# On Related Pragmatic Categories and Dependencies within Enterprise Modelling

*Remigijus GUSTAS*

Swedish Institute for Systems Development
Electrum 212, S-164 40 Kista, Sweden[1]
e-mail: gustas@sisu.se

## 1 Introduction

Requirements modelling is a highly critical phase in the development of computerised information system. The consequences of errors are difficult to trace in the further development phases. Usually, the price paid for complex information system reengineering activities is relatively too big, compared to the cost of development the completely new version of it. A great variety of problems such as incompleteness, contradictions, ambiguities, over specifications arise during analysis of several specifications of the same information system.

The starting point for information systems requirements modelling research are initial requirement statements. These statements express wishes of designers about what the system should do. Initial requirements are usually interpreted as objectives, opportunities and problems, and presented in the form of natural language text. As a matter of fact, these pragmatic representations are often contradictory, ambiguous, incomplete and inconsistent. The objective is to produce the description that it could be suitable for developers of technical system. A great variety of models and languages have been proposed to capture precise, unambiguous, complete and consistent requirements. This process is entitled 'from fuzzy to formal'[F3_94] and schematically could be represented by Fig.1.1.

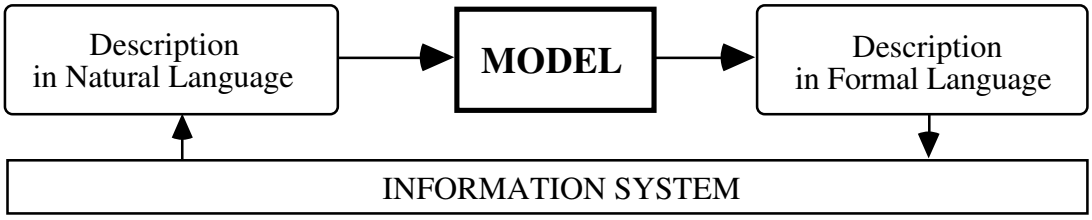


**Fig.1.1.** Illustration of the process from 'fuzzy to formal'

Ideally, wishes of potential users could be described as initial objectives which constitute a driving force for further information system design activities. Goals are important in several aspects [DLF93]:

---

[1]The author is on leave fromInformation Systems Dept., Kaunas University of Technology, Lithuania;e-mail: remigijus.gustas@if.ktu.lt

1) goals may motivate completeness of requirements, or may provide some help in detection over specifications, i.e. to discover requirements which have no motivation,

2) goals justify and explain the presence of requirements components which are not necessarily comprehensible to users of information system when they are separated from each other,

3) goals may be used to understand responsibilities of actors, i.e. they provide the basis for assigning actions to agents of the system,

4) structure of goals provide basic elements for detecting and resolving conflicts that arise from different viewpoints. Goals lead to integration of requirements which should support them.

The framework of Enterprise Modelling [F3_94] includes a number of interrelated submodels. The semantic links established in these submodels are reflecting rationale and motivation for specific information system parts [Bub93]. The enterprise model reflects specification of the information system semantic and pragmatic components. It shows the essential entities, activities, and how they interact. Information system model determines the data elements and functions that represent business entities and processes[SoZa92]. So, the enterprise model deals with understanding perspective while information system model corresponds to representation perspective [Pohl93].

The concept of enterprise here should be interpreted in a very wide sense [Bub93]. It could actually mean description of qualitative aspects of business of the whole enterprise or some small part of it. In the context of information system development, the concept of enterprise rather denote a limited area of activity of the organisation which is concerned by an application designer. The enterprise submodels are based each on a number of pragmatic and semantic dependencies which are typical for that submodel. There are four enterprise modelling components which are used for understanding of business requirements: Objectives submodel, Activities submodel, Actors submodel, Concepts submodel.

Each enterprise submodel focuses on the specific business modelling aspect[Bub93]. The objectives are motivating and driving the development of other parts of enterprise model. Sometimes, other parts of requirements can be easily stated before goals are well-understood. Elaboration of objectives submodel is then done in bottom-up way, by asking the reason for their existence. This part of enterprise model addresses the 'why' perspective in information system development. The activities submodel describes the 'how' perspective and contains organisational activities, i.e. actions or tasks of the enterprise. Activities are able to make changes to states of affairs, which are expressible in terms of concepts. The actor submodel addresses the 'who' perspective. It defines the strategic dependencies [Yu94] between actors in achieving of their goals by means of message passing and triggering actions at the same time. The semantics of each message is expressible in terms of concept model elements. The concepts submodel defines static constraints among concepts. It addresses the 'what' we are talking about.

Ultimate goal of objectives modelling is to identify, to document and to analyse the information system in terms of pragmatic [LLS94] level entities such as goals, problems, causes of problems, opportunities, and specify the pragmatic relationships among them. The objectives model addresses the 'why' perspective and it is motivating and driving the development of other submodels. So, one of the main difficulties here is unambiguous description of the pragmatic level entities on the basis of modelling artefacts of the semantic level such as concepts, actors and activities submodels constructs.

