Extending the Scope of Information Modelling.

Janis A. Bubenko jr

Department of Computer and Systems Science
Royal Instiiute of Technology,
and
Swedish Institute for Systems Development (SISU)

ELECTRUM 230, S-16440, Kista, Sweden

janis@dsv.su.se; janis@sisu.se

Report Nr. DSV 93-034[*]
August, 1993

Abstract

Traditional information modelling methods and techniques have mainly been concerned with promoting various representation formalisms for formally describing information systems, and discussing the expressive power of them. In this talk we take another track. Information systems and their formal descriptions exist for some reason. Entities, relationships and rules are defined because there is some need and rationale behind it. The problem here is to bridge the gap between ill-defined problem and application situations and specification of the formal, precise definition of functional requirements of the information system. This gap is wide and spans different "cultures" of thought and scientific inquiry in the field of computing and information science.

In this paper I outline an extended requirements and information modelling paradigm, based on "enterprise modelling", i.e. the capture, negotiations about, and specification of business objectives, concepts, rules activities and actors. This paradigm brings up a number of new problems and issues of scientific inquiry, which augment the traditional approach and brings it closer to problems experienced in real-life system development and maintenance projects.

---

[*] Invited paper Fourth International Workshop on the Deductive Approach to Information Systems and Databases, Lloret, Costa Brava (Catalonia), Sept. 20-22, 1993. Department de Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya, Report de Recerca LSI/93-25, Barcelona. The author's work on this paper is supported by SISU's participation in the ESPRIT project F3 (6612), sponsored by NUTEK, Sweden

1. Introduction

Would it not be good to know *why* things look or are as they do? Why are houses in Amsterdam so narrow on the street side? Why does the Sidney Opera House have its particular shape? Why do cars look as they do? Why does an information system have a particular functionality? In general - why does an artefact, designed by man, have a particular shape or functionality? During all centuries people have been designing systems, but neglected to systematically collect and record knowledge of design decisions, knowledge of why designers design things as they do. Certainly, many designers do not think this is possible. Some even reject the concept of a design method. At the same time we are trying to teach students of how to design houses, cities, information systems and other artefacts. How can we do that without knowing what designing is?

I have chosen this opportunity to write down some of the problems, ideas, and concerns, I have experienced during the last ten years by working as a manager as well as a professor, participated in some CEC ESPRIT projects, notably F^3 - "from Fuzzy to Formal"[10]. I entered the computing field in 1957. During the first years i this field, my interest in computers was mainly in numerical calculations of engineering structures. It then shifted to the architecture of basic software, notably operating systems. I did a fair amount of analytical and simulation work to calculate and estimate real-time system throughput and response times. In the early seventies, however, my interest switched to conceptual data structures and tools for system structure modelling. My view of systems development work during the seventies was, however, quite positivistic, i.e. I was not much concerned with the discrepancies between what was stored in the information base, and the phenomena it was reflecting in the "subjective reality". I was not very much worried about how the conceptual models, including business rules and other knowledge, were constructed, and how users participated in that process. During the eighties my interest, however, slowly progressed to considering also of how users and customers of systems did look at systems and the way their input of views, needs, and knowledge was collected and recorded in the requirements specification.

Computers were quickly moving into non-routine and organisationally embedded and critical applications. Users became more and more involved in participating in the early requirements specifications and modelling stages. At SISU, the Swedish Institute for Systems Development, we developed a number of special models to discuss and to represent the WHY part of information systems requirements. We call it Business Modelling. Business Modelling has had a great impact on our relations with affiliated companies and customers. A number of positive effects have been registered such as improvement of the users' understanding of their own application domain and its problems, improved user - designer communication, and an improved basis for designing the information system. We noted also the close resemblance of issues and problems in information systems design with general design problems, mainly in architecture [7]. An interesting observation is that Business Modelling is definitely liked by non-computer people. It has been more difficult to "sell" the WHY-modelling to computer and software experts.

SISU's experience in Business Modelling was an essential component of formulating the proposal for the ESPRIT project F^3. One of the important research objectives of F^3 is the process of moving from fuzzy and unordered objectives and requirements expressions, to ordered and formal functional requirements, in terms of a conceptual model, and a set of non-functional requirements of an information system. Our contribution to F^3 is in the fuzzy to formal area. An extended Business Modelling approach forms the basis for documenting the WHY and WHAT parts of a requirements specification, and link it then to the HOW parts of the specification. The WHY & WHAT parts describe the objectives as well as problems and opportunities of the information system's environment, the "enterprise", that form the basis for designing a new information system. This description also includes basic concepts, processes, and actors of the enterprise, as well as general information system requirements.

