I$, $\mathcal{M} \vDash_d Act_i$ and $\mathcal{M} \vDash_d \mathcal{H}_{ij}$

As usual, $\mathcal{DAct} \vDash_d$ i: $\alpha \sqsubseteq \beta$ means that for every distributed model $\mathcal{M}$, $\mathcal{M} \vDash_d \mathcal{DAct}$ implies $\mathcal{M} \vDash_d$ i: $\alpha \sqsubseteq \beta$.

## IV. REASONING TASKS FOR D3L(X)

Let $\mathcal{K}$ = ($\mathcal{DT}$, $\mathcal{DA}$, $\mathcal{DAct}$) be a distributed dynamic knowledge base of D3L(X), where $\mathcal{DT}$, $\mathcal{DA}$, and $\mathcal{DAct}$ is a distributed TBox, a distributed ABox, and a distributed ActBox respectively. Based on such a knowledge base we investigate reasoning tasks for D3L(X).

The basic reasoning task for D3L(X) is to decide the satisfiability of formulas.

**Definition 3**. A formula i:$\varphi$ is satisfiable w.r.t. a distributed TBox $\mathcal{DT}$ and a distributed ActBox $\mathcal{DAct}$ if and only if there exists a model $\mathcal{M} = <\{M_i = (W_i, T_i, \Delta^{I_i}, I_i)\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I}, \{state_{ij}\}_{i \neq j \in I}, \{sc_{ij}\}_{i \neq j \in I}, \{cs_{ij}\}_{i \neq j \in I}>$ and a state $w \in W_i$ such that $\mathcal{M} \vDash_d \mathcal{DT}$, $\mathcal{M} \vDash_d \mathcal{DAct}$, and $(M_i, w) \vDash$ i:$\varphi$.

What distinguishes D3L(X) is the power for reasoning about actions. In this paper we study the realizability, executability, and projection problems on D3L(X)-actions.

Given an action i:$\pi$, we firstly want to known whether it is realizable, i.e., whether it makes sense with respect to the knowledge specified by a distributed TBox $\mathcal{DT}$ and a distributed ActBox $\mathcal{DAct}$. With D3L(X), the realizability of actions is formally defined as follows:

**Definition 4**. An action i: $\pi$ is realizable w.r.t. a distributed TBox $\mathcal{DT}$ and a distributed ActBox $\mathcal{DAct}$ if and only if there exists a model $\mathcal{M}$ = < $\{M_i = (W_i, T_i, \Delta^{I_i}, I_i)\}_{i\in I}$, $\{r_{ij}\}_{i\neq j\in I}$, $\{state_{ij}\}_{i\neq j\in I}$, $\{sc_{ij}\}_{i\neq j\in I}$, $\{cs_{ij}\}_{i\neq j\in I}$ > and two states $w$, $w' \in W_i$ such that $\mathcal{M} \vDash_d \mathcal{DT}$, $\mathcal{M} \vDash_d \mathcal{DAct}$, and $(w, w') \in T_i(\pi)$.

According to the definition 4, the following theorem is obvious:

**Theorem 1**. An action i: $\pi$ is realizable w.r.t. a distributed TBox $\mathcal{DT}$ and a distributed ActBox $\mathcal{DAct}$ if and only if the formula i:< $\pi$ > true is satisfiable w.r.t. $\mathcal{DT}$ and $\mathcal{DAct}$.

Secondly, if an action is realizable, we want to know whether it is executable on the state described by a given ABox [15][16], i.e., whether the action can be performed successfully starting from a given state.

Let $\alpha_1 \equiv (P_1,E_1),\ldots, \alpha_n \equiv (P_n,E_n)$ be the definitions of all the atomic actions which are occurring in i: $\pi$ and are defined w.r.t. $\mathcal{DAct}$. Let $\Pi$ be the formula $(Conj(P_1) \rightarrow <\alpha_1> \text{true}) \wedge \ldots \wedge (Conj(P_n) \rightarrow <\alpha_n> \text{true})$, where $Conj(P_i)$ represents the conjunction of all the elements of the set $P_i$. Then the executability of actions can be checked according to the following theorem:

**Theorem 2**. An action i: $\pi$ is executable on states described by an ABox $A_i$ if and only if the following formula is valid w.r.t. $\mathcal{DT}$ and $\mathcal{DAct}$:

$$[(\alpha_1 \cup \ldots \cup \alpha_n)^*]\Pi \rightarrow (\text{Conj}(A_i) \rightarrow \text{i:}<\pi>\text{true})$$

Thirdly, if an action is executable, we than want to know whether applying it achieves the desired effect, i.e., whether a formula that we want to make true really holds after executing the action. This kind of inference problem is called projection problem [15][16].