# 2 Bridging Pragmatic to Semantic Representations

The objective submodel components specify pragmatic aspects[LLS94] of an information system. Those aspects can be expressed on the basis of two specifications: the specification of actual situation and the specification of desired situation of the information system. The actual situation defines the static and behavioural constraints of actual specification, and desired situation - the expected specification. The explanation of relation between two situations is depicted in Fig.2.1.

**Fig.2.1.** Relation between initial and desired specification

Development action describes an activity that has to be done to achieve are predefined set of goals. The development action take us from one situation to another, for instance from the problematic situation to the situation specified by goal statement. The activity form a set of development tasks that have to be carried out [Bub93].

Modelling of information system pragmatics typically involves such notions as goals, objectives, problems, constraints, etc. Goals are often described as desired situations that should be reached or striven for. The goals express wishes and desires concerning the properties of the product under development. A situation in this case is referred as the specification of the semantic level constraints in terms of static and dynamic dependencies. Problems are described as actual situations that are not desirable. They denote restrictions we have to suffer or try to avoid. Some of dependencies among semantic constraints and related pragmatic categories are presented in Fig. 2.3.

The notions of conceptual specification such as static and dynamic dependencies (semantic relations [Stor93]) are used for definition of constraints and rules of business. Business rules and constraints are usually specified for lower level pragmatic categories in order to satisfyce higher level goals.

Conceptual models have been established to capture a meaning of an information system. So, these models can be regarded as a basis for the specification of pragmatic aspects of a business [Gust94a]. Modelling primitives applied for specification of semantics of an information system are usually based on Entity - Relationship like (ER) notations [Chen76]. Although these notations are much more readable and understandable for human beings compared with logical formalisms, there is no uniform theoretical basis or even agreement whether the concept should be specified as entity, or as relationship, attribute, action, actor, etc. Usually, the same name of concept can have all those several roles, but just in different context of interpretation. One of the main difficulties of ER representations are that they require to distinguish sharply among such meta categories, i.e. the designer is enforced to make decision whether the concept will be interpreted as an entity, or as a relationship, or as an action, etc. strict interpretation of categories can cause different specifications of the same objective, even if the same modelling constructs are used. Because of different interpretations of the same concept, the designer is able to define the same objective in terms of several conceptual schemes. So, ER kind of notation is representation

dependent. Lack of relativeness of interpretation of basic categories of conceptual models, injure the main principle of conceptualisation [Griet82].

