



Toward a Theory of Integrated Modelling

B. Denis, J.J. Lesage, G. Timon

► To cite this version:

B. Denis, J.J. Lesage, G. Timon. Toward a Theory of Integrated Modelling. IFAC Workshop on CIM in Process and Manufacturing Industries, Nov 1992, Espoo, Finland. pp.35-40, <10.1016/S1474-6670(17)49363-3>. <hal-03740102>

HAL Id: hal-03740102

<https://hal.science/hal-03740102v1>

Submitted on 28 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

TOWARD A THEORY OF INTEGRATED MODELLING

B. DENIS, J.J. LESAGE and G. TIMON

*Laboratoire Universitaire de Recherche en Production Automatisée, ENS Cachan,
61 avenue du Président Wilson, F-94235 Cachan Cedex, France*

Abstract. For many years, complex systems designers have been able to use modelling techniques which are steady and proven. Today, most of these modelling techniques are used in «integrated methods». These methods provide analysts with a framework to design increased-complexity systems. In this paper, we propose an approach which makes integrated-methods formal by modelling them. Two examples are given to illustrate these concepts.

Key Words. Models; Modelling; Computer-aided system design; Design integrated methods; Metamodelling

1. INTRODUCTION

Today, the design of complex systems requires classic methods which include several modelling techniques. These modelling techniques allow designers to scan the abstraction cycle and the live cycle of a project as described by Kiefer *et al.* (1992). However, even if the modelling techniques used are often the same, (S.A. or IDEF0 for functional analysis; Petri nets, GRAFCET or Finite State Machines for dynamic modelling; NIAM or Entity Relationship models for data analysis;...), the design method which integrates them is often specific and evolutionary. A method is in fact a reflection of the know-how of a firm and progresses with its experience, even if the kernel of each method is more often than not inspired by the same “best-sellers” (Yourdon, SA-RT, SADT,...).

The consequence of this state of affairs is that the methodological tools marketed, however efficient, rarely satisfy the designers. The main difficulties encountered in these computer environments relate to the methods themselves which are often too distant from the firm’s specific methods and generally not evolutionary enough or adjustable to user requirements. They also relate to the “conversions of models” implied by the use of several modelling techniques.

One of the main reasons why the methodological aspects are at present not taken sufficiently into account by computer environments is that the fundamental research undertaken in the field of design methods proper is inadequate. That is why the aim of this research is to propose a technique of formal modelling of the integrated methods mentioned above.

In order to do this, the outline of this paper is as follows: after a presentation of the objectives and the interest of this research, there will be an explanation of the concept of *metamodelling*. The modelling of the integration of several modelling techniques within the same method will then be explained using simple examples.

2. THE AIMS OF THIS RESEARCH

A method is often defined as «a technique of resolving problems which is characterized by a set of clearly defined rules which lead to a correct solution» or more generally as a «set of rules and procedures to apply in dealing with a problem».

In this context, it can be postulated that an *integrated method* is a method in which the means of transforming one model into another is also explained.

To define a design method, is consequently to define - and to express in a given language - a management and development technique of a project. At present, methods definition is limited to a textual description of procedures and rules applied by the analyst. As for the modelling techniques used in a method, their description is often related to their syntax and not to their semantics as explained in a recent paper by Lhoste *et al.* (1992). It is evident that the textual expression of design methods and the syntactical expression of modelling techniques alone are not formal enough to be unambiguous and easily automated. Consequently, this research relates to the modelling of design methods.

The aim of the most recent research devoted to complex systems design, is to put the finishing touches to: sectorial design methods (Dembélé *et al.*, 1992), integration procedures between different modelling techniques (Kiefer *et al.* 1992; Roboam *et al.* 1989) or reference models (Böhms *et al.* 1993 or Vallespir *et al.* 1991). It will be shown that all these design methods, whether integrated or not, can be formally and exactly modelled.