**Theorem 3**. i:$\psi$ is a consequence of applying i: $\pi$ on states described by $A_i$ if and only if the formula $Conj(A_i) \rightarrow$ i:[ $\pi$ ]$\psi$ is valid w.r.t. $\mathcal{DT}$ and $\mathcal{DAct}$.

Let us see now how action bridge rules affect the forward propagation of knowledge in D3L. The basic idea preceding that result is that combination of action onto- and into-bridge rules allows for directional propagating the action knowledge across knowledge repositories in form of DDL(X) action subsumption axioms [8][9].

**Theorem 4 (Sequence action propagation)**. If $\mathcal{H}_{ij}$ contains i: $\alpha \xrightarrow{\subseteq}$ j: $\beta$ and i: $\pi \xrightarrow{\subseteq}$ j: $\rho$, then:

$$\mathcal{DAct} \vDash_d \text{i}: \alpha ; \pi \Rightarrow \mathcal{DAct} \vDash_d \text{j}: \beta ; \rho \quad (13)$$

where $\alpha$, $\pi$, $\beta$, and $\rho$ are actions.

**Theorem 5 (Simple action subsumption propagation)**. Combination of action onto- and into-bridge rules allows to propagate action subsumptions across knowledge repositories (see Fig. 2). Formally, if $\mathcal{H}_{ij}$ contains i: $\alpha \xrightarrow{\supseteq}$ j: $\beta$ and i: $\pi \xrightarrow{\subseteq}$ j: $\rho$, then:

$$\mathcal{DAct} \vDash_d \text{i}: \alpha \sqsubseteq \pi \Rightarrow \mathcal{DAct} \vDash_d \text{j}: \beta \sqsubseteq \rho \ . \quad (14)$$

Example 1. Let

- $\mathcal{DAct} \vDash_d$ i:collectData $\xrightarrow{\supseteq}$ j:buyBook, and
- $\mathcal{DAct} \vDash_d$ i:collectData $\xrightarrow{\subseteq}$ j:shopping.

Theorem 2 allows to infer that a buyBook action is a shopping action in $W_j$, namely $\mathcal{DAct} \vDash_d$ j : buyBook $\sqsubseteq$ shopping, from the fact that $\mathcal{DAct} \vDash_d$ i : collectData $\sqsubseteq$ collectData.

**Theorem 6 (Generalized action subsumption propagation)**. If $\mathcal{H}_{ij}$ contains i: $\pi \xrightarrow{\supseteq}$ j: $\rho$ and i: $\alpha_k \xrightarrow{\subseteq}$ j: $\beta_k$ for i ≤ k ≤ n (n≥0), then:

$$\mathcal{DAct} \vDash_d \text{i}: \pi \sqsubseteq \cup_{k=1}^{n} \alpha_k \Rightarrow \mathcal{DAct} \vDash_d \text{j}: \rho \sqsubseteq \cup_{k=1}^{n} \beta_k. \quad (15)$$

Proof. Let's show that, for any distributed model $\mathcal{M}$ that satisfies $\mathcal{H}_{ij}$, if $T_i(\pi) \subseteq T_i(\cup_{k=1}^{n} \alpha_k)$, then $T_j(\rho) \subseteq T_j(\cup_{k=1}^{n} \beta_k)$. Indeed, $T_j(\rho) \subseteq state_{ij}(T_i(\pi)) \subseteq state_{ij}(T_i(\cup_{k=1}^{n} \alpha_k)) = \cup_{k=1}^{n} state_{ij}(T_i(\alpha_k)) \subseteq \cup_{k=1}^{n} T_j(\beta_k) = T_j(\cup_{k=1}^{n} \beta_k)$.