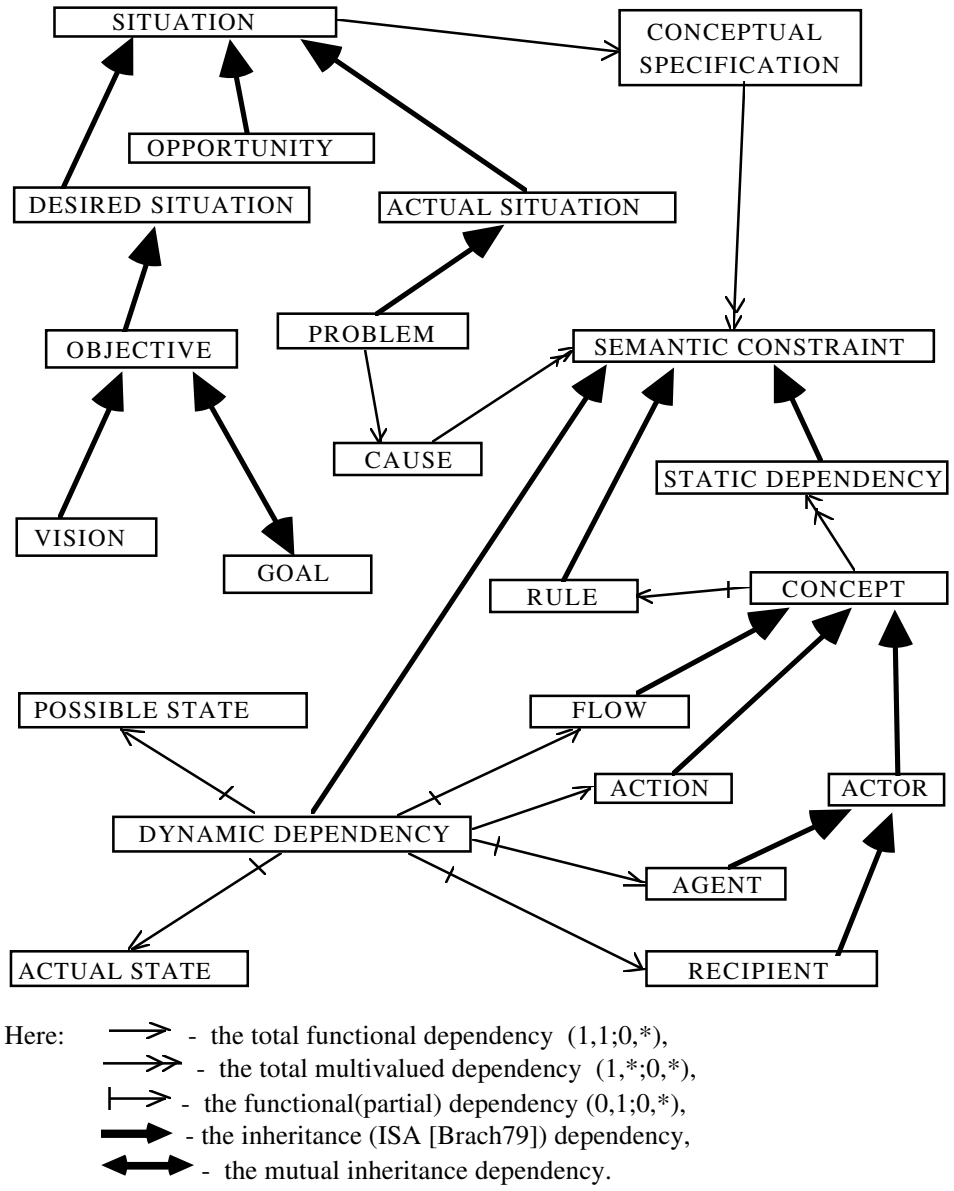


**Fig.2.3.** Meta-model of Related Semantic and Pragmatic Categories

Actually, every concept can be represented in the role of several semantic categories. Its interpretation vary and is dependent on at that time applicable set of semantic constraints. Some of dependencies among related semantic categories are presented in Fig. 5.6.

Dependencies among concepts [Gust94b] constitute the basis for derivation several semantic categories exploited in different Entity-Relationship like models [HuKi87], [Tempora92]. On an other hand, if the constraints can be represented differently, then they are dependent on representation dimension. Different representations of the same universe of discourse gives rise for discrepancies - structural differences of semantically equivalent data subschemes [JoJa94]. Lack of flexible interpretation of semantic categories can be a reason of complications in the process of integration of semantic representations caused by several goals.

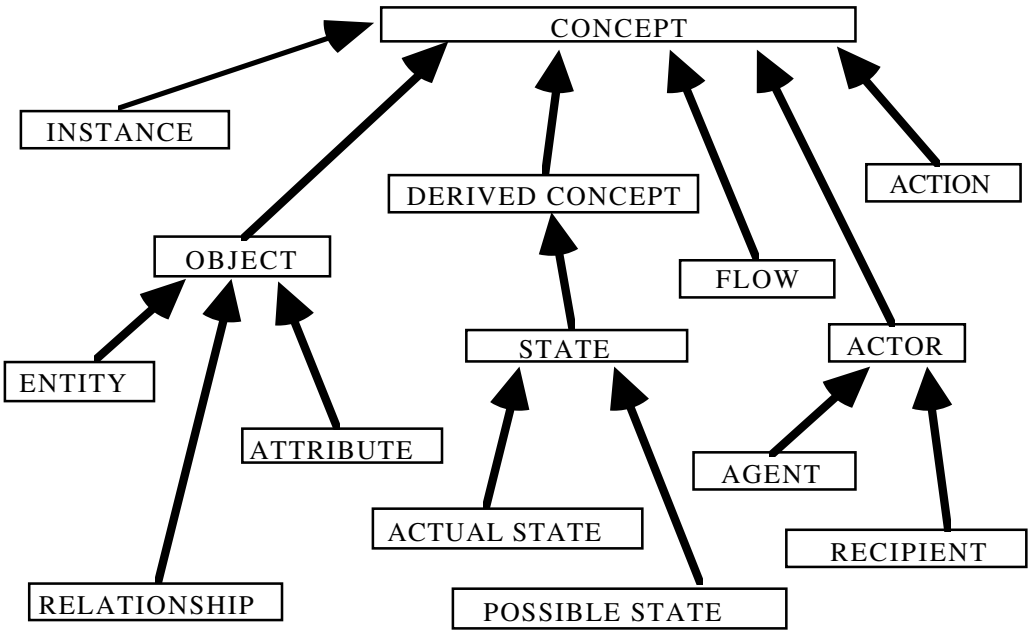


**Fig.2.4.** Related Semantic Categories

The pragmatic (non-operational) descriptions of goals, problems and opportunities are expressed by natural language statements, while the operational description of these pragmatic categories is specified in terms of static and dynamic dependencies. Semantic dependencies can be regarded as operational requirements of business. The operational requirements are specified among several semantic categories due to the existent 'laws' and restrictions of the enterprise.

