INTEROPERABILITY COMPLIANCE VALIDATION OF HLA FEDERATIONS BASED ON COLOURED PETRI NETS

Stéphanie Combettes, Alexandre Nketsa
Gr. OCSD
LAAS-CNRS
7, avenue du Colonel Roche
31077 Toulouse cedex, France
E-mail :{scombett, alex}@laas.fr

Abstract : Validating complex systems is nowadays achieved by simulation of cooperating components. As complex systems have to support multidisciplinarity and multiformalism it induces these components may be different. Moreover cooperation between them means a data exchange. Exchanged data among various components may have different format and then a match is necessary. The latter is called interoperability, an essential notion. HLA OMT aim is to provide a template to document the exchanged data and their characteristics and so to promote interoperability. However verification of the document consistence becomes laborious if a huge number of data is exchanged. We propose a way to check a part of the consistence of OMT by using Coloured Petri Nets. We consider the parts consist of data compliance, completeness and structural conformity and we check only the first two ones. To do it, firstly OMT is transformed into a Coloured Petri Net and secondly verification may be achieved by its structure analysis.

INTRODUCTION

Nowadays different simulations have to be able to cooperate together to allow first their reuse or one of their components, and secondly simulation of large complex systems (Nance 1999). However matching between exchanged data is not trivial and requires an understanding among simulations. The latter is called interoperability. An architecture (High Level Architecture) has been developed to provide a common architecture and to facilitate interoperability among simulations (Dahmann et al. 1998 ; DMSO 1998). Moreover a template (Object Model Template) documents HLA relevant information about classes of simulation, federation objects, their attributes and interactions. The model objects are described in a SOM (Simulation Object Model) for a federate (an individual federation member) and in a FOM (Federation Object Model) for the federation (set of interacting federates). Practically this documentation is a set of tables detailing object classes, their attributes, interactions classes and their parameters, and complex data types (DMSO 1998).

This record is a necessary but not sufficient condition to enable interoperability. The problem of conformity and correspondence between data is not resolved yet. Despite this documentation, additional problems may appear such as omission, redundancy, non equality among types of data etc… These conflicts are gathered according to four categories: description, structural, syntaxical and semantical errors .

We have relied on HLA OMT to study interoperability problem as it's a precise documentation having a template. Our approach consists in testing completeness (description error) and data compliance (semantical error) among SOMs and FOMs. To do this we transform the table structure of OMT into a Coloured Petri Net. We analyse it to check eventual conflicts and finally validate it. We use Coloured Petri Nets because it's a formal tool enabling a mathematical analysis of net properties allowing to conclude about conflict existence.

Section 2 of the paper describes each kind of conflicts enumerated in the introduction. Section 3 gives an overview of Coloured Petri Net formalism and depicts the Coloured Petri Net construction from OMT tables. Section 4 presents an analysis of this method. Section 5 suggests our future research and section 6 gives conclusions.

CATEGORIES OF CONFLICTS

Interoperability consists in specifying an exchange mechanism which is transparent to the simulation type to be conducted. In others words it's "the ability of a model or simulation to provide services to and accept services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together" (D0DD 1994).

Leclerc in (Leclerc 2000) made a research in the interoperability domain of geographical information systems (GIS), and proposed to share and exchange information from GIS by exploiting the semantical aspect of spatial information (representation conflicts).

In a similar way, we have searched for conflicts that may occur within OMT specification (DMSO 1998). After they have been enumerated we have identified sets of error types. Thus we have grouped together comparable conflicts within same categories that are four of them.

Moreover, Horst in (Horst 1998) had deduced from OMT specification and HLA rules, some test procedures to be applied for validation and verification of OMT interoperability. We are allowed to notice that some of our errors may be deduced from the OMT Test Procedures proposed by Horst (Horst 1998), but she doesn't take into account all semantical conflicts.

In the following, the four error categories are described. We have refined categories found previously in (Combettes and Nketsa 2002).

