2.3 Integration of the two Levels: MMM-Logic

A specification of an object-oriented system is a pair ($\Sigma_{SYS}$, $\phi_{SYS}$, $\phi_{UML}$), where, $\phi_{SYS}$ is a formula describing the objects inherent to the application domain and $\phi_{UML}$ is the formula defining the meta-model semantics.

The formula $\phi_{SYS}$ is constructed over the extended language MMM and thus it can express at the same time model properties (e.g. behavioral properties of objects), meta-model properties (e.g. properties about the schema of the system) and properties relating both aspects.

2.3.1 Standard Models

A model of a MMM specification is standard if it satisfies the following properties:

- The carrier set interpreting the sort name Object is isomorphic to Nat.
- The following formula, defining the meaning of the event 'new', must be true in every state of the model:
  $\forall C.\text{Class} [\text{new}(C)] \land \text{lastOld} = \text{succ}(\text{lastOld}) \land \exists \text{lastOld} \land \text{my_class(lastOld)} = C$
- For all sort C such that C : Class, the carrier set of C in the model must contain exactly one element: the interpretation of the constant c, denoting the class itself
- Objects do not react to messages that they do not understand, formally
  $\forall o \in \text{Object} \land \text{message} \rightarrow (\text{name}(o) \land \text{operationNames(my_class(o)))} \rightarrow [o \bowtie] \text{false}$
2.3.2 Example of MMM-formulae

As an example, in this section we show the appropriateness of MMM-formulae for the specification of contracts [Helm and Holland 90]. A contract defines the behavioral composition of a set of communicating participants. Each contract specifies the following aspects: firstly, it identifies type obligations, where the participant must support certain external interface, and causal obligations, where the participant must perform an ordered sequence of actions and make certain conditions true in response to these messages. Through type obligations, contracts capture structural dependencies between objects (this information is present at the meta-model level). Through causal obligations, contracts capture behavioral dependencies between objects (this information is present at the model level). Secondly, the contract defines invariants that participants cooperate to maintain.

As the concrete example consider the contract in which a subject object, containing some data, and a collection of view objects, which represent the data graphically, cooperate so that at all times each view always reflects the current value of the subject. This contract is formally defined by the formula below:

$$(\exists s : \text{Class})(\exists v : \text{Class})\ (\text{name}(s) = \text{subject} \land \text{name}(v) = \text{view})$$

"type obligations"

$$(\forall s_1)(\forall v_1) ((\text{myClass}(s_1) = s \rightarrow [s_1, \text{attachView}(v_1)]) \land [s_1, \text{detachView}(v_1)] \land [s_1, \text{setValue}(n)] \land \text{Enabled}(s_1, \text{notify}) \land \text{Enabled}(v_1, \text{update}))$$

"causal obligations"

$$(\forall v_1)(\forall s_1) ((\text{myClass}(v_1) = v \rightarrow [v_1, \text{setSubject}(s_1)]) \land [v_1, \text{reflects}(v_1, v_1, \text{mySubject} = s_1)] \land (v_1, \text{update}) \land \text{ Enabled}(v_1, \text{update}))$$

"Invariants"

$$(\forall s_1)(\forall v_1) ((\text{myClass}(s_1) = s \rightarrow (v_1) (v_1, \text{views} \rightarrow \text{reflects}(v_1, v_1, \text{mySubject} = s_1))$$

Let specificSubject and specificView be two classes belonging to a model M. Let $\Phi$ be the formula defining the contract. Intuitively specificSubject and specificView conform the contract if and only if its operations and attributes satisfy both typing and causal obligations and invariants required by the formula defining the contract. That is to say, the formula $\Phi$ (under renaming of class names) is true in the model M (i.e. $M \models \Phi[\text{subject}/\text{specificSubject}, \text{view}/\text{specificView}]$).

2.4 Transparent use of the conceptual model

To gain acceptance of the proposed formal conceptual model by typical engineers, we are developing a semi-automatic transformation method. This transformation method defines a set of