# 3 Goals and Pragmatic Dependencies

A goal hierarchy could be formed of interconnected goals of different levels of abstraction ranging from higher level business objectives to lower level operational goals [Wang93]. The structural decomposition and refinement of business objectives may be understood in correspondence to the hierarchical decomposition of processes performed in structured analysis [Yourd89]. The goals are broken dawn into lower level business functions each of which is intended to perform some specific task. The decomposition goes until a level is reached where the function is sufficiently well understood, i.e. can be specified in terms of semantic constraints. The example of a goal decomposition hierarchy is depicted in Fig. 3.1.

The goal decomposition hierarchy describes the reasons for specification of information system semantics. The higher level goals gives reasons for the introduction of lower level pragmatic categories. At the lower level of a goal hierarchy, the goals are much easier to formalise than the more vague higher level business objectives. The reason is quite simple - the lower level goals consist of lesser number of semantic constraints.

The opposite side of a goal is a problem. A problem describes situation which is not desirable. This situation can be specified by the set of static and dynamic constraints. A Specification of problem is defined as a semantic difference between set

of semantic constraints at the actual situation and set of constraints at the desired situation. So, the semantics of a problem can be regarded as part of the specification of product at the actual situation. It is very important that the problem can't exist with outstating of the goal. If the designer has no predefined goal then the problem does not make sense. A goal describes a desirable situation. Specification of goal is unambiguously defined as part of semantic description of a desired situation which is deduced as a semantic difference between set of constraints at the desired situation and set of constraints at the actual situation. So, according to this understanding, the specification of the goal is an opposite side of specification of the problem. At the same time, the specification satisfyce [Simon84] the goal or the problem and is regarded as the extension of it.

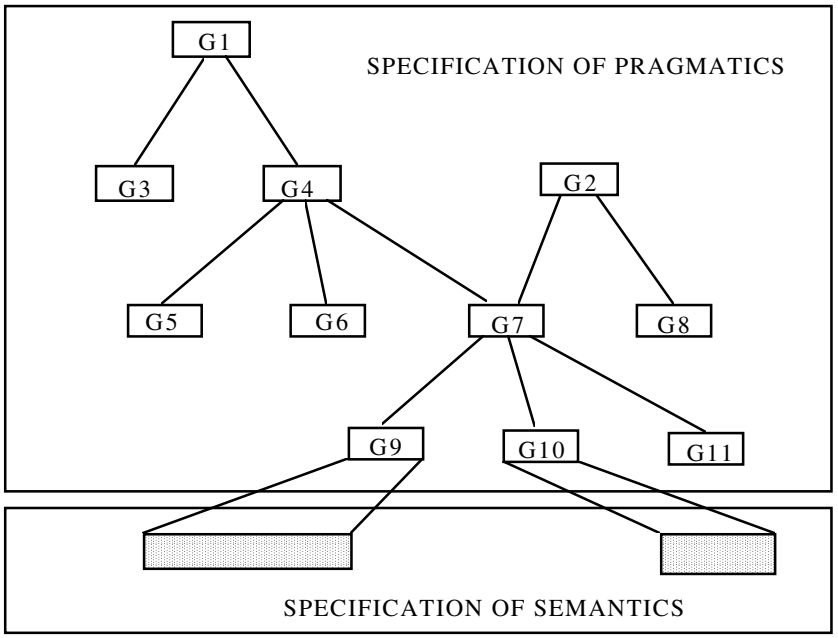


**Fig.3.1.** The goal decomposition hierarchy

Any two goals can specify contradictory statements, i.e. one goal can influence negatively or hinder to the achievement of the another goal. It means that interpretations of the notion of goal and problem are relative and dependent on actor objectives. The same description for one actor could be interpreted as a goal, while for another - as a problem. To specify contradictions between several goals of agents, the negative influence dependency [F3_94] is introduced. An influence from A to B ($A \Rightarrow B$) dependency means that situation denoted by B is affected by situation A. The negative influence pragmatic relationship( $\xrightarrow{-}$ ) from one goal to another indicates that the first hinders to the achievement of the second. The positive influence dependency ( $\xrightarrow{+}$ ) would mean, that the achievement of some goal, automatically contributes to the achievement of the another. The influence dependency is characterized by the following axioms:

1) if $A \xrightarrow{+} B$ , $B \xrightarrow{-} C$ then $A \xrightarrow{-} C$ ,
2) if $A \xrightarrow{-} B$ , $B \xrightarrow{-} C$ then $A \xrightarrow{+} C$ ,
3) if $A \xrightarrow{+} B$ , $B \xrightarrow{+} C$ then $A \xrightarrow{+} C$ .

The specialisation pragmatic dependency from A to B is denoted by $A \prec B$ if and only if the specification of B is a subset of the specification of pragmatic element A. The 'motivation' dependency [F3_94] is usually used instead of specialisation to

refine a goal. The 'cause' dependency can be used in the case of specialisation of a problem. So, the consistency of decomposition of complex pragmatic entities to simple ones can be verified as far as their structures are specified by semantic constraints. Graphical notations of types of pragmatic relationships are presented in Fig. 3.2.