We will in this paper show the components of the WHY & WHAT parts, the so called Enterprise Model, and discuss how it can be applied in practical situations. We examine in section 2 general objectives and roles of a requirements specification. We note its importance as an essential part, not only for user designer communication, but also as a part of the user-designer "contract. Section 3 explains the five submodels of the Enterprise model. We use as illustration the F^3 project case itself, i.e. we are showing fractions of an Enterprise Model of the F^3 project. In section 4 we comment on the "way of working" when developing a requirements specification, and the Enterprise Model in particular. This is a part we have little experience with yet. The F^3 Enterprise Modelling approach is in the process of being developed in detail. While it seems so far rather "unique" and gives a substantial extension over existing "standard" methods, by including the WHY-part, we cannot today assure all its benefits or drawbacks. The assessment of a method is discussed in section 5, where we point out that testing (appropriation) of a method in a practical situation is not a repeatable process. Some future research issues are given in the concluding section.

2. Requirements specification

This section takes a general, top-view of the requirements specification problem. We observe that this is largely a problem of user and designer competence, and a problem of communication. The structured, model based approach of F^3 is here assumed to improve both. The different roles of a requirements specification are then discussed. The difficulties in managing a requirements specification process are also noted.

2.1 Objectives of requirements specification

No useful system can be built in isolation. It must correspond to anticipated user needs and requirements. It must always "fit" a context of existing components and systems [2, 16]. The system must also be able to sell - for a price. Information systems are exactly like this. An additional problem here is the difficulty to evaluate the benefits, as information systems are organisational infrastructure components. Developing business requirements for information systems is a task of knowledge acquisition and management. It is also an issue of design of business processes as well as of organisational components and actors. This implies complex communication and validation of design results, and of selecting knowledge sources, methods, tools, and techniques. It is furthermore a problem of selecting ways of working according to the situation, the purpose and the expected result.

In practice, the term "requirements capture" is often used. The term is, however, somewhat misleading because it indicates that the requirements "are already there", and all we need is some method and tool-set to better apprehend and to get the requirements from the user. Everyone who has been playing the role of a user knows that this is never the case. What we need is concepts, methods and tools that can help us to analyse a situation, to design (or re-design) business processes, and to develop and define information system requirements step by step together with the users and requirements holders, without getting lost in details and loosing sight of the whole picture.

A big problem, experienced by most business managers as well as data processing managers in Sweden, is the lack of good "requirements holder competence regarding requirements specification" (source: The Swedish Data Processing Association's Survey, RODER, 1992) and, consequently, the lack of good communication between system developers and requirements holders and users. Requirements specifications are therefore, more often than not, inadequate for contracting and evaluating information systems engineering work. We believe the situation is similar in many other countries. Similar concerns in the software engineering field in U.S.A. have been expressed in the well-known article by Curtis et. al. [8], where communication and domain knowledge problems have been found to be factors hindering adequate systems to be developed.

A distinguished feature of the F^3 approach is the method for developing, acquiring, and communicating early, enterprise knowledge and user requirements by a structured, iterative, working and modelling approach. The approach is conceptually guided by a number of conceptual sub-models, each focusing on a particular aspect of the application. This part of the requirements gathering process is called "Enterprise Modelling". An Enterprise Model for an application then contains populated sub-models, or descriptions, conforming to (being instances of) the different sub-meta-models.

A considerable number of commercially exploited methods and CASE environments for information systems work exist today. However, these products address the middle and/or later stages of the systems development life-cycle. Practically none of them address, in a model based, structured way, the early, business objectives driven, requirements generating stages, and the problem of moving from the informal to the formal domain (this is also confirmed by the EUROMETHOD method classification survey of some widely used methods in Europe). Existing methods are not designed for explicit capturing, and representing in a structured way, of "business and organisational knowledge" to be subsequently used by the information system design phase. Links between business and enterprise models and information system specification models are not maintained. Management of change and evolution of the enterprise, and mapping of changes there on information system requirements and components, cannot therefore be explicitly done.

