

# Overviews of Models Defined with Charts of Concepts

Xavier Castellani

*CEDRIC IIE (CNAM) research laboratory  
18 allée Jean Rostand, F-91025 Evry Cedex, France  
Phone: 33 1 69 36 73 34. Fax: 33 1 69 36 73 05  
Email: [castellani@iie.cnam.fr](mailto:castellani@iie.cnam.fr)  
<http://www.iie.cnam.fr/~castellani/>*

**Keywords:** Charts of concepts, Metamodelling, Overviews of models.

**Abstract:** Models of analysis and design methods get more and more complex which makes it necessary to create overviews of these models both defined by selections of concepts and macro views of models. It is not easy to create overviews of models defined by texts and ER or OO metamodels. Hence, we propose to define models with charts of concepts which are directed graphs where nodes represent the concepts involved in a model, and arcs illustrate dependencies between definitions of its concepts. As charts of concepts are directed graphs, they easily permit when selecting concepts to see the concepts that must be selected with the chosen concepts and those that are not selected. A macro view of a model is a directed graph where nodes represent charts of concepts, and arcs illustrate macro dependencies between these charts. So, charts of concepts allow defining overviews of concepts of models both defined by selections of concepts and macro views of these models. We present charts of concepts of an overview of the UML.

## 1. INTRODUCTION

Models of analysis and design methods are more and more complex. Reasons for this greater complexity lay in the growing number of their concepts, and in the increasing interdependency between the definitions of concepts. Hence, it is very difficult for analysts, designers, and students to study all the concepts of analysis and design methods. Then, methodologists

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35500-9\\_30](https://doi.org/10.1007/978-0-387-35500-9_30)

E. D. Falkenberg et al. (eds.), *Information System Concepts: An Integrated Discipline Emerging*  
© IFIP International Federation for Information Processing 2000

and teachers have to select concepts in the models of these methods and have to present global views of these models.

That is why the question this paper intends to answer is: how to make an abstraction of models of analysis and design methods? Our proposition is to create overviews of models. We consider that an overview of a model must be both defined by a selection of concepts and a macro view of this model. A macro view of a model is a view of its sub-models (or its groups of concepts), and of relationships between its sub-models.

## **1.1 Difficulties to Create Overviews of Models**

Generally, models of analysis and design methods are defined with texts and metamodels defined with the entity-relationship model [7] or with an object-oriented model. For example, the metamodel of the UML [1] is modeled with the MOF (Meta-Object Facility) [10].

A lot of works are developed on metamodelling of analysis and design methods. Certain of these works allow comparisons of models of methods to evaluate them or to define core models of methods [11] [15]. The goal of other works is to define meta-case tools as MetaEdit [14]. Each case tool, such as GraphTalkMeta, Kogge, MetaEdit, MetaView, Mviews, ... , has its notation to define metamodels.

It is not easy to create overviews of models defined by texts and ER or OO metamodels for two reasons. First, it is difficult to select concepts in texts that define a model. It is also difficult to select concepts in a model defined with an ER or an OO metamodel because it is not easy to see concepts that are to be selected (because they are used to define the selected concepts), and those which are not selected. This difficulty results in particular from the fact that these metamodels are defined with several kinds of relationships: associations, specializations/generalizations or inheritances, aggregations, compositions. And second, it is difficult to create macro views of models defined by texts and ER or OO metamodels. That results from the fact that these representations do not allow defining easily views of sub-models (or groups of concepts) of models, and of relationships between sub-models.

## **1.2 Our Suggestion: Create Overviews of Models Defined with Charts of Concepts**

As it is difficult to create overviews of models defined by texts and ER or OO metamodels, we advocate to create overviews of models starting from simple graphical representations of models. These representations must

allow easily selections of concepts in models, and easily representations of macro views of models.

The simpler graphical representations of models are directed graphs such as each node represents a concept, and only one kind of relationship is defined between concepts. We call these relationships *definitional dependencies*. A definitional dependency is defined from a concept  $CP_x$  to a concept  $CP_y$  if  $CP_x$  is used in the definition of  $CP_y$ . We call *charts of concepts* these graphical representations of models [2] [3].

As charts of concepts are directed graphs, they easily permit when selecting concepts to see the concepts that must be selected with the chosen concepts and those that are not selected.

Macro views of models are also simple to be defined with directed graphs where nodes represent charts of concepts, and arcs illustrate macro dependencies between these charts.

As example, we present an overview of the version 1.1 of the model of the Unified Modelling Language (UML) [1]. We do not recall definitions of concepts of this model. The justifications of the definitional dependencies between concepts result from these definitions. These justifications are not specified in this paper for sake of conciseness. The reader could refer to definitions of concepts in [1].

### **1.3 Outline of this Paper**

This paper is organized as follows. Charts of concepts of a model are defined in Section 2. Overviews of models defined with charts of concepts are described in Section 3; an overview of the UML is presented in this section (charts of concepts of this overview are in Appendix). A brief discussion on the interest of charts of concepts to create overviews of models is in the conclusion.

## **2. CHARTS OF CONCEPTS OF A MODEL**

A definition of a concept of a model is given in Section 2.1. Definitional dependencies between concepts are specified in Section 2.2. The definition of a chart of concepts is given in Section 2.3. Precision on the notation used to represent charts of concepts is in Section 2.4.

## 2.1 What is a Concept of a Model?

For us, *a concept of a model is anything in this model that can be used by software analysts and/or designers to represent or to specify phenomena of the real world or technical phenomena*. A concept is identified by a unique name; there cannot be two concepts in a chart with the same name.

It is not easy to find concepts of a model. It is a classical problem whatever the model used to represent a metamodel, ER or OO for example.