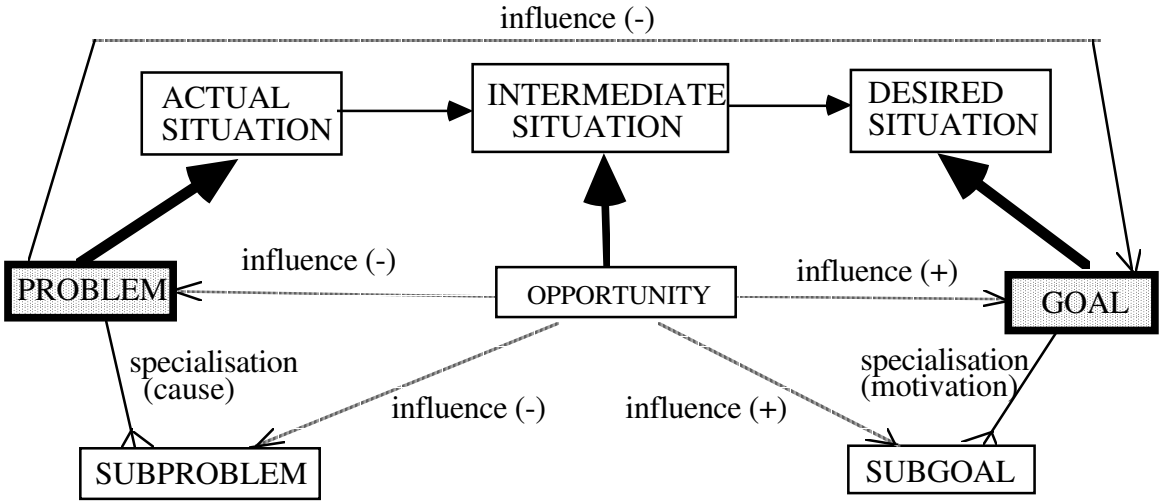


**Fig.3.2.** Types of pragmatic dependencies

Using the above introduced primitives, is not difficult to explain notions of an opportunity and a cause of problem exploited in the area of enterprise modelling [F3_94]. Opportunities express statements that may be referred to improve an already existing situation. Interpretation of the notion of opportunity is relative and dependent on the context. In some enterprise modelling methodologies[F3_94], [Will93] definition of it may be different. Sometimes, an opportunity is regarded as an intermediate situation that can exist between a problematic situation and a desired situation at some phase of the development action. It can be represented by a part of the desired specification (see Fig.3.2). Often opportunities are transformed into subgoals. Another interpretation of an opportunity can be defined as a kind of development action which is able to take us from one situation to another, for instance from the problematic situation to the situation specified by goal or subgoal statement. This case is illustrated in Fig.3.3.

A cause of problem can have different interpretations as well. One of the most simple definitions of a cause could be based on the notion of situation which is a reason for a problem to exist. Causes are usually used for specification the explanations of problems and regarded as extensions of problems. Every specification of subproblem is a cause for problem, and thereby, subproblem cause problem. Cause of a problem could be formed by union of causes of subproblems. If the semantics of each cause can be expressed by exactly one constraint rule, then it shows very higher quality of structuring in the phase of goal modelling.

The specialisation dependency relates more specific goals or problems with more general ones. If goal A is motivated by goal B then the extensionof goal A (semantic specification) constitute part of specification of goal B. It allows to define some useful operations on goals (problems), such as operation of union and operation of intersection of consistent goals. If subgoals are conflicting the goal is inconsistent. The goals are consistent if and only if they are not contradictory. Any two consistent

goals $G_2$ and $G_3$ have their greater common specialisation denoted by intersection ($G_2 \sqcap G_3$) and their least common generalisation denoted by union ($G_2 \sqcup G_3$).