The importance of establishing a firm and explicit connection between the development of business objectives and strategies, and the information system development, is manifested in a number of references, e.g. [12, 29, 31, 32]. It is usually rather difficult and/or costly to make other than cosmetic changes to an information system, already in operation. This fact, together with the importance of the information systems to the business, the high cost of developing new systems, and the lifetime of an information system of approximately 6-7 years, indicates the importance of, both understanding the business, where it is going, and deriving the requirements from this understanding. An adaptable information system can be a direct asset of the business, while a system difficult to continuously customise to the ever changing business needs can severely hinder the evolution and the running of the business.

The expected traits of the F^3 methodology are not only extended modelling facilities for requirements capturing and reasoning about requirements specifications, but also a method for establishing a more solid and complete basis for the design and implementation of information systems which are expected to exhibit improved properties in terms of completeness, changeability, and manageability.

2.2 The roles of a Requirements Specification

A requirements specification plays several roles in an information systems development process. It is the basis for and the part of a contract between requirements holders and system developers. It is an architectural, implementation independent "drawing" of the future system to be ordered. It provides an explicit basis for reasoning with and among the clients about various qualities of the system to be built. Later, after delivery of the system, it is the only (!) instrument that can be used to check if the implemented system corresponds to what has been ordered. When the environment and, consequently, the requirements on an information system change, as they frequently do, the requirements specification's role becomes a model and a description to be used to re-model changes in the environment and discuss how these changes should be accommodated by changes of the information system. The focus of Enterprise Modelling work in F^3 is to develop a framework of concepts, models, and methods by which requirements can be effectively captured, specified, and communicated. Thereby it provides an input to the Information System design and prototyping activities where the focus is on formal and executable information systems specifications.

According to the F^3 approach, an Enterprise Model is a partial description of basic assumptions, requirements, and knowledge of the problem domain, needed for designing and implementing an information system. It contains also information to be referred to in the "contract". More specifically, it includes interrelated components describing issues like:

- Why is the information system needed? Which are the goals of the environment, the application? What are the importance and priorities of goals? How are goals related to each other?
- What concepts or subjects is the enterprise about (including their relationships to objectives, activities and processes, and actors) ? How are they defined? What business rules and constraints monitor these objects?
- Which business activities and processes are there, or should be there, to manage the organisation in agreement with the goals? How are the business processes, tasks etc. performed (work-flows, process models)? Which are their information needs?
- who is performing which processes and tasks?
- what requirements on the information system to be developed are generated by these processes?

The above list represents some of the descriptions a Requirements Engineering process must consider and include in a requirements specification. A methodology for Requirements Engineering needs, however, also methods and processes that support the work with and the production of such descriptions. Methods are needed for how to work with users and clients in very early stages in order to, in a participative way, acquire and validate requirements and domain knowledge (at several levels of fuzziness, precision, and agreement) as effectively as possible. Requirements acquisition is, typically, characterised by:

- co-operative work, where co-ordination is vital,

- complex user - analyst dialogue, interaction, participation, and problems of (mutual) understanding of enterprise goals, concepts, processes, and other issues,
- management of reasoning and decisions by individuals as well as by groups,
- often ill-defined, unclear problem situations,
- the need for re-use of existing systems and/or knowledge of existing systems,
- the need to comply to predefined organisational rules and restrictions,
- the need to resolve conflicting user/client views and requirements,
- need to formulate WHY and WHAT rather than HOW, i.e. to formulate the reason, the problem and the "equations of the domain", rather than the solution,
- the need to consider many situational factors, such as user skills, attitudes, experience, and leadership, in order to take the most adequate approach,
- difficult project management: high level of uncertainty, unknown, detailed task structure, dynamically evolving, and
- the need for support of controlling a complex design process, and the need for good configuration control, and design management.

The quality of requirements specifications has a critical impact on the utility and maintenance cost (re-work) of the implemented system [3]. Therefore, the above aspects must be considered in developing an adequate method and a support environment for Requirements Engineering. Requirements analysis for information systems is a strongly iterative and creative *design* activity. The problems experienced are "wicked" and ill-structured [26, 27]. In many respects, in particular concerning the acquisition and management of initial, "fuzzy" requirements, it resembles the process of requirements acquisition and the design of spatial objects in architecture [17].