The advantages of formal modelling are many. As regards design methods modelling, the essential advantages are listed below:

- the use of a common method by all of the parties involved in a project must be exactly the same, there can be no margin for «free interpretation» or for «astuteness». Everyone knows that, in opposition to textual descriptions, formal models do not admit of free interpretation. To model a method is thus to contribute to the increased *reliability* and *quality* of the method;
- whatever it may be, a method cannot be fixed once and for all. It must evolve at the same time as the firm and must take into account its structural and organizational modifications, and the evolution of the state of the art of project development. This ground work is facilitated by the existence of models. To model a method, is thus to contribute to the increased *evolutionarity* and *perenniality* of the method;
- the complex systems design cannot be realized without considerable assistance from a software environment. Automating a procedure described by a formal model is easier than automating a procedure described textually. To model a method, is thus to contribute to the increased *degree of automation* of the method.

The essential concept upon which this approach is built is that of *metamodelling*. The essential aspects of this approach will be now described.

3. A SHORT DISCUSSION OF METHODS

The kernel of each design method is composed of modelling techniques. The specific contribution of each method generally consists:

- in guiding the analyst in producing models (the notion of «directions for use» of which a modelling technique is generally devoid),
- in guiding the analyst in translating models from one modelling technique to another.

As regards the concept of assistance in the construction of models, the method proposed by D.A. Marca

et al. (1989) can be quoted. This method consists in building an activity/data matrix which enables SADT actigrams to be produced.

As regards the techniques of model conversion, three approaches can be distinguished:

- an *approach through translation* which consists in applying basic translation rules from a model (expressed in terms of modelling technique A) to another equivalent model (expressed in terms of modelling technique B) as proposed by Brode (1988);
- an *approach through transformation* which looks to project a given model through reference models to attribute it with a specific semantics. This is the approach chosen, for instance by A. Nielsen and H. Holm (1991);
- an *approach through association* which privileges the notion of different views expressing various aspects of the same modelled system. For example P. Ward (1986) retains the data flow diagram as described by T. DeMarco for functional analysis and finite state machines to represent the dynamics of the system.

4. THE APPROACH

4.1. Metamodelling

To model a design method is then:

- to model each of the modelling techniques retained (with the aim of expressing the syntax of the models),
- to model the production techniques of models (with the aim of expressing the semantics associated with the models),
- to model the techniques in order to transform one model into another (the integration aspect of the method).

The term chosen for the global model is *metamodel*. This global model then becomes the model of a modelling technique or of a design method.

4.2. Choice of the metamodelling technique

The formalism chosen to model design methods is that of the Entity-Relationship model, for the reasons listed below:

- it is well suited to representing the semantics of modelling techniques and not only their syntax,

- it allows one to represent at once simply and formally the procedures for converting models (often expressed by relationships between entities of the models),
- it conveys an «information system» point of view of the method considered. This point of view is, in the manner of CIM systems, a predominant aspect of the integration.

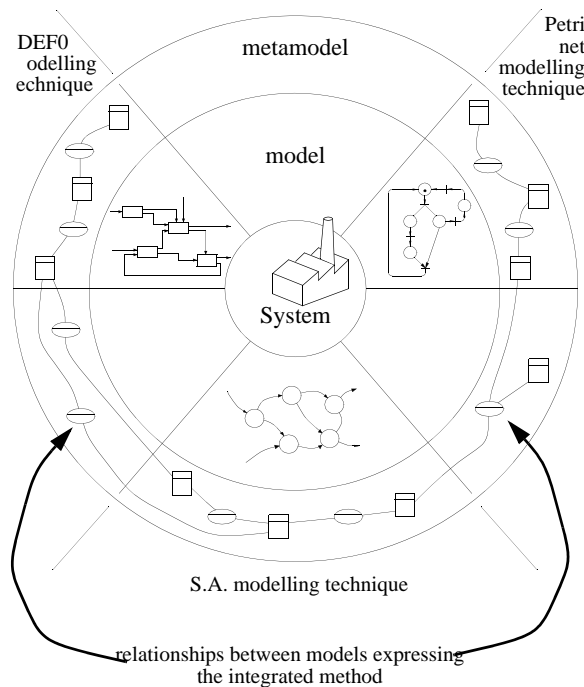


Fig. 1. Principle of the approach

5. EXAMPLES

To illustrate the relevance of metamodeling to the different approaches for integration, two examples will now be given. The first picks up the idea of the approach *through association* (the example relative to SA-RT) and the second describes the approach *through translation* applied to transform an IDEF0 diagram into a colored Petri net.