Proceedings 14th European Simulation Symposium
A. Verbraeck, W. Krug, eds. (c) SCS Europe BVBA, 2002

Description errors

It corresponds to the well-documentation and definition of all object classes, their attributes, their interactions and parameters. It means errors such as omission, redundancy of object class attributes and characteristics, typing errors. Moreover, some additional constraints have to be checked. It concerns the characteristics of attributes. If datatype is defined by the user (not a base attribute type), then entry for the characteristics "Units", "Resolution", "Accuracy" and "Accuracy condition" shall be N/A (Horst 1998), for example. Same order constraints concern other tables.
This kind of conflicts appears within a FOM or a SOM and verification may guarantee that there will not be incoherencies in their description. Thus SOM (or FOM) **completeness** is checked but not global interoperability.

Structural conflicts

Structural conflict is the fact that an object in FOM and/or SOM doesn't have same data structure. It is concerned with data organisation. So it may be :
An aggregation conflict: in at least two specifications (SOMs and/or FOMs), attributes (or object classes) may be grouped together in another way according to various points of view.
A generalisation conflict: it may occur when a same object is differently defined in specifications.
A description conflict: it means that an object doesn't have same details. For example, the attribute of date object may be whether just a Year or year/month/day. We notice that in effect description conflicts are included in aggregation and generalisation conflicts.
Moreover, as multiple inheritance is forbidden (DMSO 1998 ; Horst 1998), each class has to be checked to trial this property.
Verifying that none structural conflict is in the tables means that **structural conformity** is validated.

Syntaxical conflicts

Syntaxical conflicts mean that entities may have different names but represent the same entity (synonyms), or on the contrary they may have identical writing name but different sense (homonyms and more precisely homographs). This may be caused by :
An object naming conflict: it means that in at least two specifications, a same object has different names (synonyms) or a same name stands for two objects (homonyms).
An attribute naming conflict: same attributes are named differently. For example an object representing a date may be composed of the attributes: Year/Month/Day whereas in another specification this same object may have: Y/M/D.
To detect this kind of conflicts is not really easy but may become important in syntax validation among specifications within a federation.

Semantical conflicts

Semantical conflicts are problems due to a faulty conformity in the characteristic (parameter) description of attributes. It may be caused by heterogeneous types. For example, Month attribute type of Date object (previously cited) may be whether an integer (from 01 to 12) or a string (from January to December). Likewise, unit incoherence may affect data value. If an attribute unit is metre and another one in centimetre. Difference between resolution, type, condition of accuracy or condition of update causes also semantical problems and by consequence a non-compliance between data. Here SOM (or FOM) **compliance** is checked.

We propose in the next section, a method based on the Coloured Petri Nets (CPN) to test completeness and data compliance.

COLOURED PETRI NETS

There exists a relationship between OMT tables which describe the dependences between classes and their own characteristics. These relations may be represented by means of Petri Nets. Among all derivations of Petri nets, the choice of High Level Petri Nets has become evident as tokens have to be individualised. Because tables don't take into account methods but only attributes (static part), colours are a better choice instead of objects.
Thus our choice turns towards Coloured Petri Nets (CPN) in order to verify a part of interoperability. We give first a brief overview of CPN formalism, next we detail the coloured token structure and describe the CPN construction. Finally a short example is exposed to illustrate our approach.

Formalism of CPN

CPN is an abstraction of Ordinary Petri Nets where tokens are not individually identifiable. Ordinary Petri Nets may be mathematically analysis (Jensen 1990). It gives properties first about the net behaviour (liveness, boundedness etc…) and secondly about the net structure, independently of the marking (transition or place invariants). CPN may be seen as a folding of several Ordinary Petri Nets (Jensen 1996). Folding means that the net is made up of disjoint subnets with nearly identical structure. The subnets are grouped and each token member of a subnet is converted into coloured token. The global net becomes then more compact and as a consequence it highlights similarities and differences between Ordinary Petri Nets. In CPN tokens may be individually identified according to their colour. Moreover they may have an attached data value (complex type). Each place (transition) is connected to a transition (place) by an arc labelled by a function of colour, and each place contains a specific set of colours (tokens). The place marking is the number of colours in this place. Each transition may be mapped by a guard function, i.e. a boolean expression whose purpose is to define an additional constraint that must be fulfilled before transition is enabled.