3. Enterprise Modelling in F^3 - an objectives driven approach

In this section we will describe how conceptual modelling of information systems, in the ESPRIT project F^3, is extended by modelling also of the WHY and WHAT part of the requirements. This is done in the Enterprise Modelling part.

3.1 Overview

The main objective of the Enterprise Model (EM) and the enterprise modelling technique, is to understand and to document the business, its objectives, problems, concepts, actors, and activities, better and in a more structured way than by simply providing natural language descriptions. An enterprise model is the structured description application knowledge, assumptions, and requirements, relevant and needed for the design and implementation of an information system. In our view, it has to answer the following questions:

1) Why is the system built, what is its justification?,
2) Which are the business processes, and which are to be supported by the system?,

3) Which are the actors of the organisation performing the processes?,

4) What concepts are they processing or talking about; which are their information needs?, and

5) Which initial objectives and requirements can be stated regarding the information system to be developed?

Enterprise modelling and requirements specification is based on the notion that a requirements specification process, from a documentation point of view, implies populating (instantiating) a number of interrelated meta-models (Figure 3-1). The Enterprise Model (EM) includes the following submodels: an objectives model (OM), an activities and usage model (AUM), an actor model (AM), a concept model (CM), and an information system requirements model (ISRM). These submodels correspond to questions 1) to 5) above.

As can be understood from the figure, the Objectives Model is assumed to be *motivating*, or "driving", the development of the other submodel instances of the Enterprise Model. It is, however, not always this path, that is the best in practical situations. At times some information system requirements "are already there" (right or wrong) and can easily be stated. Elaboration of them, is then done "upward" by asking the reason for their existence, the concepts they concern, the activities they apply to, etc. At other times it may be helpful to start the modelling with some essential concepts or objects of the enterprise. The other kinds of intermodel relationships, 1 to 3 in figure 3-1, are briefly explained in the figure.

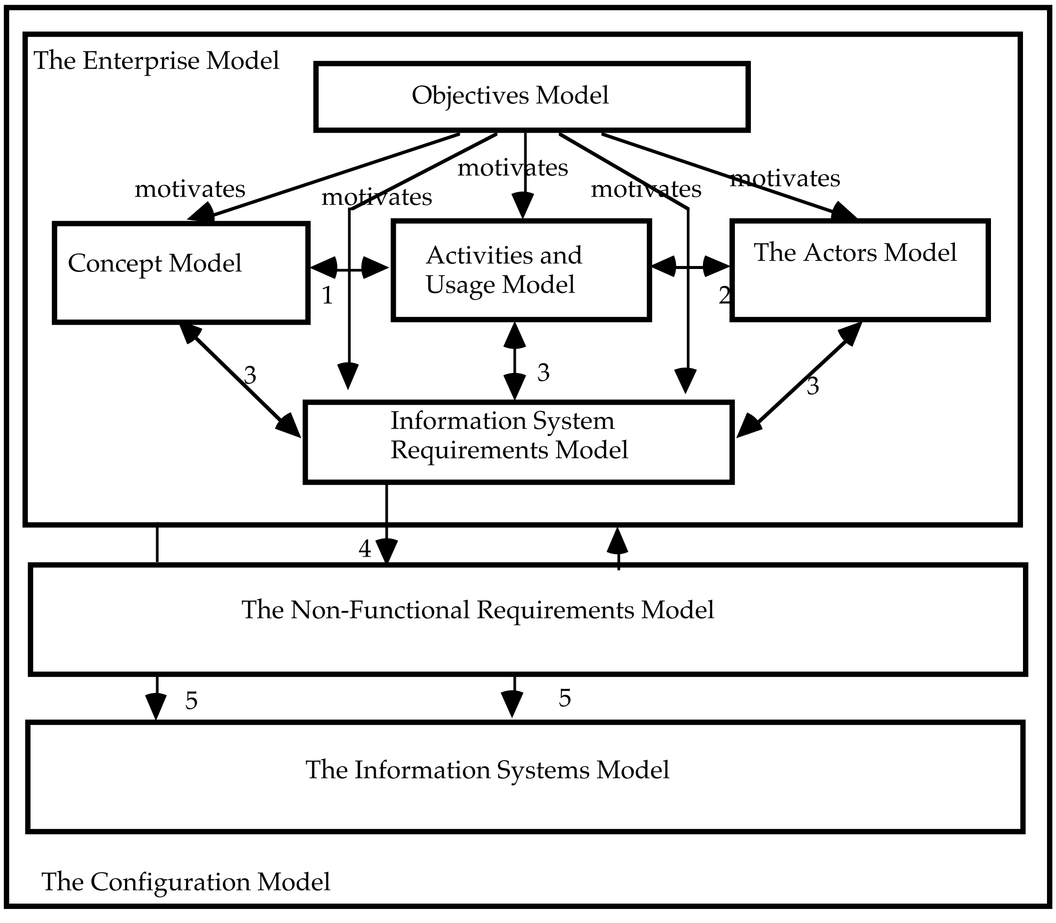
The EM is linked by "implemented_by" relationships to the information system model (ISM) components and by "refined_by" relationships to components of the model representing more precisely defined non-functional requirements (NFRM) of the information system [6, 18]. The development and design of instances of EM, ISM and NFRM is in F^3 supported by additional methods and models. For instance, models for re-use of existing model and component libraries will use a technique developed in the ESPRIT-project ITHACA [5]. The approach for managing and tracing the development process and its decisions and model manipulating operations [20, 21] has many issues in common with the approach of the ESPRIT project NATURE [15]. A knowledge-based technique, ERCAV (Early Requirements Capture And Validation), is also being developed, which will support refinement and operationalisation of EM components [9].