It should be stated with regard to these two examples, that the application of the approach proposed is bounded to model integrated methods quoted in the references.

5.1. Example relating to SA-RT

The first example has been constructed on the basis of SA-RT method which grew up around real-time embedded systems. The book written by Hatley *et al.* (1988) has been chosen arbitrarily as a reference to produce the following metamodel.

There are two main parts to the SA-RT method: «the requirement model» and «the architecture model». More precisely, this example concerns both the utilization of finite state machines and data and control flow charts in «the requirement Model». Integration links shown in Fig. 2 are expressed in an informal way in the following sentences from the reference manual: «By studying the events and actions on the State Transition Diagram (STD) as well as the control signals flowing into and out of the bar symbols on the Control Flow Diagram (CFD), you will see that the inputs to the bar symbols are the STD events, and that the output from them are the STD actions.»

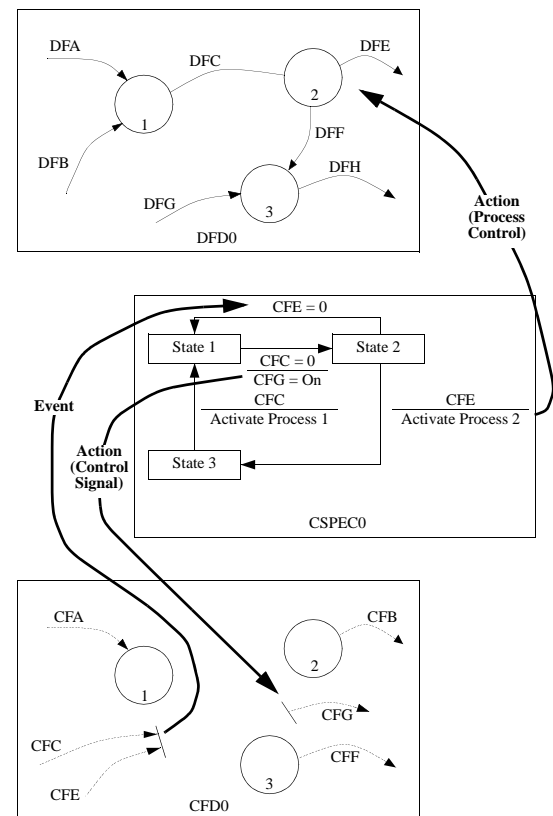


Fig. 2. Informal expression of the integration between data and control flow diagrams and state transition diagram by Hatley *et al.* (1988)

The different stages of the approach are:

- to model the different modelling techniques used (state transition diagram, data flow diagram, control flow diagram),
- to model the associations between these different modelling techniques,
- to integrate the three associated metamodels and to provide a unique metamodel of the integrated method shown on Fig. 3 (the entity «process» has been duplicated in this Figure to make it easier to understand the model).

