Specifying Policies Using UML Sequence Diagrams

An Evaluation Based on a Case Study

Bjørnar Solhaug,
Dag Elgesem and Ketil Stølen

2007 IEEE Workshop on Policies for Distributed Systems and Networks, Bologna, 13 June 2007
Motivation and Objective

- Policies are increasingly being used for the management of information systems
- The UML is currently the *de facto* standard for the modeling of information system
- Can we specify policies using the UML?

- The UML is not a policy specification language

- Objective: Evaluate the suitability of using the UML for policy specification
  - Specifically UML sequence diagrams
Sequence Diagrams

Positive trace
- Valid behavior
- !x, ?x

Negative traces
- Invalid behavior
- !x, ?x, !y, ?y

Message
Send event
Receive event

Lilfeline

sd example

A

B

x

?x

!x

?y

y

!y

neg

ref

sd example

sd E

A

B

z
Policy Concepts

- A policy is a set of rules governing the choices in the behavior of a system (M. Sloman, 1996)
Kripke Semantics for Deontic Logic

Acceptable worlds

Possible world (state)

\( p \)

\( P E p \)
State Triggered Permission

- The `ConfigureMachine` traces are positive when the time constraint and the OCL expression hold (permission).

\[
\{10pm < t < 4am\} \quad t
\]

States trigger

Addressee

Software engineers are permitted to configure TC when time is between 10pm and 4am.

\[
\text{states } p
\]

\[
\text{ref } \text{ConfigureMachine}
\]

\[
\text{states } q
\]

\[
\text{actor.oclIsTypeOf(SW engineer)}
\]
Kripke Semantics for Deontic Logic

$p \rightarrow P Eq$
Implied Prohibition

The ConfigureMachine traces are negative when the time constraint or the OCL expression does not hold (prohibition)

Software engineers are permitted to configure TC when time is between 10pm and 4am
Evaluation – Deontic Expressions

- Sequence diagrams can express permissions, obligations and prohibitions
- Fulfill the requirements to the semantics as defined in standard deontic logic
- Particularly, the following hold
  - $OBp \rightarrow PEp$ (SDL axiom)
  - $PEp \leftrightarrow \neg OB\neg p$ (SDL definition)
  - $PRp \leftrightarrow OB\neg p$ (SDL definition)
- But with state triggers, we additionally get
  - $p \rightarrow PEq$ implies $\neg p \rightarrow PRq$
  - $p \rightarrow PRq$ implies $\neg p \rightarrow PEq$
Evaluation

Composition using combined fragments

UML state

State trigger

Event trigger

Policy trigger

Policy rule

Policy

Addressee

Behavior

Deontic modality

Obligation
Permission
Prohibition

Lifeline

Interaction

No customized UML constructs
Future Work

- Based on the evaluation
- Introduce customized policy specification constructs
- With a semantics matching deontic logic

- The syntax and semantics should capture the theory of standard deontic logic