The concept of "Enterprise" should here be interpreted in a very wide sense. It could actually mean the whole enterprise or it could mean some small part of it. More often than not, in developing information systems, the concept "enterprise" will denote a limited area of activity of the enterprise of which we are concerned. As we will show later, a particular interpretation of "enterprise" could be *the process of capturing requirements and developing a requirements specification*. Depending on how familiar we are with the application domain, its conceptual structure, and the requirements for computer support, the need to perform an extensive, in-depth environment modelling may vary. For instance, in a limited technical situation, where the requirements and the conceptual model is well defined, the need for extensive enterprise modelling is limited.

In the following we will examine more in detail the different submodels of the Enterprise Model. We will exemplify them using fragments of the F^3 project case itself, where a requirements modelling method and a tool-set, which here is the *information system* for model building, group work and communication, is being developed.

3.2 The Enterprise Submodels

The Enterprise Modelling framework includes a number of interrelated submodels, as shown in Figure 3-1. In the information system modelling and design phase, semantic links are established from these models to the information systems model, reflecting the rationale, or the motivation, for designing specific IS components. For instance, when selecting an object of the Information Systems Model (e.g. a data model component), it should be possible to trace is rationale and origins (e.g. why has it been introduced?) to the EM- and NFRM components.



1 "deals-with" type of relationships
2 "performed-by" type of relationships
3 "concerns" type of relationships
4 "refined-by" type of relationships
5 "implemented-by" and "supported-by" type of relationships

Fig. 3-1. The Enterprise Model and its interrelated models

The enterprise submodels contain each a number of components, i.e. objects and relationships of different types, which are typical for that submodel. Relationships can be internal to a submodel ("intra" relationships) or "external", where they relate components of two or more submodels (“inter” relationships). Objects as well as relationships may have attributes. Populating models and their relationships is, furthermore, controlled by a number of static as well as dynamic consistency rules, which control their state consistency as well as their permissible state transitions.

The different submodels are exemplified using the F^3 case itself, i.e. we are here showing a few components of the requirements statement and modelling effort, at some development stages, when the requirements for the F^3 method and tool-set were developed [4, 11, 13]. A few words about the case are needed. F^3 is in the process of developing a new method and a tool-set for requirements engineering. It is assumed that the method and tool-set will be applicable in a large number of organisations and for a large number of types of systems. The F^3 case's "enterprise" is the environment of using a new method and a supporting tool-set for capture, refinement and specification of requirements for an organisation's information system. While the environment may differ from one user to another, we have to hypothetically assume a particular type of actors and working procedures. We have to assume a set of objectives, problems, and requirements for a method and the supporting tool-set. In our case, these assumptions are made on the basis of initial statements about requirements for methods and tools made by F^3's industrial user partners. The statements made by them in the initial stages are neither consistent, operational, nor complete.