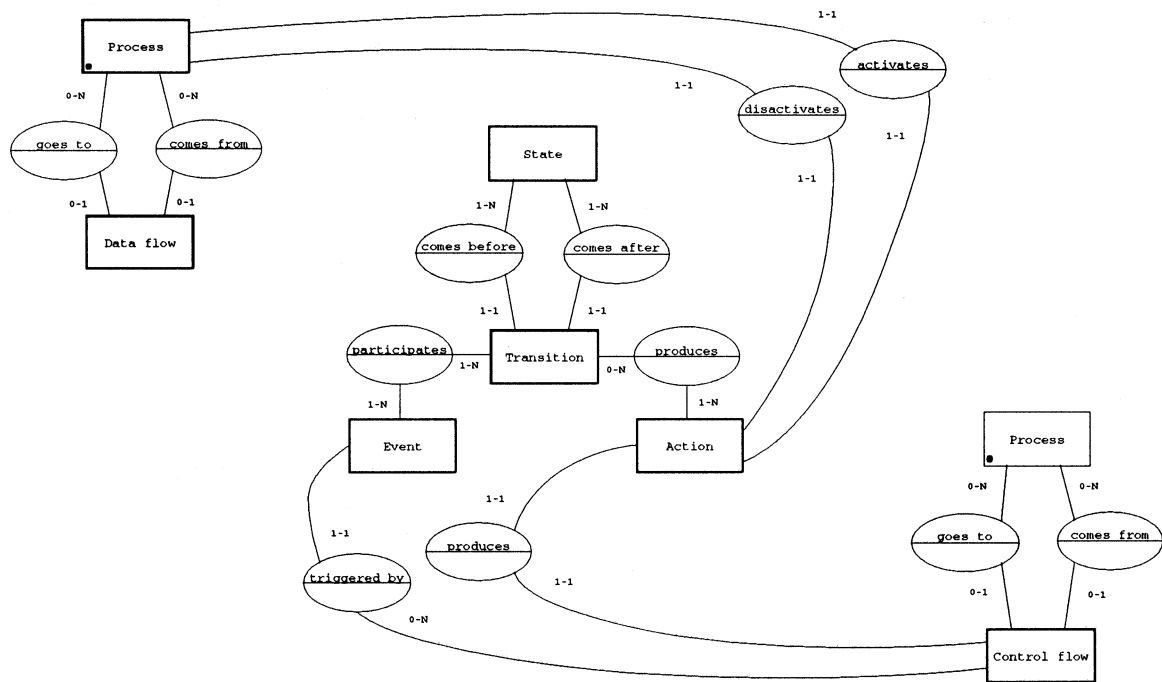


Fig. 3. Metamodel of the integration between data and control flow diagrams and state transition diagram according to its description in Hatley *et al.* (1988)

5.2. Example relating to IDEF0 and colored Petri net

The aim of this second example is to illustrate the approach *through translation*. It concerns, in this case, the translation of an IDEF0 diagram into a colored Petri net (CPN).

The conversion is accomplished with the aid of the translation rules proposed by Meta Software Corporation (Brode 1988). Further information about IDEF0 is available in Ross *et al.* (1979) and about Petri nets in T. Murata (1989).

Figure 4 illustrates the principle of translation. For example, an IDEF Activity becomes a CPN Transition (rule 1) or an Arc between two IDEF Activities is translated into an Arc (Transition, Place), a Place and another Arc (Place, Transition) in the corresponding colored Petri net (rule 5.b).

The statement of this example gives rise to remarks:

- to show, in the partial metamodel put forward (Fig 5), the link between translation rules and integration relationships, the label attributed to the latter is the number of the corresponding rule as described in the reference;
- the topology chosen allows to regroup the integration relationships between the models of the two