It is the case for the model of the UML. The model of the UML is presented with texts and an OO metamodel. All concepts, or supposed to be, are not in this metamodel. We have studied glossaries and indexes of these documents to decide if notions represent or not concepts. We have also used our culture on the object-oriented approach to take decisions.

Examples taken in the UML:

We have taken into account as concepts notions of "Action associated to an entry transition" and "Action associated to an exit transition" used in statechart diagrams (of which the chart is in Figure 23 in Appendix), because these two notions are used to define the notion of "Point of execution of an operation". This last notion has also be considered as a concept.

We have not taken into account as concepts notions of "Cyclic activity" and "Sequential activity" in statechart diagrams by considering the little importance granted in the documents to these two notions.

## 2.2 Definitional Dependencies Defined between Concepts of a Model

The notation of charts of concepts settles on definitional dependencies between concepts.

### *Definitional Dependencies Between two Concepts*

*We create a definitional dependency from a concept CPx to a concept CPy if CPx is used in the definition of CPy*. A such definitional dependency means that concept CPx must be presented before concept CPy.

Example: in Figure 1 is depicted the definitional dependency that exists from concept "Object" to concept "Message" of the basic concepts relative to objects of the model of the UML (the chart of the basic concepts relative to objects is in Figure 15 in Appendix).

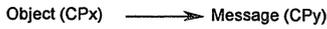


Figure 1. Example of definitional dependency between two concepts

*If there exists a definitional dependency from a concept CPx to a concept CPy and an inverse dependency from CPy to Cpx then a co-definitional dependency exists between these two concepts.*

Example: in Figure 2 is depicted a co-definitional dependency that exists between concept "Conditional branching" and concept "Branching condition" of the sequence diagrams of the model of the UML (the chart of the sequence diagrams is in Figure 22 in Appendix).



Figure 2. Example of co-definitional dependency between two concepts

If a co-definitional dependency exists between two concepts, then a circuit of definitional dependencies exists between these two concepts.

*Non Representation of Definitional Dependencies Resulting from Transitivity of Definitional Dependencies*

If there exists a definitional dependency from a concept CPx to a concept CPy, and a definitional dependency from CPy to a concept CPz, then there is a transitive definitional dependency from CPx to CPz. To simplify charts of concepts we do not represent the definitional dependency that exists from CPx to CPz.

Example: in Figure 3 is depicted definitional dependencies that exist between three concepts of the basic concepts relative to objects of the model of the UML; the definitional dependency between the first one and the third one is not represented in the chart of the basic concepts relative to objects (this chart is in Figure 15 in Appendix).

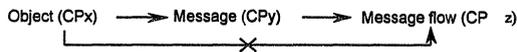


Figure 3. Example of non representation of a definitional dependency resulting from transitivity of definitional dependencies

*Logical Expressions Between Definitional Dependencies*

If definitional dependencies exist respectively from a concept CPx to concepts CPy, CPz, ... , that means that CPx has been defined before CPy and CPz and ... These and ... are not represented in the charts of concepts.

Example taken in the chart of basic concepts relative to objects of the UML (chart which is in Figure 15 in Appendix): the graph in Figure 4 represents the fact that concept "Object" is used in the definitions of concepts "Key", "Object persistence", "Object flow", and "Message". Then, the concept "Object" must be presented before concepts "Key", "Object persistence", "Object flow", and "Message".

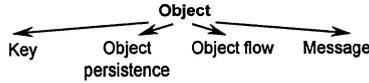


Figure 4. Example of definitional dependencies starting from a concept

If several definitional dependencies end to a same concept CPy, a logical expression mentioned on the graph specifies how concept CPy depends on the concepts beginning these definitional dependencies.

Example of logical expression defined with a simple *and* taken in the chart of concepts of the class diagrams of the UML (which is in Figure 20 in Appendix): concept "Sub-set constraint" depends on concept "Constraint", and on concept "Role".

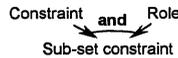


Figure 5. Example of definitional dependencies ending to a concept with an *and*

Example of logical expression defined with a simple *or* taken in the chart of concepts of the class diagrams of the UML (which is in Figure 20 in Appendix): concept "Constraint on an association" depends on concept "Sub-set constraint", or on concept "Constraint or".

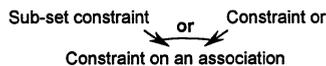


Figure 6. Example of definitional dependencies ending to a concept with an *or*

Example of logical expression defined with an *and* and an *or* taken in the chart of concepts of the sequence diagrams of the UML (which is in Figure 22 in Appendix): concept "Arrow of a sequence diagram" depends on concept "Message flow" or (on concept "Event", and on concept "Use case").

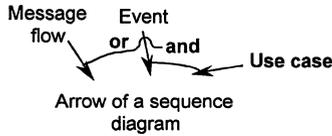


Figure 7. Example of definitional dependencies ending to a concept with an *and/or*

*Root Concepts*

A root concept of a model is a concept that is not defined using any other concept of this model. A root concept of a model has no ascendant in the charts of concepts of this model. A root concept is denoted by a small black triangle.

Example: in Figure 8 is depicted the concept "Identity" of the basic concepts relative to objects of the model of the UML, concept which is a root concept of this model (the chart of the basic concepts relative to objects is in Figure 15 in Appendix).



Figure 8. Example of root concept

*Terminal Concepts*

A terminal concept of a model is a concept such that no other concept definition of this model makes use of it. A terminal concept of a model has no descendant in the charts of concepts of this model. A terminal concept is denoted by a small black circle.