This is how the requirements engineering process is assumed to start. The problems we are tackling in requirements engineering are, typically, "ill-structured" and "wicked" [26]. This means, among other issues, that 1) there is no definitive formulation of a wicked problem, 2) it is not easy to know when the job is done, 3) there is no way of testing the "solution", and that 4) every wicked problem is essentially unique. As Rittel writes: "Here (in the solution of wicked problems) the aim is not to find the truth, but to improve the characteristics of the world where people live. Planners are liable for the consequences of the actions they generate; the effects can matter a great deal to those people that are touched by those actions" (see [26] p. 144).

It seems that general principles, issues and problems of technology, planning and design are equally applicable in determining requirements for our information systems. There is no subjectively right or best system for a particular problem and environment. Furthermore, the environment never stays still, and the problems, it has to tackle, change. Requirements for information systems are generated by people. They reflect what people currently think is important for the organisation or for themselves. The examples shown below in connection with the different model types should be interpreted like this - they are current *beliefs of people*, asked what properties they would find useful in a method and a tool for requirements engineering. The people - the "design actors" - are the ones that decide what a "good" or "satisfactory" solution is.

3.3 The Objectives submodel

The objectives submodel (OM) is intended for describing and discussing the reason or motivation for activities, actors, and concepts of the other submodels - it addresses the “why”-perspective of the enterprise and IS development. The concepts of this model are related to the enterprise itself and determined by its members' rationale. The objectives model component types are typically: enterprise goal, problem, opportunity, cause, rule, and development action (task). The relationship types used are *motivates*, and *influences*. For instance, a goal may motivate a set of sub-goals, and influence (positively or negatively) another set of goals. The objectives model component types and relationships are informally described as follows.

GOAL	A goal describes a desired set of states of the enterprise a design agent aspires the enterprise to be in. A goal can be operational (measurable) or non-operational. It may concern different strategic levels (strategic, tactical, operational), and concern different subject matters, e.g. business areas, processes, or products.
PROBLEM	A problem describes that the design actor finds the enterprise in a set of states which is not desirable. Problems typically imply that the design agent is aware (or can be made aware) of a goal which is hindered by the problem so formulated. Thus, every problem must, eventually in the modelling process, be related to a goal by a relationship of type negative influence.
CAUSE	A cause is a description of the reason for a problem to exist.
OPPORTUNITY	An opportunity describes a situation, that exists in or is available for the enterprise, and that can be taken advantage of in continued modelling work, e.g. to support some goal.
RULE	A rule describes something affecting the enterprise, that limits the actions that can be taken.
DEVELOPMENT ACTION	Describes development work that has to be done to achieve some goal. They form a set of development tasks that eventually have to be carried out. Normally, development actions are found at the more detailed and refined end of an objectives model structure.
motivates	Motivates is a relationship that typically refines a higher level (and often "fuzzy") OM component into a set of more precise components. Motivates can also be used to refine rules and to link components of the OM to components of other types of submodels.
influences	Influences is a relationship that describes how one component of the Objectives Model may influence another (set of) components. The influential link has a direction, a sign and a strength. The direction indicates the causality direction between the components, the sign indicates whether the influence is contributional or the opposite, the strength, if applied, indicates the degree of influence, i.e. strong, medium, weak.

In figure 3-2 an example of a small fraction of some version of the objectives model of F^3, at some development stage, is shown. A number of issues can be discussed here. First we should note that the "enterprise" and its objectives here is the hypothetical organisation being a future user of the F^3 method and tool-set. The objectives model is therefore concerned with setting up the goals, problems, etc. for the method and its appropriation in the environment.

We can assume that this model has been developed in a number of group sessions, where participants have each "posted" a number of different goals, problems, opportunities, and development actions. The problem now is to "sort the things out", i.e. to eliminate, as much as possible, redundancies in the propositions, to relate propositions to each other by motivate and by influence relationships, to refine components by introducing "finer grained" components, and, in general, to "agree" on what has been posted. The things shown in the figure reflect a particular stage of analysis. At this stage, there is, obviously, yet no common agreement nor understanding among the design actors of all the things posted. Some components are not connected to other components, e.g. O25 (Users are not sufficiently involved in the RE process) and O63 (Users do not realise the severity of not being involved). What goals do these problem statements reflect? Whether O9 (Users should not need to know a formal representation technique) influences O30 (The approach must be based on a number of meta-models) is also a question to be further discussed, and the components perhaps reformulated.