The ISDA model for integrating the Static and Dynamic Aspects of Information Systems

Thanh Thoa Pham Thi

University of Geneva, CUI, 24 rue General Dufour,
CH-1211 Geneva 4, Switzerland
thanh.phamthi@cui.unige.ch

1 Context

Current methods for information systems modelling often focus on the different aspects of information systems (IS). The methods, for example, Merise [10], IDA[3] and UML[4] are interested in the Static, Dynamic, Organizational, and Architecture aspects. Each aspect may dispose one or several models, which have different formalism and interpretation.

This paper concentrates on the Static and Dynamic aspects, which are the most fundamental aspects at the Informational level. At this level, the Class model represents the Static aspect; meanwhile the Dynamic aspect is represented by the State-chart or Object life cycle model.

These two types of models are indeed deeply interrelated. However, due to the differences between the two models, the verification of consistency of these models becomes an interesting challenge, especially in the case of the evolution of the models.

In the literature, many works have concerned with this challenge that can be grouped into two categories: verifying the consistency in each model [3][5] and verifying the consistency between different models [8][9]. Most of them used formal methods and transformed the specifications from source models, which have different formalisms, into a common formalism. In general, the Z formal language and First-order logic are used as common formalisms. When the specifications are transformed into the common formalism, they are verified the consistency within /between the two aspects. The method of [8] is different. The common formalism is a graph consisting of nodes and edges with multiplicities, and then a system of linear inequalities is derived from the graph for solving the consistency verification.

In our standpoint, the verification task can be avoided if we take into account this challenge at the modelling phase. It is the reason why we propose in this paper a unique model: **the ISDA model** that can be used to specify the Static aspect, Dynamic aspect as well as interrelations between them. Furthermore, the formalism of this model must not be more complicated than the formalisms of other methods so that IS professionals from different background can easily work with it. Besides, the proposed model should be easily implemented.

This model can be applied in the recently research and industrial domains such as the interoperability of information systems, components for enterprise systems, and evolution of information systems.

2 The ISDA model

As mentioned above, the main purpose of our work is to propose such a unique model for representing both the Static and Dynamic aspects of IS. This model is called the *Integrating Static and Dynamic Aspects model* (in short the ISDA model).
The concepts in our model are inspired from the object-oriented paradigm with some improvements to support more integration and evolution. In fact, this makes the differences between our model with other methods and therefore expresses the advantages of our approach.

This section presents the outline concepts of the ISDA model. Firstly, we present concepts of the Static aspect. Secondly, we continue with the concepts of the Dynamic aspect. Finally, we show how to present interrelations between the two aspects in our model.

2.1 Static aspect

In our approach, the *Class model*, which represents the Static aspect of IS, describes the structure of classes and relationships between them.
There is only one relationship type in the Class model in our approach: the ***existential dependency*** relationship, it helps supporting the evolution of the specification [6][7].
There is an existential dependency relationship from the C1 class to the C2 class if, for every object in the C1 class, the existence of this object depends on the existence of one and only one object in the C2 class. This dependency is permanent in the life of objects in the C1 class.

On the other hand, the well-known concept: *Specialisation* or *Is-A* in current modelling methods have often implemented by the inheritance mechanism. Consequently, once an object is specified in a class, it stays permanent in this class.
In our approach, we propose the *Dynamic specialisation* (DS), which is a particular case of existential dependency. The main objective of the DS is to authorize an object to change the class, to which it belonged to, in its life cycle. Consequently, the dynamic specialisation is more flexible than the specialisation concepts in other approaches:

- Objects can move between classes in the Dynamic Specialisation graph (DS graph). A DS graph is defined as a set of classes linking by dynamic specialisations;
- When objects move from a class (called the Super-class) to a sub-class of this class (called the Sub-class), it may be *active* or *inactive* in the Super-

class. When an object is inactive then one cannot execute any method on it, but one can consult it;

- As mentioned above, an object may stay in several classes in a DS graph. So it depends on the fact that in which class, one desires to access the object then the object shows/ hides its different informations. The concept ***Access-View*** is defined to describe this situation. An Access-View of a class consists of all properties defined on the C class and *certain* informations (attributes/ methods) of super-classes of C. There is no inheritance in the dynamic specialization [2].
- DS is not a reflexive relationship, i.e. if C1 is a super-class of a C2 class then the C2 class cannot be a super-class of the C1 class. In some situations of DS, an object may move from a sub-class to a super-class; it requires keeping the trace of the object. The concept ***Loop*** is defined for specifying this situation.