Example: in Figure 9 is depicted the concept "Key" of the basic concepts relative to objects of the model of the UML, concept which is a terminal concept of this model (the chart of the basic concepts relative to objects is in Figure 15 in Appendix).



Figure 9. Example of terminal concept

**2.3 Definition of a Chart of Concepts. Precision on Charts of Concepts**

*Definition of a Chart of Concepts*

A chart of concepts of a model is a directed graph where nodes represent the concepts involved in the model, and where arcs illustrate the definitional dependencies resulting from the definitions of the concepts.

*Examples: Charts of Concepts of the Version 1.1 of the UML*

Authors of the UML present concepts of its model in two categories of groups of concepts: basic concepts, and concepts which allow defining 9 types of diagrams. We defined charts of concepts for each of these groups of concepts.

. The basic concepts of the model of the UML are: basic concepts relative to objects, basic concepts relative to classes, auxiliary elements, extension mechanisms, and data types. Selected charts of concepts of these groups of basic concepts are in Appendix.

. The 9 types of diagrams of the model of the UML are: use case diagrams, class diagrams, object diagrams, collaboration diagrams, sequence diagrams, statechart diagrams, activity diagrams, component diagrams, and deployment diagrams. We do not recall definitions of these 9 types of diagrams. The reader could refer to their presentation in [1]. Selected charts of concepts of these diagrams are in Appendix.

*Representation of Concepts of a Chart of Concepts Used in Another Chart of Concepts*

Certain models are shared in sub-models or groups of concepts. It is the case of the model of the UML which is shared in parts that contain 5 groups of basic concepts and groups of concepts that allow the representation of 9 types of diagrams.

If a concept CP<sub>x</sub> defined in a chart of concepts CC<sub>i</sub> is used in a chart of concepts CC<sub>j</sub>, the name of the concept CP<sub>x</sub> is mentioned surrounded with an ellipse in the chart of concepts CC<sub>i</sub>. The three cases of surrounding or not surrounding a concept with an ellipse in a chart of concepts are explained in Figure 10.

For sake of clearness, we have not represented definitional dependencies from concepts that begin definitional dependencies and that end to concepts that belong to other charts.

If a concept of a chart is beginning of a definitional dependency that ends to a concept of another chart, this concept is not therefore terminal. So, it is not represented with the small black circle that would mean that it is terminal.

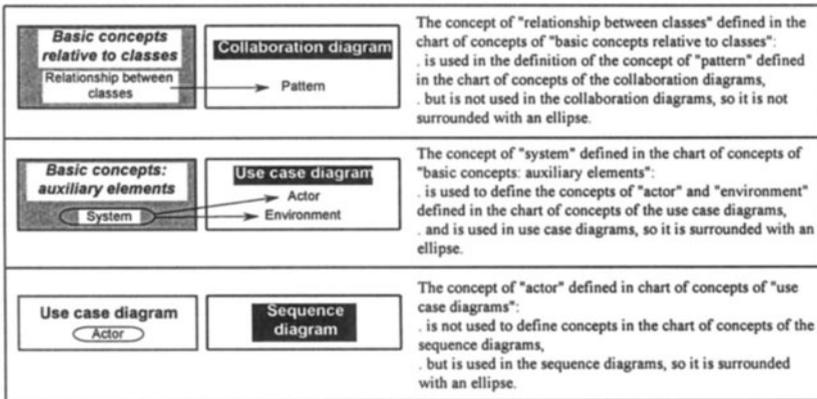


Figure 10. Examples of the three possible cases of surrounding or not surrounding a concept with an ellipse

*Circuits Between Definitions of Concepts, and Non Connected Sub-graphs*

It is very important to detect anomalies in a model. The definition of a model through text or through a ER or an OO metamodel does not permit to see easily circuits that can exist in the definitions of some of the concepts. As charts of concepts are directed graphs, such circuits are very easy to find. Moreover, charts of concepts allow finding easily independent sets of concepts represented with non connected sub-graphs.

Circuits Between Definitions of Concepts. A circuit exists between concepts CPx, CPy, ..., CPz, if concept CPx is used in the definition of concept CPy, concept CPy is used in the definition of concept CPz, ... , and concept CPz is used in the definition of concept CPx.

The existence of a circuit between concepts of a model relates a default in definitions of these concepts because the presentation of one of these concepts necessitates the presentation of all concepts of this circuit. Then, the selection of one of the concepts of a circuit necessitates the selection of all concepts of this circuit.

Let recall that co-definitional dependencies define circuits between two concepts.

Examples: we found several circuits between definitions of concepts of the UML model; one of them is in the chart of basic concepts relative to objects which is in Figure 15 in Appendix; this circuit is in Figure 11.

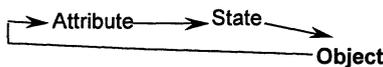


Figure 11. Example of circuit between concepts

Non Connected Sub-graphs in Charts of Concepts. Concepts of non connected sub-graphs between concepts of a model can be presented apart from the other concepts. Such concepts are not sufficiently integrated in a model.

Selections of concepts of a non connected sub-graph of a model are not concerned by selections of concepts that do not belong to this non connected sub-graph.

Examples: in charts of concepts of the UML, we found four non connected sub-graphs in the chart of concepts of sequence diagrams which is in Figure 22 in Appendix; one of them is in Figure 12.

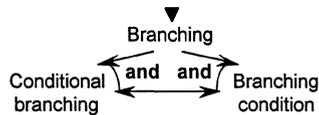


Figure 12. Example of non connected sub-graph

## 2.4 Precision on the Notation Used to Define Charts of Concepts

The notation used to represent charts of concepts is slightly different from the one of Sowa's conceptual graphs [13]. Conceptual graphs allow the use of different kinds of conceptual relationships. On the contrary, all the arcs used in charts of concepts have the same signification which is "*a definitional dependency is defined from a concept CPx to a concept CPy if CPx is used in the definition of CPy*".