Coloured token

Coloured token may have complex data structure and so modelling of object classes and their attributes is facilitated. An (object) class is composed of a attribute set. By consequence we consider a class corresponds to a place and attributes to tokens. A class may have several attributes and in parallel a place may contain several tokens.

Coloured tokens are identifiable and may be a colour set. Now we give details about the colour structure.

Firstly to model each attribute of object class a colour is defined that is called colour ATTRIBUTE. It is composed of a (sub) colour set representing its own characteristics. So colour is a complex data type composed of type, unit, cardinality etc…So it is a composition of colours. As a result we have:

Colour ATTRIBUTE: record of:
Datatype : Colour TYPE
Cardinality : Colour SIZE
Units : Colour UNIT
Resolution : Colour RESOLUTION
Accuracy : Colour ACCURACY
Acc.Condition : Colour A-CONDITION
Update Type : Colour U-TYPE
Upd.Condition :Colour U-CONDITION;

Secondly we define TYPE colour to model complex or enumerated datatypes. Thus TYPE colour is a composition of all types that may characterise attributes i.e. a base data type, an enumerated data type or a complex data type. So a complex data type colour called COMPLEX-DATATYPE is specified. In a similar way we define ENUMERATED-TYPE colour. Each of two last colours is composed of its own attribute list: CDT-FIELD and ENUMERATOR ones. (CDT stands for Complex Datatype). Thus we have:

Colour CDT-FIELD: record of
Datatype : Colour TYPE
Cardinality : Colour SIZE
Units : Colour UNIT
Resolution : Colour RESOLUTION
Accuracy : Colour ACCURACY
Acc.Condition :Colour A-CONDITION;
Colour complex-datatype: List cdt-field;
Colour ENUMERATOR: record of
Identifier : integer;
Colour enumerated-type : List enumerator;

Whatever the class types (object or interaction class), colour structure is similar. Only colours will be different (PARAMETER colours).

Coloured Petri Net construction

The Coloured Petri Net used to check consistency is a hierarchical one i.e. a CPN combining a number of smaller nets. These smaller nets are non-hierarchical but coloured nets (Jensen 1996) and called pages. The hierarchical CPN describes verification of consistency. Pages describe the hierarchical structure of object classes. In others words CPN view is set in a higher abstraction level than pages. The verification of all the pages is checked in parallel. In the following we focus on Coloured Petri subnets (pages).

The subnets are constructed from OMT tables. Object class structure table informs about the hierarchy of object classes. Thus, as multiple inheritance is not allowed, each hierarchy is independent. In that case a coloured Petri subnet may be realised to represent a complete hierarchy. In this subnet each class of hierarchy is represented by a place. Mother-to-child relationships are modelled by arcs and transitions. Each class contains as coloured tokens as owned attributes. (An attribute is a coloured token). An expression function maps each arc and a guard function each transition. These functions represent the way transitions may be enabled, so fired.

A subnet is constructed similarly from the OMT. This is a reference subnet. If OMT tables are consistent, we deduce that subnets and reference subnets are identical through their structure and colour set of equivalent places. To check consistency, a subnet and the referent one are in a way merged by their transitions. Hence in the resulting subnet, two places come before a transition: the referent place and the one of SOM (FOM) subnet. Tokens within these two places are supposed to be equal. Therefore, the guard function added on this transition checks equality between tokens of places.