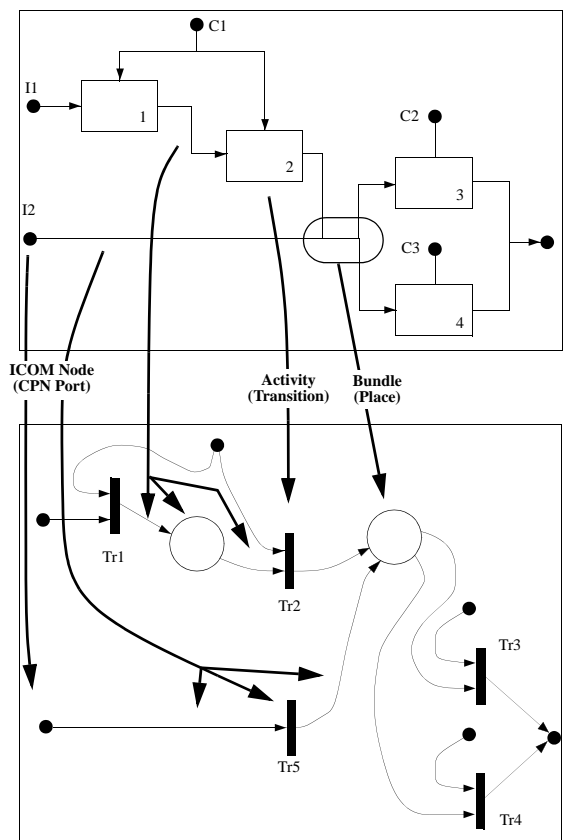
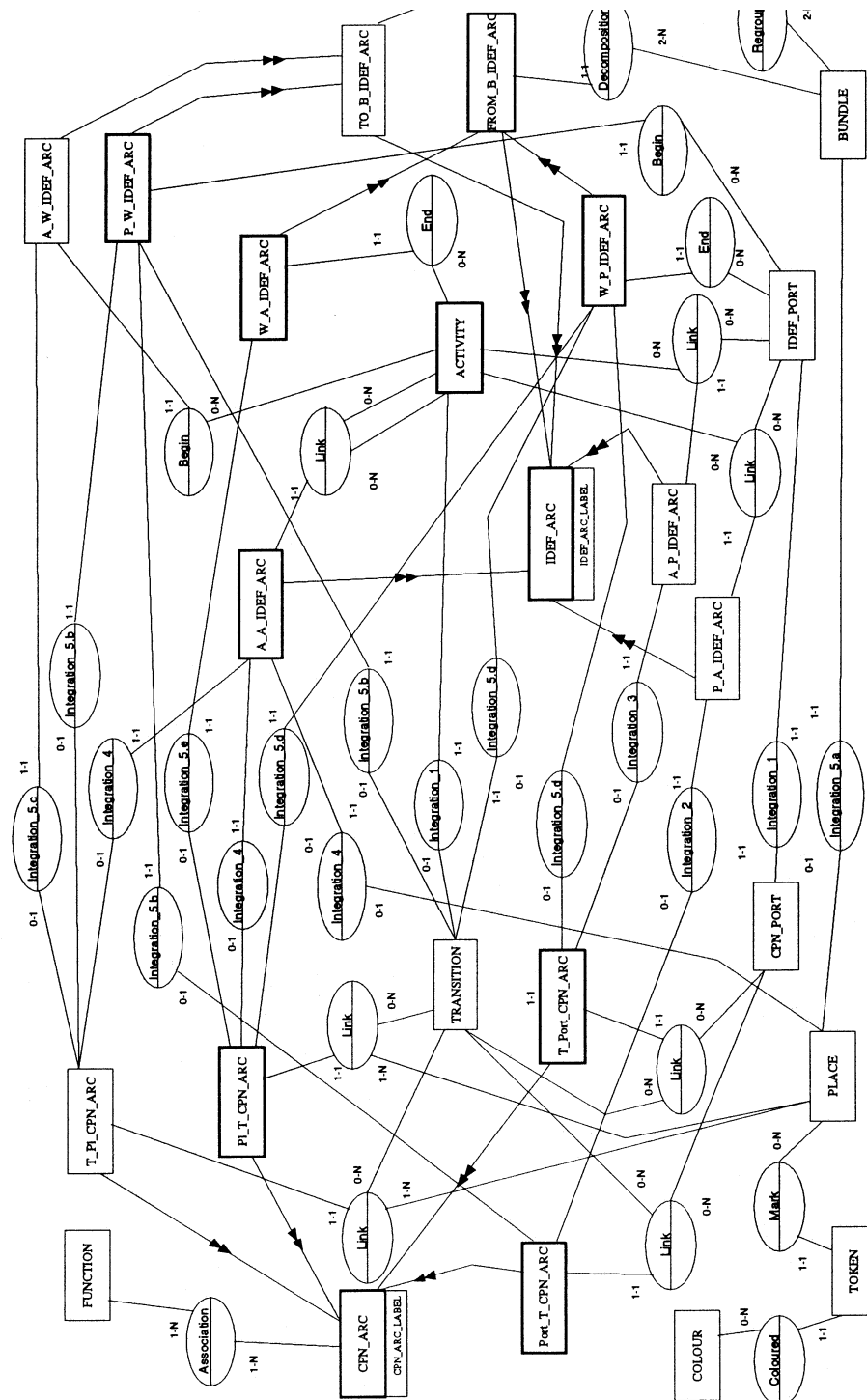