We do not consider that it would be interesting to permit the use of different kinds of definitional dependencies in metamodels defined with charts of concepts for two main reasons. First, it would be more difficult to create charts of concepts. And second, metamodels defined with charts of concepts would be more difficult to be used to verify models, to select coherent sub-sets in models, to create customized versions of models, to gradual learn models, to evaluate models, and to create overviews of models.

As charts of concepts are defined with only one kind of arcs, they can be considered as particular conceptual graphs. But with charts of concepts we can define logical expressions between definitional dependencies (cf. Section 2.2).

Many operations can be defined on conceptual graphs: copy, restriction, join, projection, specialization, generalization, simplification. With charts of concepts we can use an other operation: selection of parts of a model in its charts of concepts.

In the field of conceptual modeling, several authors have proposed relationships between concepts to build conceptual schemes. For example, Kangassalo [12], proposes a system and a methodology for conceptual modeling and information construction called COMIC. In this model, the basic epistemological relation between concepts is a relation of *"intentional containment"*. An intuitive explication can be given by saying that *"a concept CPy contains intentionally a concept CPx if the knowledge that forms the concept CPy contains the knowledge that forms the concept CPx"*.

In our charts of concepts, definitional dependencies are created from the definitions of the concepts. Therefore, we reckon that charts of concepts are more easily usable than COMIC diagrams to create representations of existing models.

Moreover, Kangassalo considers that in principle there is an infinite number of epistemological relations, and he mentions that the most commonly used are the aggregation, the generalization, and the value transformation.

We above explain how we consider that charts of concepts of metamodels are defined with only one kind of arcs. So, we do not consider that it would be interesting to specialize definitional dependencies in metamodels defined with charts of concepts.

It has to be pointed out that charts of concepts should not be confused with types lattices: a concept depending on another concept is not necessarily "A Kind Of" that concept, or does not inherit from that concept. So, the definitional dependencies have not the same power of semantics that these kinds of relationships because they are only defined by taking into account the fact that concepts are defined with others. A complementary study of the semantics of a definitional dependency can conclude to define "A Kind Of" relationship or an inheritance between the two concepts concerned.

The definitional dependencies are defined between concepts without taking into account possible values of these concepts. So, definitional dependencies should not be confused with dependencies such as associations defined with roles, aggregations, compositions, or dependencies from the relational model such as functional dependencies, multivalued dependencies, or inclusion dependencies [8] [9]. A complementary study of the possible values of two concepts linked with a definitional dependency can conclude to define one of these dependencies.

### 3. OVERVIEWS OF MODELS DEFINED WITH CHARTS OF CONCEPTS

Recall that we consider that an overview of a model is both defined by a selection of concepts and a macro view of this model. Selection of concepts in a model is presented in Section 3.1. The definition of a macro view of a model is presented in Section 3.2.

#### 3.1 Selection of Concepts in a Model Defined by Charts of Concepts

*A selection of concepts in a model defined with charts of concepts is an extraction of concepts in this model to create a restricted version of this model in which:*

- . *the selected concepts are selected with their ascendants,*
- . *a non selection of concepts implies the non selection of their descendants.*

The mechanism of selection of concepts in a model is the following one. We associate to each concept  $CPy$ :

- . a logical choice variable  $Cy$ : the value "true" means that the manager of software analysts and designers wants to select concept  $CPy$ ;
- . a logical function  $Fy$  of which arguments represent the selection variables of the antecedents of  $CPy$ , arguments joined by logical operators.

The logical selection variable of the concept  $CPy$  is:  $Sy = Cy$  and  $Fy$ ; the concept  $CPy$  is selected if  $Sy$  is true.

The evaluation of the logical function  $Fy$  associated to the concept  $CPy$  depends on values of selection variables of its antecedents, which depend on values of selection variables of their antecedents, etc.

To select several concepts the selection mechanism above-mentioned to select one concept is repeated, and the selected concepts are marked.

#### *Example: Selection of Concepts to Define an Overview of the UML*

First of all we have to fix a goal and criteria to select concepts in the model of the UML; examples:

- . goal: to initiate professionals working in computer science that are not familiarized with the OO approach in the UML.;
- . selection criteria:
  - . to only consider object-oriented concepts;
  - . to take an interest to classes instead of objects, to conceptual and logical levels more than to the physical level, to static representations instead of dynamic representations.

Selection of concepts in the charts of the basic concepts of the UML. According to our goal and to our criteria to select concepts in the UML we obtain the selection of concepts in the charts of basic concepts in Figure 15

to 18 in Appendix. As no concept of extension mechanisms are selected, we do not select the chart of this group of concepts.

Selection of concepts in the charts of concepts of the 9 types of diagrams of the UML. According to our goal and to our criteria to select concepts in the UML we do not select concepts of the object diagrams, the activity diagrams, the component diagrams, and the deployment diagrams. Charts of concepts of the selected diagrams: use case diagrams, class diagrams, collaboration diagrams, sequence diagrams, and statechart diagrams, are in Figures 19 to 23 in Appendix.

Table 1 summarizes the numbers of concepts defined in charts of concepts of the complete model of the UML version 1.1, and of our selection of concepts. The number of concepts of the complete model is high (233). This number justifies the necessity to select concepts to present an overview of this model. The selected concepts of the proposed overview of this model contains 71 concepts. This number is reasonable for an overview. With our academic experience we consider that this overview can be presented in three or four days.