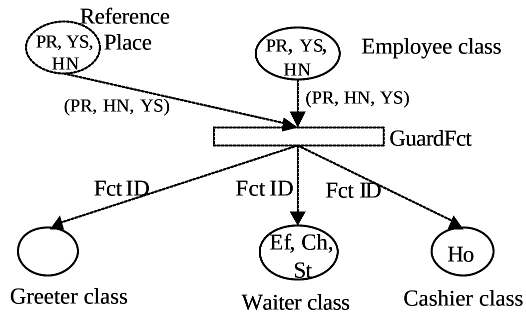
Note that as inheritance exists, attributes of mother class are too ones of child classes. If a place has more than one previous transition, it represents multiple inheritance, which is forbidden. We may represent SOM and FOM tables with this method.

Illustrative example

This example describes only one class hierarchy that is a part of a SOM. First table documents each class with its attributes and characteristics. Second one shows hierarchical relationship between them, and finally a CPN subnet (page) is constructed from these two tables.

Class	Attribute	Data-type	Cardinality	Units	Reso-lution	Accu-racy	Accuracy Condition	Update Type	Update Condition
Employee	Pay-Rate PR	float	1	Cents /hour	1	Perfect	Always	Conditional	Merit increases
	Years-Service YS	Short	1	Years	1	Perfect	Always	Periodic	1/year, on anniversary
	Home Number HN	String	1	N/A	N/A	Perfect	Always	Conditional	Employee Request
Waiter	Efficiency Ef	Short	1	N/A	1	Perfect	Always	Periodic	Perf review
	Cheerfulness Ch	short	1	N/A	1	Perfect	Always	Periodic	Perf review
	State St	Waiter tasks	1	N/A	N/A	N/A	N/A	Conditional	Work flow
Cashier	Honesty	short	1	N/A	1	Perfect	Always	Conditional	Bill errors

Class	Sub-class
Employee (S)	Greeter (PS)
	Waiter (PS)
	Cashier (PS)



GuardFct = 2 PR & 2 YS & 2 HN

Fct ID = function Identity

Each place represents a class and within place are set its colours, i.e. attributes. The reference place contains attributes with right characteristics because they are extracted from the OMT tables. By means of the guard function, if tokens in two places are equal, then transition is enabled and then fired. Thus, description or semantical errors may be checked. This step is done again for the other places with attributes. We don't go on this example because of lack of place.

ANALYSIS

One of the main reasons of choosing CPN is good properties that it provides. *Good net properties* derive from structural and behavioural analysis. It implies a preliminary study of marking and may indicate liveness, boundedness or re-initialisability of net. We precise that colour sets are taken into account in the property study of CNP.

The net *liveness* tells us that it is possible, for each reachable marking M′ to find an occurrence sequence starting in M' and containing an element from X (X being a set of binding elements). Its formal definition is :

Let a marking $M \in \mathbf{M}$ and a set of binding elements $X \subseteq BE$ be given :

X is **live** iff there is no reachable marking in which it is dead i.e. iff: $\forall M' \in [M_0\rangle\ \exists M'' \in [M'\rangle\ \exists x \in X : M''[x\rangle$.

It guarantees that firstly none deadlock may be caused by the net structure (depending too of the net marking) and secondly absence of dead parts (never reached).

The net *re-initialisability* tells us that it is possible for each reachable marking M′ to find an occurrence sequence starting in M′ and reaching again the initial marking M_0. Its formal definition is :

Let N a net, a marking $M \in \mathbf{M}$ and a set of binding elements $X \subseteq BE$ be given:

N is **re-initialisable** iff : $\forall M' \in [M_0\ \exists x \in X : M''\ [x\rangle M_0$.

A *non-re-initialisable* net prevents from analysing liveness and boundedness properties or means that a deadlock occurred.

We clarify that a re-initialisable net doesn't imply its liveness, and a live net doesn't imply it's re-initialisable.

Transformation from tables into CPN is linear. Thus to allow property studies, we make the hierarchical net to be re-initialisable. In order to this, some places and transitions are added which loop on the initial place. Conditions on additional arcs and transitions are chosen to be always satisfied. The liveness of a net implies the liveness of its subnets. Therefore if hierarchical net is live, all the pages (subnets) are too live. It means that consistency is validated.