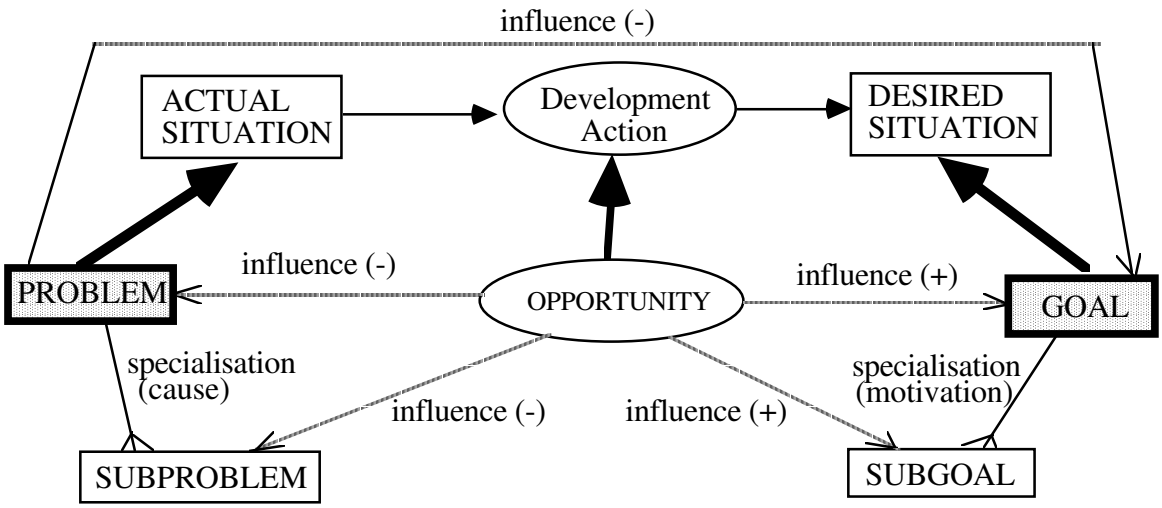


**Fig.3.3.** Pragmatic categories and dependencies

Union of goals is defined as union of specifications of them. If any two goals $G_2$ and $G_3$ are specialisation the goal $G_1$ (see Fig.3.4,a) then their union is the specialisation the same goal as well.

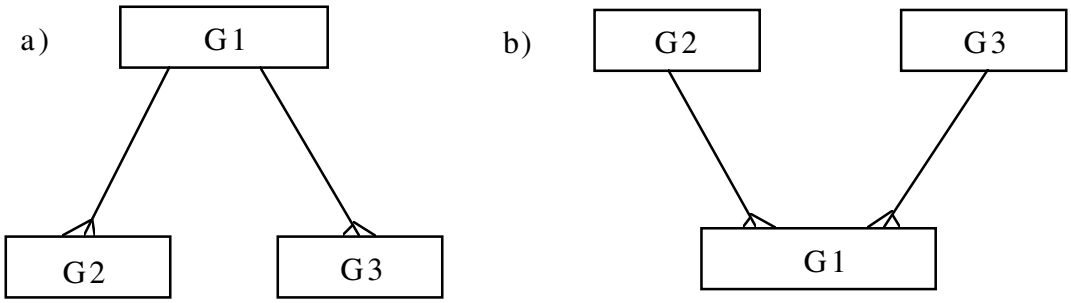


**Fig.3.4.** Operations with goals

Operation of union of goals can be characterised by the following properties:

1) $(G_2 \sqcup G_3) \succ\!\!- G_1$ if and only if $G_2 \succ\!\!- G_1$ ,$G_3 \succ\!\!- G_1$ ,

2) $G_2 \succ\!\!- (G_2 \sqcup G_3)$ ,

3) $G_3 \succ\!\!- (G_2 \sqcup G_3)$ ;

Intersection of goals is defined as intersection of extensionals of goals. If any goal $G_1$ is the specialisation two goals $G_2$ and $G_3$ (see Fig.3.4,b) then it is the specialisation of intersection of these goals. Operation of intersection of goals can be characterized by the following properties:

1) $G_1 \succ\!\!- (G_2 \sqcap G_3)$ if and only if $G_1 \succ\!\!- G_2$ ,$G_1 \succ\!\!- G_3$ ,

2) $(G_2 \sqcap G_3) \succ\!\!- G_2$ ,

3) $(G_2 \sqcap G_3) \succ\!\!- G_3$ .

Using the set of pragmatic relationships it is possible to infer some other valid (not specified) dependencies of the objectives model. The pragmatic dependencies could be derived according to the following inference rules:

1) if A $\succ\!\!-$ B ,C $\dashrightarrow$ A then C $\dashrightarrow$ B ,
2) A $\succ\!\!-$ B , B $\succ\!\!-$ A if and only if A $\succ\!\!\prec$ B ,
3) if A $\succ\!\!-$ B , B $\succ\!\!-$ C then A $\succ\!\!-$ C .

Here: $\succ\!\!\prec$ - the mutual specialisation dependency. Mutual specialisation dependency is useful for decomposition of goals, i.e. if$((G_2 \sqcup G_3) \succ\!\!\prec G_1)$ the $G_2$ and $G_3$is decomposition of $G_1$, i.e. components $G_2$ and $G_3$ are sufficient for refinement of goal $G_1$.

# 4 Concluding Remarks

We have presented a new approach to describe the pragmatic level objects and dependencies in terms of the semantic constraints. Information system semantics can be defined by using artefacts of the concepts, activities and actors submodels which constitute the basis of goal modelling. The concept submodel defines static constraints of the enterprise. Semantic constraints among concepts are basically expressible by artefacts of semantic data modelling [HuKi87]. The activities submodel contains the enterprise state dependencies. The behavioural constraints among actions could be defined on a basis of data flow diagrams [Yourd89] and the states of flows can be interpreted as different roles [Lind95] of concepts. Finally, the actors submodel defines the message passing dependencies [Yu94] between actors in achieving their goals. The semantics of each message is expressible in terms of concepts submodel constraints.