	Complete model	Concepts of the overview of the UML
<b>Basic concepts</b>		
Chart of basic concepts relative to objects	24	11
Chart of basic concepts relative to classes	8	7
Chart of basic concepts: auxiliary elements	12	3
Chart of basic concepts: extension mechanisms	8	0
Chart of basic concepts: data types	32	18
<b>Total number of basic concepts</b>	<b>84</b>	<b>39</b>
<b>Diagrams</b>		
Chart of concepts of use case diagrams	7	6
Chart of concepts of class diagrams	42	17
Chart of concepts of object diagrams	7	0
Chart of concepts of collaboration diagrams	9	3
Chart of concepts of sequence diagrams	23	2
Chart of concepts of statechart diagrams	33	4
Chart of concepts of activity diagrams	6	0
Chart of concepts of component diagrams	11	0
Chart of concepts of deployment diagrams	11	0
<b>Total number of concepts of charts of diagrams</b>	<b>149</b>	<b>32</b>
<b>Numbers of concepts</b>	<b>233</b>	<b>71</b>

Table 1. Numbers of concepts of the model of the UML version 1.1 and of the overview of this model presented in this paper

### 3.2 Macro View of a Model Defined with Charts of Concepts

We propose to represent *a macro view of a model with a directed graph where nodes represent the selected charts of concepts of this model, and where arcs illustrate the macro dependencies resulting from concepts of charts used to define concepts of other charts*. Macro dependencies between charts of concepts are presented hereafter. An example of macro view of the UML is given hereafter.

#### *Macro Dependencies between Charts of Concepts of a Model*

*There exists a macro dependency from a chart  $CC_i$  to a chart  $CC_j$  if at least one concept of the chart  $CC_i$  is used to define a concept of the chart  $CC_j$ .*

The fact that a macro dependency exists from a chart of concepts  $CC_i$  to a chart of concepts  $CC_j$  means that the chart  $CC_i$  must be presented before the chart  $CC_j$ .

We consider that a chart of concepts is a particular concept. So, the logical expressions between definitional dependencies, the non representation of definitional dependencies resulting from transitivity of definitional dependencies, and the root and the terminal concepts, can be applied to macro dependencies between charts of concepts.

Example: in Figure 13 is depicted the macro dependency that exists from the chart of concepts of collaboration diagrams to the chart of concepts of sequence diagrams of the UML.

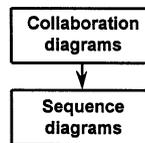


Figure 13. Example of macro dependency between two charts of concepts

#### *Example of Macro View of Charts of Concepts of the UML*

Example: according to our goal and to our criteria to define a selection of concepts in the UML (cf. Section 3.1), the macro view of the selected charts of concepts of this model is defined with the groups of concepts of which names are not hatched in Figure 14. Among the non selected groups of concepts are the object diagrams and the activity diagrams. Note that these two diagrams cannot be represented with the version 4 of the UML case tool Rose commercialized by Rational Software.

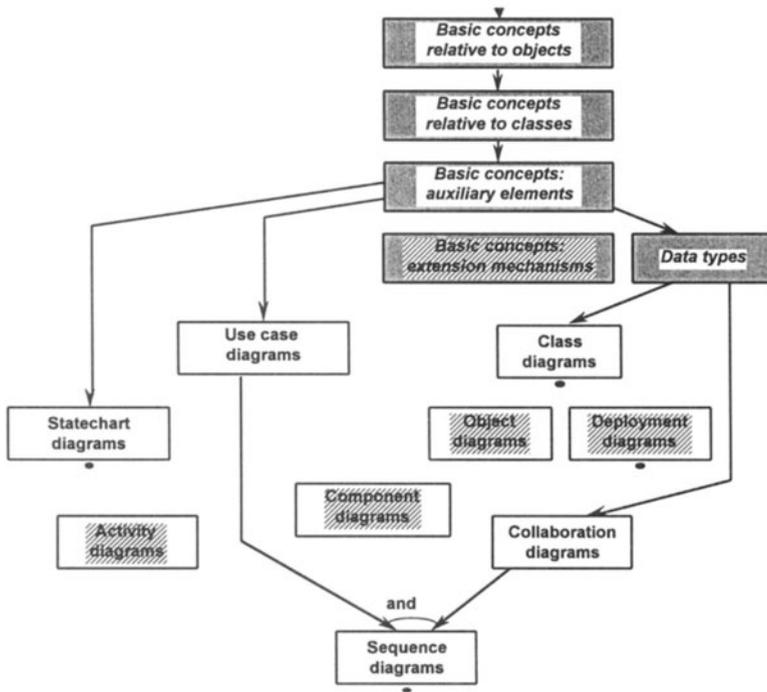


Figure 14. A macro view of charts of concepts of the UML model

## 4. CONCLUSION

We explain in the introduction why ER or OO metamodels do not easily allow creation of overviews of models both defined by selections of concepts and macro views of these models. We show in this paper how charts of concepts allow creation of overviews of models. As charts of concepts are directed graphs, they easily permit when selecting concepts to see the concepts that must be selected with the chosen concepts and those that are not selected. Macro views of models are simple to be defined with directed graphs where nodes represent the selected charts themselves, and arcs illustrate macro dependencies defined between them.

As example we present an overview of the UML. This selection of concepts in this model contains 71 concepts. It was easily obtained starting from the complete charts of concepts of the UML.

It is easy to delete concepts in a selection of concepts of a model by seeing the concepts that are deleted with the chosen concepts. It is also easy to add concepts in a selection of concepts of a model by seeing the concepts that must be selected with the new concepts.