**Theorem 7 (Simple concept subsumption propagation)**. Combination of concept onto- and into-bridge rules allows to propagate subsumptions across knowledge repositories. Formally, if $\mathcal{P}_{ij}$ contains i:$A \xrightarrow{\subseteq}$ j:$F$ and i:$B \xrightarrow{\supseteq}$ j:$G$, then:

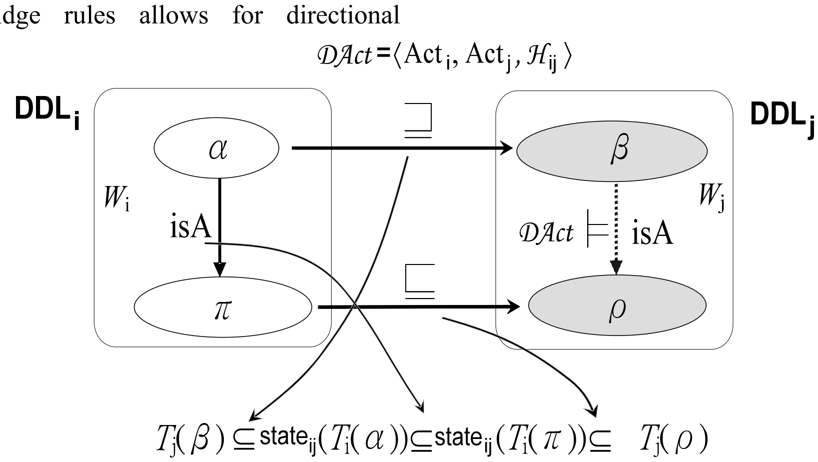


Figure 2. Graphical intuition of action subsumption propagation in D3L(X).

$$\mathcal{DT} \vDash_d i{:}B \sqsubseteq A \Rightarrow \mathcal{DT} \vDash_d j{:}\ G \sqsubseteq F\ . \tag{16}$$

**Theorem 8 (Generalized concept subsumption propagation)**. If $\mathcal{P}_{ij}$ contains $i{:}B \xrightarrow{\supseteq} j{:}G$ and $i{:}A_k \xrightarrow{\subseteq} j{:}F_k$ for $i \le k \le n$ ($n \ge 0$), then:

$$\mathcal{DT} \vDash_d i{:}B \sqsubseteq \sqcup_{k=1}^{n} A_k \Rightarrow \mathcal{DT} \vDash_d j{:}G \sqsubseteq \sqcup_{k=1}^{n} F_k\ . \tag{17}$$

**Theorem 9 (Concept into/onto action subsumption propagation)**. If $\mathcal{H}_{ij}$ contains $i{:}A \xrightarrow{\supseteq} j{:}\ \alpha$ and $i{:}B \xrightarrow{\subseteq} j{:}\ \beta$ (see Fig. 3), then:

$$\mathcal{M} \vDash_d i{:}A \sqsubseteq B \Rightarrow \mathcal{M} \vDash_d j{:}\ \alpha \sqsubseteq \beta\ . \tag{18}$$

**Theorem 10 (Generalized concept into/onto action subsumption propagation)**. If $\mathcal{H}_{ij}$ contains $i{:}A \xrightarrow{\supseteq} j{:}\ \alpha$ and $i{:}B_k \xrightarrow{\subseteq} j{:}\ \beta_k$ for $i \le k \le n$ ($n \ge 0$), then:

$$\mathcal{M} \vDash_d i{:}A \sqsubseteq \sqcup_{k=1}^{n} B_k \Rightarrow \mathcal{M} \vDash_d j{:}\ \alpha \sqsubseteq \cup_{k=1}^{n} \beta_k\ . \tag{19}$$

**Theorem 11 (Action into/onto concept subsumption propagation)**. If $\mathcal{H}_{ij}$ contains $i{:}\ \alpha \xrightarrow{\supseteq} j{:}A$ and $i{:}\ \beta \xrightarrow{\subseteq} j{:}H_k$ for $i \le k \le n$ ($n \ge 0$), then:

$$\mathcal{M} \vDash_d i{:}\ \alpha \sqsubseteq \beta \Rightarrow \mathcal{M} \vDash_d j{:}\ A \sqsubseteq \bigsqcup_{k=1}^{n} H_k \tag{20}$$

$$\mathcal{M} \vDash_d i{:}\ \alpha \sqsubseteq \beta \Rightarrow \mathcal{M} \vDash_d j{:}\ A \sqsubseteq \bigsqcap_{k=1}^{n} H_k\ . \tag{21}$$

where $\alpha$, and $\beta$ are actions and $A$ and $H_k$ ($1 \le k \le n$) are concepts.