However, if hierarchical net is not live, then one or several subnets are not live. Liveness of the net may check completeness (none description error) and compliance (none semantical error). It detects eventual deadlocks, that means at least one transition could not have be fired. Thus, as transition compares tokens of SOM (or FOM) place with the ones of the OMT (reference) place, then colours of characteristics are not identical, or a colour is missing.

The boundedness is here not useful because it doesn't help on the consistency verification and validation.

The study of net properties established from tables allows to check completeness and data compliance in the SOM or FOM. Moulding used in (Moulding and Newton 1998) the Relational Algebra to express the OMT Test Procedures (Horst 1998) as he considers the individual tables of SOMs and FOMs as relations. We have adopted the same point of view concerning relations between tables, so that they are easily represented with CPN. The OMT Test procedures are in our method represented in several ways. First the inheritance constraint is expressed through the net structure and secondly conditions on objects classes are expressed through functions on arcs and transitions, and with the colours. Our method allows a mathematical study of the net, which directly verifies completeness and data compliance of SOMs and FOMs.

Semantical compliance (heterogeneous types for example) is here checked, contrary to Moulding approach. However, structural compatibility is not yet verified.

This method has the other main advantage that translation may be done automatically. That means there is no need human interpretation, hence new source of error is avoided. Structure of tokens is similar for all object classes and interactions. Moreover hierarchical CPN use makes easier after inserting a new federate, verification of its SOM. It only requires an additional page, added in parallel. We remind you that each object class hierarchy are each other independent.

PERSPECTIVES

Now our work studies how to check structural conformity. We will try to demonstrate that formal analysis, exploring place or transition invariants may help in structural conformity checking. An invariant is defined as an equation which is satisfied for all reachable markings. We wish to establish an invariant for each class hierarchy in the OMT tables and compare it with the corresponding one of SOM or FOM tables. Different invariants may suppose a different net structure and perhaps a structural conflict.

CONCLUSION

We proposed here a new method to verify and validate a part of interoperability. CPN are used and its structure is analysed in order to validate completeness and compliance. A short example illustrates this method. We are working currently on the conformity validation.

References

Combettes S. and A. Nketsa. 2002. "HLA interoperability analysis based coloured Petri nets" *16th European Simulation Multiconference (ESM'02)* (Darmstadt, Germany, June 3-5).

Dahmann J.S.; F. Kuhl; and R. Weatherly. 1998. "Standards for simulation: As simple as possible but not the simpler the High Level Architecture for simulation" *Simulation Vol 71 ?6 pp 378-387*.

DMSO. 1998. "High Level Architecture Object Model Template Specification." *US Department of Defence*.

DoD Directive 5000.59. 1994. "DoD Modeling and Simulation (M&S) Management" Enclosure 2, p.14. January 1994.

Horst M. 1998. "Test Procedures for High Level Architecture Object Model Template" *Georgia Tech Research Institute, Georgia Institute of technology. Version 1.3,* 6 April .

Jensen K.1990. "Coloured Petri Nets : A High Level Language for System Design and Analysis." *K. Jensen and G. Rozenberg Eds, Springer-Verlag*.

Jensen K. 1996. "Coloured Petri Nets : Basic concepts, Analysis Methods and Practical Use, Volume 1." *Springer-Verlag, second Edition*.

Leclerc E. 2000. "Interopérabilité sémantique des systèmes d'information géographique : une approche basée sur la médiation de contexte." *PhD*. Université de Bourgogne, France.

Moulding M.R and A.R. Newton. 1998. "Automated Consistency Checking of HLA Specifications to Support the Verification of Synthetic Environments." *SISO Simulation Interoperability Workshop,* (spring 1998. Orlando, Florida).

Nance R.E. 1999. "Distributed simulation with federated models : expectations realizations and limitations" *Proceedings of the 1999 Winter Simulation Conference*. (Phoenix Arizona, December 3-8).