Fig. 4. Informal expression of the integration between IDEF0 and colored Petri net proposed by Brode (1988)



modelling techniques (IDEF0 and CPN) at the middle of the metamodel;

- ## 6. CONCLUSION

The design approach of complex systems today tends towards the use of integrated methods which call for a large number of modelling techniques. To be practicable, these methods must be formalized in a clear and unambiguous fashion. The aim of the formal approach put forward in this paper, is to contribute to

the practicability and to the quality of design methods by increasing their reliability and their degree of automation. Metamodelling allows for a formal expression to be given for modelling techniques as well as for the integration approach applied in each method. Furthermore, the examples given show the relevance of metamodelling to approaches for integration.

Few limits have been shown too, notably to the formalization of the procedure followed by the designers. This dynamic aspect of a method may advantageously be translated into a declarative form. On going research tries to take into account this limit in order to integrate semantic and dynamic aspects of a method in a global metamodel.

7. REFERENCES

- Böhms, H.M., and F.P. Tolman (1993). *A methodology for expressing CIM reference architectures*, Robotics & Computer-Integrated Manufacturing, Vol. 10, No. 1/2 (to appear).
- Dembélé, S., J.P. Boussières, and F. Lhote (1992). *Systemic modelization of material flow in manufacturing workshops*, Proceedings of ADPM'92, pp. 153-160, Afcet, Paris.
- Hatley, D., and I. Pirbhai (1988) *Strategies for real-time system specification*, Dorset House Publishing, New York.
- Kiefer, F., and J.J. Lesage (1992). *A methodological integration between functional and data analysis*, CIRP Proceedings - Manufacturing Systems, Vol. 21, (to appear).
- Lhoste, P., H. Panetto, and M. Roesch (1992). *Grafcet: from syntax to semantics*, APII special issue «Advances in GRAFCET», Dunod, Paris, (to appear).
- Marca, D.A., and C.L. Mc Gowan (1988). *SADT: Structured Analysis Design Technique*, McGraw-Hill, New York.
- Murata, T. (1989). *Petri nets: properties, analysis and applications*, Proceedings of the IEEE, Vol. 77, pp. 541-580.
- Nilsen, A., and H. Holm (1991). *A generic and integrated architecture for real-time control of automatic shop floor in one of a kind production*, Proceedings of CAPE'91, pp. 637-648, North-Holland.
- Roboam, M., M. Zanttin, and L. Pun (1989). *GRAI-IDEF0-MERISE (GIM): integrated methodology to analyse and design manufacturing systems*, Computer-integrated manufacturing systems, Vol. 2, pp. 82-98.
- Ross, D.T., and J.W. Brackett (1979). *ICAM definition methodology: Guide to understanding activity diagrams IDEF0*, US Air Force Report, Ref: CAM-I Ltr dtd 17Jul79.
- Vallespir, B., D. Chen, M. Zanettin and G. Doumeingts (1991). *Definition of a CIM architecture within the ESPRIT 'IMPACS' project*, Proceedings of CAPE'91, pp. 731-738, North-Holland.
- Ward, P. (1986). *The transformation schema: an extension of the data flow diagram to represent control and timing*, IEEE transaction on software engineering, Vol. 12, pp. 198-210.