This paper underlines the use of charts of concepts to create overviews of models. A work in progress is developed to map metamodels defined with an object-oriented model onto charts of concepts. Then it would be possible to map OO metamodels onto charts of concepts to create overviews of models.

The possibility to define selections of concepts in models is exposed in [3] with presentation of the version 1.0 of the UML. This application is developed and extended in [5], paper in which a core of the version 1.1 of the UML is presented. Charts of concepts allow creation of customized models, by easily selecting concepts, modifying definitions of concepts, and adding concepts in models because they are directed graphs [6].

For the same reason, charts of concepts easily allow evaluation of models with metrics usable to measure sizes, complexities, and anomalies in models such as circuits between concepts and non connected sub-graphs [4].

To summarize, the principal uses of charts of concepts are: verification of models, selection of coherent sub-sets in models, creation of customized versions of models, definition of overviews of models, and evaluation of models. So, the charts of concepts are a general proposal in the field of the method engineering [6].

That is why we think that charts of concepts would be a good tool to represent the information systems concepts of the FRISCO framework (anonymous ftp://ftp.leidenuniv.nl/pub/rul/fri-full.zip).

## ACKNOWLEDGMENTS

I would like to thank Vincent Bataille, PhD student, and Hong Jiang, PhD in computer science, for their valuable remarks on this paper.

## REFERENCES

1. Booch G., Jacobson I., Rumbaugh J., "Unified Modeling Language (UML), Version 1.1", Rational Software Corporation, Santa Clara, 1 September 1997.
2. Castellani X. "Méthodologie générale d'analyse et de conception des systèmes d'objets (MCO). L'ingénierie des besoins". Masson Editor, 1993.
3. Castellani X., *Cartes de concepts permettant de créer des versions personnalisées d'UML*. Journal Ingénierie des Systèmes d'Information (ISI), Volume 5, n° 5/1997, September 1997, Hermès.

4. Castellani X., *Evaluation of Models Defined with Charts of Concepts: Application to the UML Model*. CAISE'98/IFIP 8.1, Third International Workshop on Evaluation of Modelling Methods in Systems Analysis and Design (EMMSAD'98), Pisa, Italy, 8-9 June 1998.
5. Castellani X., Hamamouche N., *Definition of Customized Models: Application to Define a Core of the UML Model*. Tenth International Conference on Software Engineering and Knowledge Engineering (SEKE '98), San Francisco, USA, 18-20 June 1998.
6. Castellani X., Hamamouche N., *Model Engineering using Charts of Concepts: Application to the UML Model*, International Conference on Object-Oriented Information Systems (OOIS' 98), Paris, 9-11 September 1998, proceedings published by Springer-Verlag.
7. Chen P.P., *The entity-relationship model*, ACM transactions on database systems, Vol. 1, No. 1 March 1976.
8. Codd E.F., *A relational model of data for large shared data banks*, ACM, Vol. 13, No. 6, June 1970.
9. Codd E.F., "Further normalization of the data base relational model", Data base systems, Prentice Hall, Englewood Cliffs, New Jersey, 1972.
10. DSTC & CO-SUBMITTERS, Meta-Object Facility (MOF), Object Management Group, Inc (OMG), Framingham, USA - ad/97-08-14 - 1997.
11. Henderson-Sellers B. and Firesmith G., *COMMA: Proposed core model*, Report on Object Analysis and Design, Vol. 4, No. 1, 1997.
12. Kangassalo H., COMIC: A system and methodology for conceptual modelling and information construction, Data & Knowledge Engineering, Vol. 9, No. 1, October 1992.
13. Sowa J.F., "Conceptual structures: Information processing in Mind and Machine", Addison-Wesley Publishing Company, 1984.
14. Tolvanen J.P., Marttiin P., Lyytinen K., Rossi M., Tahvanainen V.P. and Smolander K., *Modelling Requirements for Future CASE: Modelling Issues and Architectural Consideration*, Information Resources Management Journal, Vol. 8, No. 1, 1995.
15. Welke R.J., Kumar K., *Method engineering: a proposal for situation-specific methodology construction, in Systems Analysis and Design: A Research Agenda*, Cotterman and Senn(eds), Wiley, 1992.

## APPENDIX: CHARTS OF CONCEPTS OF THE PROPOSED OVERVIEW OF THE UML

Charts of selected basic concepts and of selected diagrams of the proposed overview of the UML model are in this appendix:

- . charts of selected basic concepts: basic concepts relative to objects, basic concepts relative to classes, auxiliary elements, data types (Figures 15 to 18);
- . charts of concepts of selected diagrams: use case diagrams, class diagrams, collaboration diagrams, sequence diagrams, statechart diagrams (Figures 19 to 23).

The names of the non selected concepts are hatched in these charts of concepts.