To simplify the process 'from fuzzy information system requirements to formal' [F3_94],[Will93], a designer should be able to deal with models of several dimensions [Pohl93] separately. The dimensions describe different aspects of requirements and correspond to the syntactic, semantic and pragmatic level [LLS94] of modelling. The relation between the process 'from fuzzy to formal' and the abstract levels of information system engineering is schematically represented by Fig.4.1.

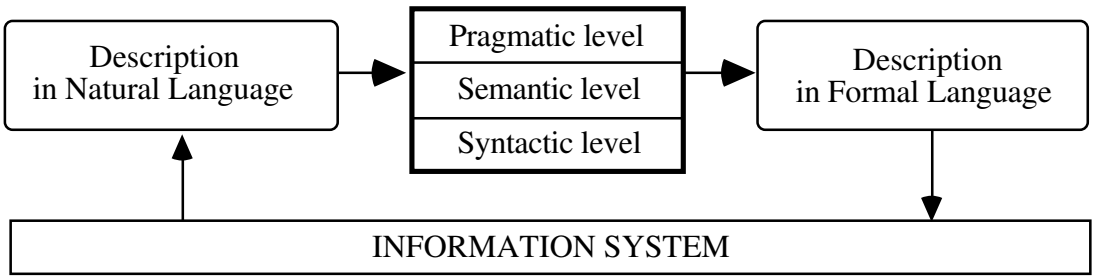


**Fig.4.1.** Three levels of information system modelling

A mixture of modelling constructs from those three different levels in the same specification language, makes the application development process very complex, because the same set of requirements can be specified across three different perspectives. So, the one of main objectives of this study is separation and identification of modelling primitives of semantic and pragmatic levels.

Primitives of syntactical level are well-understood. They are closer to implementation perspective and can be represented by constructs of programming languages. For example, relational database theory is typical representative of this level. Object oriented data [RBPEL91] models capture modelling primitives of both - semantic and syntactic levels. Perhaps, it could be a reason or good explanation of the fact, why nowadays we have no theoretical grounds in this field. Data flow diagrams

and ER kind of models are dealing with those two perspectives as well. To our opinion, it's too complicated to think about data processing and semantic aspects of the information system at the same time. This is very similar to the procedure of explanation of business objectives in terms of program modules and file structures. The relation between information system representation levels and enterprise submodels is depicted in Fig.4.2.

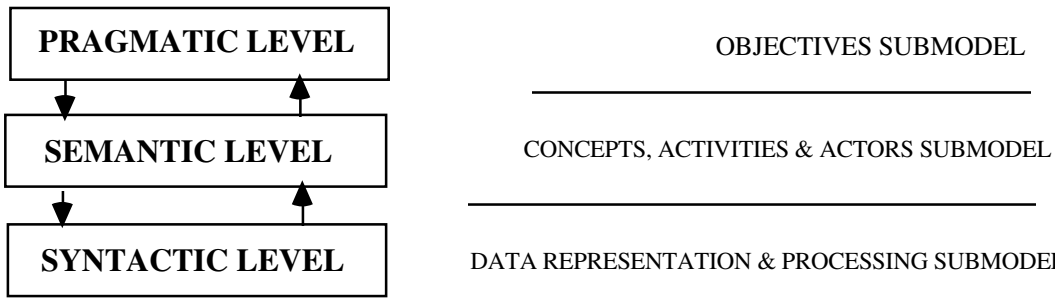


**Fig.4.2.** Three levels of enterprise modelling

The semantic level primitives are established to capture the meaning of the information system. This kind of formality let us have the interaction among models of the semantic and pragmatic level and, so, interplay among different enterprise dependencies is automatically present. The positive side of this semiotic ladder for modelling of information system requirements is that consequences of changes in one level is possible to trace into another.

The specification of requirements, needs to be analysed in order to determine whether they are ambiguous, incomplete or inconsistent. Enterprise submodels constitute a unified basis for integratability of requirements caused by information system constraints of different origins. Usually, the consequences of using of several meta-models, such as ER diagrams and data flow diagrams [Yourd89], for description static and dynamic aspects of the information system are that interaction between the specifications of different nature is lost. It means that designers becoming unable to trace the specification as a whole. Furthermore, it is impossible to define semantics of the objectives if the designer can not distinguish among the most primitive constraints of static and behavioural parts. We advocate the idea of the homogeneous basis for modelling semantic level dependencies within the enterprise. Traditionally, information system modelling techniques are based on the entity notations and several links between entities. These notations are not provided by the rules of reasoning and the capabilities of integratability of representations defined in contexts [JaPo93] of different goals. Furthermore, the theoretical foundations of the existing notations are too weak to establish such capabilities on the basis of the existing artefacts. Our approach is dealing with an interplay among semantic and pragmatic level links. It could facilitate an integratability of several 'worlds' within enterprise. There are a number of research issues we are thinking of at this stage, for instance: the need for analysis of the semantic level artefacts to establish sound and complete basis for modelling static and behavioural constraints, the principles of integratability of objectives, representing independent views of the enterprise stakeholders, and the problem of reasoning according to several semantic and pragmatic links across different perspectives.