The theorems 4-11 are important as they constitute the main reasoning step of the tableau algorithm proposed for D3L(X). Given the limited space available, in this article I will not delve into the details of the proofs of the above properties.

## V. CONCLUSIONS

The last decade of basic research in the area of Dynamic Description Logics DDL(X) has created a stable theory, efficient inference procedures, and has demonstrated a wide applicability of DDL(X) to dynamic knowledge representation and reasoning. Distributed Dynamic Description Logic D3L(X) is a natural generalization of the DDL(X) framework designed to formalize multiple ontologies interconnected by semantic mappings. State of the art languages D3L(X) for ontology mapping enable to express semantic relations between homogeneous components of different ontology-based information systems, namely they allow to map concepts into concepts, individuals into individuals, and actions into actions. In many real cases, however, we have to design an expressive mapping language to semantically understand resources from remote and heterogeneous systems. The approach proposed in this paper is to provide an extension of Distributed Dynamic Description Logics D3L(X), which is composed of a set of stand alone DDL(X) pairwise interrelated with each other via collection of homogeneous/heterogeneous bridge rules. Furthermore, we study the realizability, executability, and projection problems on D3L(X)-actions. It is demonstrated that the three primary reasoning tasks on actions can be reduced to the satisfiability problem on formulas.

The paper represents a work in progress. Thus it has many open issues for the future research directions. In recent years Shi et al. developed a tableau algorithm for deciding the satisfiability of DDL(X)-formulas. Furthermore, it is also a valuable and interesting work to develop a tableau-based distributed reasoning procedure for providing the capability of global reasoning in D3L(X) and decomposing large reasoning tasks to sub-tasks that could be concurrently processed by different local reasoning engines. Based on this algorithm, reasoning tasks on actions, such as the realizability problem, the executability problem and the projection problem, can all be effectively carried out.

## ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities (No.11QG13 and No.12ZP09) and the National Natural Science Foundation of China (No.61105052 and No.71101048).

## REFERENCES

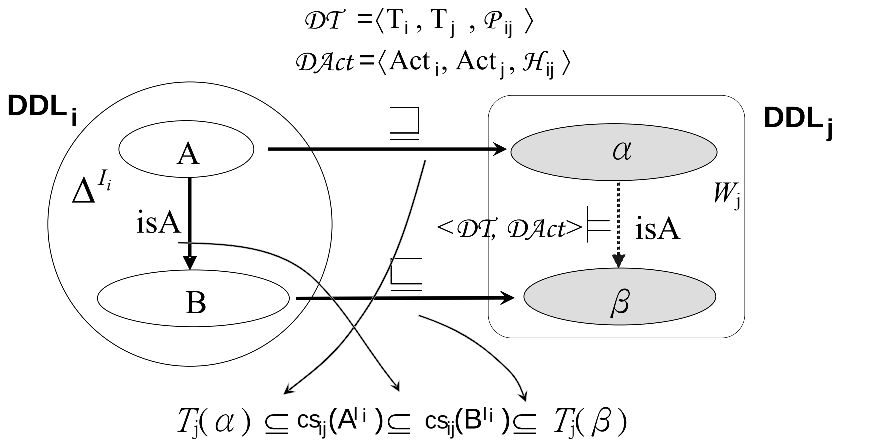


Figure 3. Graphical intuition of heterogeneous action subsumption propagation.

[1] Bonatti, P., Lutz, C., Wolter, F.: Description logics with circumscription. In: Proc of the 10th Int Conf on Principles of Knowledge Representation and Reasoning, pp. 400--410. AAAI Press, Menlo Park (2006).

[2] Jiang, Y., Wang, J., Tang, Y., Deng, P.: Semantics and reasoning of description logic μALCQO. Journal of Software, vol. 20, pp. 491--504. (2009)

[3] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2002).