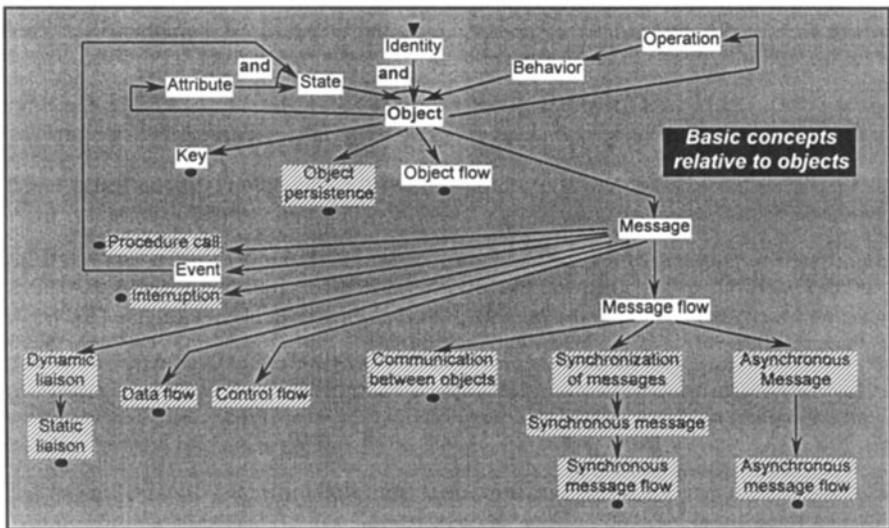


Figure 15. Chart of basic concepts relative to objects

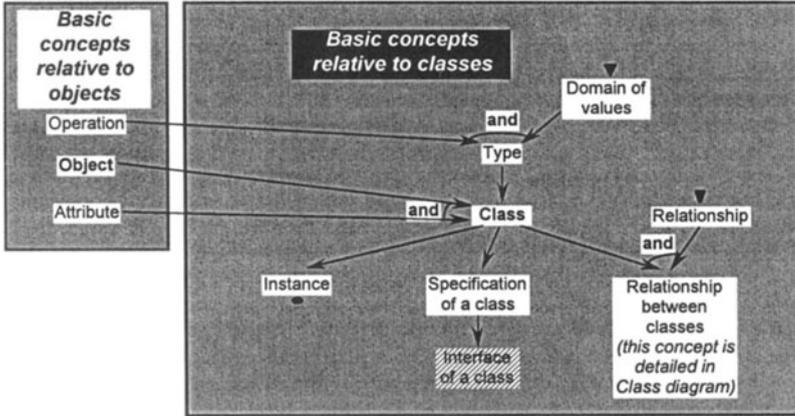


Figure 16. Chart of basic concepts relative to classes

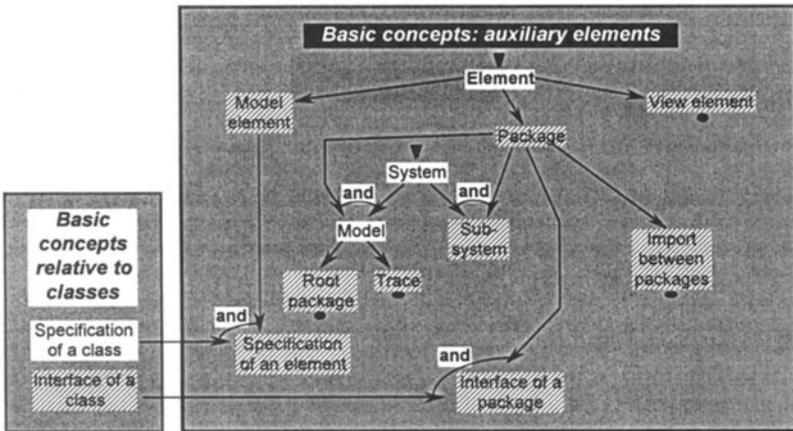


Figure 17. Chart of basic concepts: auxiliary elements

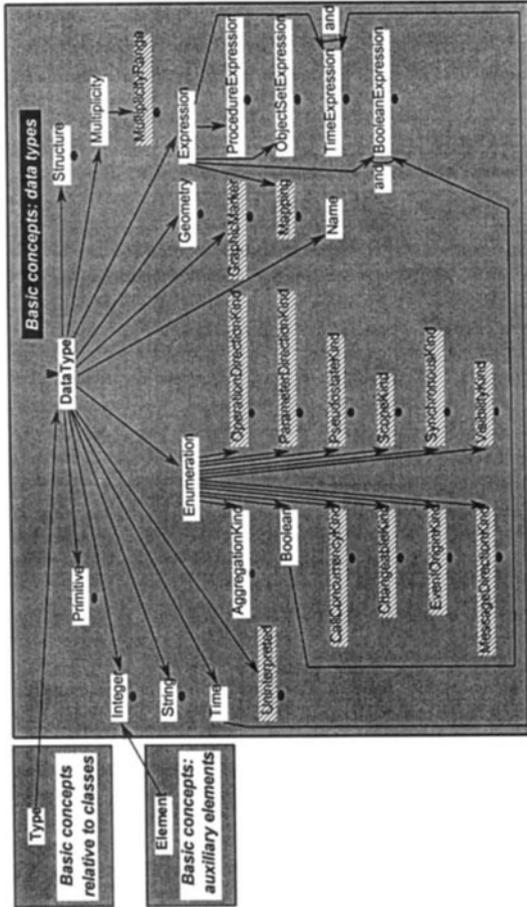


Figure 18. Chart of basic concepts: data types

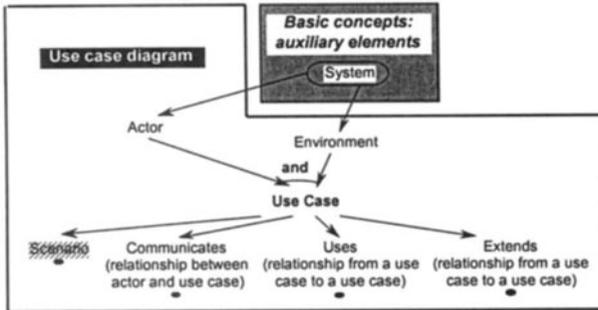