Concerning the formalism, the Class model in our approach is probably simpler than other approaches, because it uses a minimum set of concepts. Thanks to DS, the ISDA model can express more remarkable situations in which objects can change its class, can hold its traces, and can return to be an object of its super-class.

2.2 Dynamic aspect

The *Object Life Cycle* (OLC) represents the Dynamic aspect in the ISDA model. An OLC is presented by a bipartite graph, which is called *Node-Star Net* (NSN). A node corresponds to a *state*, and a star corresponds to a *transaction.* A transaction changes states of objects; it may be a function, a processus of systems or even a method of class.

In fact, there are some differences compared with OLC or state-chart in other methods [4][11]:

- We can indicate that an object holds former states when it changes from one state to other state. This primitive is not supported by other methods;
- Designer can define pre-conditions and post-conditions on transactions, meanwhile in the Petri net, there are only the conventions such as AND, XOR on logic connectors.

Concerning the formalism, the Node-Star Net is as complex as the Petri net, however it is able to express more situations above.

2.3 Integration of two aspects

We specify the correspondences of an object life cycle of a class to a DS graph with the respect to the order of states' change. Each state of an object life cycle of a *C* class corresponds to a class in a DS graph, which has *C* as its root class. If *S1* is a source state and *S2* is a destination state of S1 transformed by the transaction *T*, then the *C1* class (corresponding to *S1*) is a super-class of *C2* (corresponding to S2).

By this way, the dynamic specialisation concepts is used to present interrelations between the Static aspect and the Dynamic aspect. So this concept is a corner-stone of the integration of the Static and Dynamic aspects.

3 Conclusion and future works

We have presented the concepts of the ISDA model that aims at supporting the consistency between the Static and Dynamic aspects of information systems at the Informational level. This model uses the dynamic specialization to express the interrelations between these two aspects.

We also consider that the Organizational aspect is a very important aspect of information systems. Therefore, in the future works, we will take into account the consistency of this aspect with two Static and Dynamic aspects.

At the time being, we are developing an environment that allows specifying the specifications of the ISDA model and then implementing them automatically. Accordingly, we are applying the ISDA model in several application domains such as the interoperability of information systems, components for enterprise systems, and workflow applications.

References

1. Al-Jadir L., Evolution-Oriented Database Systems, Thèse Faculté de SES Université de Genève, 1997.
2. Al-Jadir L., Léonard M., If we refuse the Inheritance…, Proc. Int. Conf. on DEXA, Firenze, 1999.
3. Bodart F. et Pigneur Y., Conception assistée des systèmes d'information – Méthode - Modèles- Outils, Masson Paris 1989.
4. G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley 2000..
5. Formica A., Frank H., Consistency of the static and dynamic components of object-oriented specifications, Data & Knowledge Engineering, Volume 40, Issue 2, February 2002.
6. Léonard M, M7 : une approche évolutive des Systèmes d'Information, Inforsid à La Garde, juin 1999.
7. Léonard M, IS engineering getting out of classical system engineering, ICEIS, Angers France 2003.
8. Pascal Fradet, Daniel Le Métayer and Michaël Périn, Consistency checking for multiple view software architecture, ESEC/ FSE'99 volume 1687 of Lecture Notes in Computer Science, ACM Press.
9. Holger Rasch, Heike Wehrheim, Consistency between UML classes and Associated State Machines, workshop "Consistency problem in UML based software development", the 5th IC on The Unified modeling language, the language and its application, October, Dresden, Germany, 2002.
10. Hubert Tardieu, Dominique Nancy et Daniel Pascot. Conception d'un système d'information : construction de la base de données. Editions d'Organisation, 1979.
11. Nanci Dominique, Espinasse Bernard, Ingénierie des systèmes d'information : Merise, Deuxième génération, Vuibert Paris 2001.