[4] Horrocks, I.: DAML+OIL: A description logic for the semantic web. Bull of the IEEE Computer Society Technical Committee on Data Engineering, vol. 25, pp. 4--9. (2002)

[5] Horrocks, I., Patel-Schneider, P.F., Harmelen, F.V.: From SHIQ and RDF to OWL: the making of a web ontology language. Journal of Web Semantics, vol. 1, pp. 7--26. (2003)

[6] Ma Yue, Sui YueFei, Cao CunGen: The correspondence between the concepts in description logics for contexts and formal concept analysis. Science china-information sciences, vol. 55, no. 5, pp. 1106--1122. (2012)

[7] Bobillo Fernando, Straccia Umberto: Generalized fuzzy rough description logics. Information sciences, vol. 189, pp. 43--62. (2012)

[8] Shi, Z., Dong, M., Jiang, Y., Zhang, H.: A logical foundation for the semantic web. Science in China, Ser. F, vol. 48, pp. 161--178. (2005)

[9] Chang, L., Shi, Z., Gu, T., Zhao, L.: A Family of Dynamic Description Logics for Representing and Reasoning About Action. Journal of automated reasoning, vol. 49, no. 1, pp. 1--52. (2010)

[10] Wang, Z., Yang, K., Shi, Z.: Failure Diagnosis of Internetware Systems Using Dynamic Description Logic. J. Softw. China, vol. 21, pp. 248--260. (2010)

[11] Chen, L., Hu, H., Shi, Z.: Reasoning about Web Services with Local Closed World Assumption. In: Proc of Int Conf Web Intelligence, pp. 367--370. IEEE press, New York (2009)

[12] Jung, JJ.: Reusing ontology mappings for query routing in semantic peer-to-peer environment. Information Sciences, vol. 180, no. 17, pp.3248--3257. (2010)

[13] Borgida, A., Serafini, L.: Distributed description logics: assimilating information from peer sources. Journal of Data Semantics, vol. 2800, pp. 153--184. (2003)

[14] Kutz, O., Lutz, C., Wolter, F.: Epsilon-connections of abstract description systems. Artificial Intelligence, vol. 156, pp. 1--73. (2004)

[15] Baader, F., Lutz, C., Milicic, M., Sattler, U., Wolter, F.: Integrating description logics and action formalisms: first results. In Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'2005), Pittsburgh, PA, USA. (2005)

[16] Reiter, R.: Knowledge in action: logical foundations for specifying and implementing dynamical systems. MIT Press. (2001)


**Zhuxiao Wang**, born in Sichuan, China, in 1981, is a faculty member in the School of Control and Computer Engineering, State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, China. He got his Ph.D. degree from the Institute of Computing Technology, the Chinese Academy of Sciences in 2010. His current research interests include knowledge representation and reasoning, self-healing systems, distributed computing. He is a member of Chinese Society for Electrical Engineering. E-mail: wangzx@ncepu.edu.cn.

**Jing Guo**, born in Tianjin, China, in 1983, is an engineer in National Computer network Emergency Response technical Team/Coordination Center of China. She got her Ph.D. degree from the Tsinghua University in 2011. Her current research interests include distributed computing, information and network security. E-mail: guojing.research@gmail.com.

**Fei Chen**, born in Beijing, China, in 1978, and got the master degree of Engineering from the North China Electric Power University in 2003, majored in computer science and technology.

She is a lecturer in the Department of Computer Science of the School of Control and Computer Engineering in the North China Electric Power University, China. Her current research interests include data mining, decision support system. E-mail: chenfei@ncepu.edu.cn.

**Kehe Wu**, born in Jiangsu, China, in 1962, received his Ph.D. from North China Electric Power University. He is now the director of Beijing Engineering Research Center of Electric Information Technology and the Deputy Dean of Control and Computer Engineering School. His current research interests include electric power information security, massive data processing, and intelligent software technology.

He has worked in the electric power information systems arena for over 20 years. His experience spans the full systems life cycle. He is a coauthor of a paper entitled "Research and Design of Runtime Software Monitoring and Control Model" in the Proceedings of the 2nd International Conference on Software Technology and Engineering in Puerto Rico, United States of America, 2010. He holds several well recognized professional security related certifications. E-mail: wkh@ncepu.edu.cn.

**Peng Wang**, born in Shandong, China, in 1980, is a faculty member in Institute of Information Engineering, Chinese Academy of Sciences, China. He got his Ph.D. degree from the Institute of Computing Technology, the Chinese Academy of Sciences in 2011. His current research interests include cloud computing, massive data processing, programming models, knowledge representation and reasoning. E-mail: wangpeng@iie.ac.cn.