Figure 19. Chart of concepts of use case diagrams

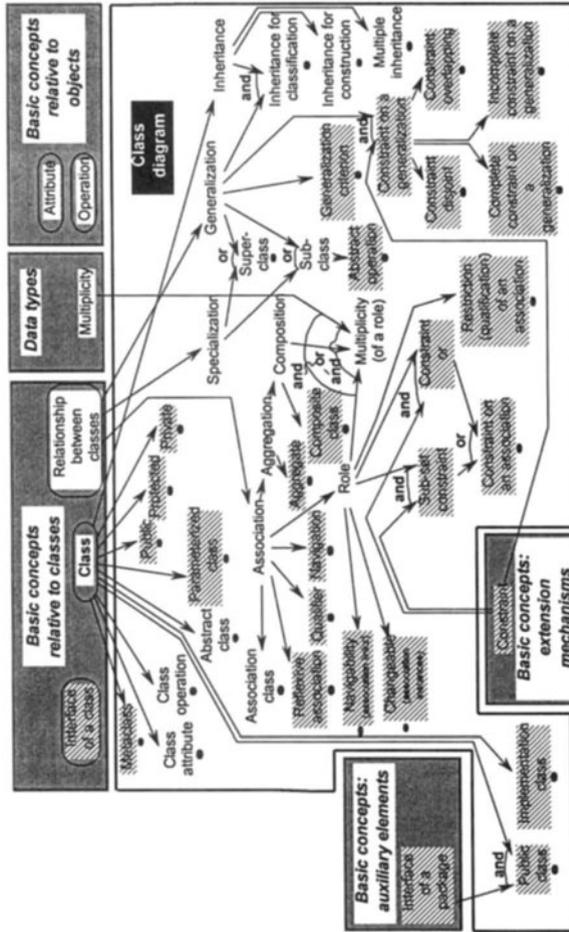


Figure 20. Chart of concepts of class diagrams

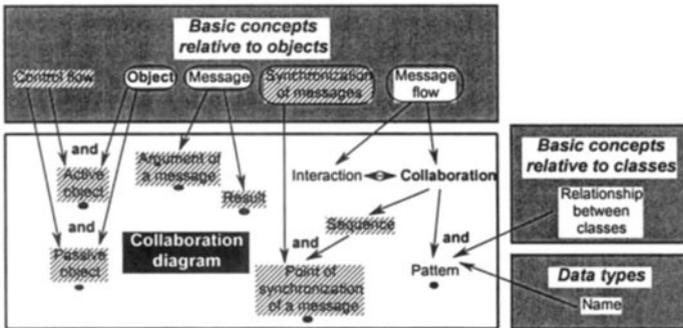


Figure 21. Chart of concepts of collaboration diagrams

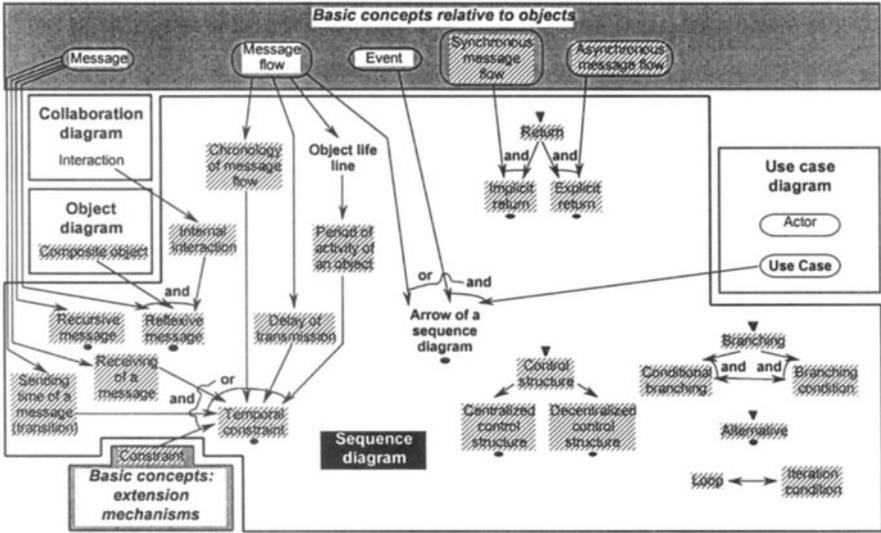


Figure 22. Chart of concepts of sequence diagrams

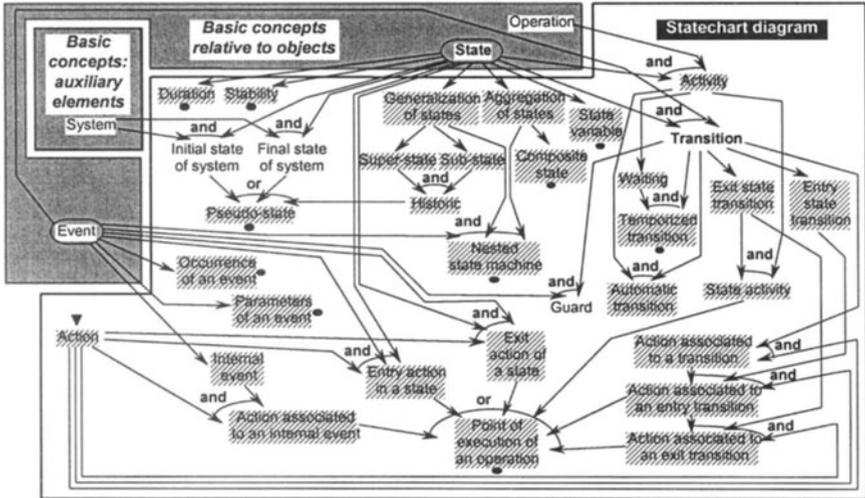


Figure 23. Chart of concepts of statechart diagrams