# 5 Acknowledgements

This work was partially funded by ESPRIT Basic Research Action 6353entitled NATURE which is concerned with Novel Approaches to Theories Underlying Requirements Engineering. NATURE partners are RWTH Aachen (Germany), University of Paris I (France), City University (London, UK), FORTH Research Institute (Crete, Greece) and Swedish Institute for Systems Development (Kista, Sweden). The participation of SISU in this project is sponsored by NUTEK, Swedish National Board for Industrial andTechnical Developments.

# 6 References

[Brach79]
R Brachman. On the Epistemological Status of Semantic Networks. Associative Networks, Academic Press, pp. 3-50, 1979.

[Bub93]
J A Bubenko. Extending the Scope of Information Modelling. Fourth International Workshop on the Deductive Approach to Information Systems and Databases, Polytechnical University of Catalonia, pp. 73-97, 1993.

[Chen76]
P Chen. The Entity-Relationship Model: Towards a unified view of data. ACM Transactions on database systems 1, 1, pp. 9-36, 1976.

[DLF93]
A Dardenne, A van Lamsweerde, S Fickas. Goal Directed Requirements Acquisition. Science of Computer Programming, 20, pp. 3-50, 1993.

[F3_94]
F3 Consortium. 'F3 Reference Manual (Esprit III Project 6612)', SISU, Kista, Sweden, 1994.

[Griet82]
J J van Griethuisen (ed.). Concepts and Terminology for the Conceptual Schema and Information Base, Report ISO TC97/SC5/WG5, No 695, 1982.

[Gust94a]
R Gustas. From Conceptual to Business Modeling. Proc. of Baltic Workshop on National Infrastructure Databases -Baltic DB'94, Vilnius, Lithuania, 17-20 May, pp.218-229, 1994 .

[Gust94b]
R Gustas. Towards Understanding and Formal Definition of Conceptual Constraints. The Fourth European - Japanese seminar on Information Modeling and Knowledge Bases, May 31 - June 03, Kista, Sweden, 1994.

[JaPo93]
M Jarke, K Pohl. Establishing Visions in Context: Toward a Model of Requirement Engineering. 14th Int. Conf. on Information Systems, Orlando,USA, 1993.

[HuKi87]
R Hull, R King. Semantic Database Modelling: Survey, Applications, and Research Issues. ACM Computing Surveys, Vol.19, No.3, pp.201-260,1987.

[JoJa94]
P Johannesson, M H Jamil. Semantic Interoperability - Context, Issues, and Research Directions. The 2-nd Int. Conf. on Cooperating Information Systems, Toronto, Canada.

[Lind95]
P Lindgren. Anything, Everything and Things Playing Roles. International Conf. on Information System Concepts, Marburg, 1995.

[LLS94]
O I Lindland, G Sindre, A Solvberg. Understanding of quality in conceptual modelling. IEEE Software, pp.42-49, 1994.

[Pohl93]
K Pohl. The three Dimensions of Requirements Engineering. Int. Conf. on Advanced Information System Engineering CAiSE'93, Springer-Verlag, 1993.

[RBPEL91]
J Rumbaugh, M Blaha, W Premerlani, F Eddy, W Lorensen. Object-Oriented Modelling and Design, Prentice Hall Int., 1991.

[Simon84]
H A Simon. The Sciences of the Artificial. MIT Press, 1984.

[SoZa92]
J F Sowa, J A Zachman. Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal, 31(3), pp. 590 -616, 1992.

[Stor93]
V C Storey. Understanding Semantic Relationships. VLDB Journal, Vol.2,pp.455-487, 1993.

[Tempora92]
TEMPORA Consortium. 'Tempora (P2469) Project Manual', BIM Information Technology, Belgium, 1992.

[Yu94]
E S K Yu. Modelling Strategic Relationships for Process Reengineering. PhD theses, Dept. of Computer Sciences, University of Toronto, Canada,1994.

[Yourd89]
E Yourdon. Modern Structured Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1989.

[Wang93]
B Wangler. Contributions to Functional Requirements Modelling. PhD thesis, Dept. of Computer and Systems Sciences, Royal Institute of Technology, Stockholm, Sweden, 1993.

[Will93]
H Willars. TRIAD Modelleringshandbok: Business Modelling Overview, SISU, 1993.