Bottom-up Modelling with CIMOSA

W. Reithofer
Institute for Real-Time Computer Systems and Robotics,
University of Karlsruhe, D-76128 Karlsruhe, Germany
e-mail: rhofer@ira.uka.de

Abstract

CIMOSA enterprise models provide a basis for documenting, operating, and redesigning a company. However, up to now CIMOSA is not commonly accepted in industries. This is mainly due to the fact that CIMOSA users do not know how to create their particular models efficiently. Currently, they are faced with the collection of a tremendous amount of data which makes the model creation process an unaffordable effort. This paper proposes a new bottom-up modelling methodology for the design of particular CIMOSA models: The modelling starts with the identification of the most important process according to a specific goal. This core model is then expanded from inside to outside by adding further processes which are also relevant. With respect to the implementation of a corresponding modelling tool, the basic data structures for the model representation and the procedures which are needed to process them are also shown. Finally, our modelling and simulation prototype tool, the Virtual Factory Lab Victor is introduced.

Keywords

CIMOSA, Enterprise Modelling, Simulation

1 INTRODUCTION

Two of the main problems of today's enterprises are:

- **Enterprise redesign:** To replace the classical, function-oriented processes by product-oriented, computer integrated processes, enterprises are redesigned. The starting point of such an project is usually a thorough investigation of the „as-is" situation. The „as-is" processes are analyzed and redesigned to remove inconsistencies and to make them as straightforward as possible. Afterwards, the resulting „to-be" processes must be implemented in the physical system (see, e.g., [McMenamin, 1988]).

J. Goossenaerts et al. (eds.), *Information Infrastructure Systems for Manufacturing*

- **(Re-)Design of virtual enterprises:** A *Virtual Enterprise* [Rembold, 1994], [Goldman 1995] or *Extended Enterprise* [Browne, 1994] is a temporary alliance of several distributed work units. These units cooperate to manufacture particular products to meet new market needs very rapidly. The work units may be units of one or several enterprises. One individual enterprise may be involved in several virtual enterprises with different partners at the same time. The most important benefit of a virtual enterprise is the accumulation of core competencies of the involved partners and the reduction of time-to-market by optimally using existing production resources. Virtual enterprises already exist, but their formation, operation, reformation or dissolution is currently very costly in terms of management and administration. The (re-)design of a virtual enterprise is mainly concerned with the problem of assembling a consortium, splitting up the common task into subtasks and assigning the subtasks to the partners.

A model which could help to solve these problems, must represent the most important aspects of an enterprise like objects, functions and processes. Furthermore, particular models of individual enterprises should have the same structure to be comparable and reusable. This can be achieved by deriving individual models from a generic meta model.

Several existing enterprise meta models or reference architectures fulfill these requirements. A comparison of them can be found in [CEN, 1994]. Common to most enterprise models is that they integrate basic models for describing the various aspects, e.g., SADT for describing functions, entity attribute representations for describing objects etc. We work with the Open Systems Architecture for Computer Integrated Manufacturing Systems CIMOSA [AMICE 1994], [Formal Reference Base, 1994]. CIMOSA is based on a formal, properly defined model representation. Most parts of CIMOSA have become a standard (ENV 40.003, Framework for Enterprise Modelling). However, one has to admit that until now CIMOSA has not been generally accepted in industries. When designing an individual model, companies are faced with an immense effort which they cannot fulfill within an acceptable time span. Of course, they do not want to spend a lot of resources to develop a model which is already out dated before it is finished. The reasons for this huge modelling effort are the weak description of the modelling process, and the lack of tools which support the model creation and model validation.

2 REVIEW OF EXISTING WORK

2.1 The CIMOSA business modelling process

Apart from the model itself, the [Formal Reference Base, 1994] describes the CIMOSA model creation process. This description explains how to create a particular model from the reference architecture. The modelling process is decomposed into the three subprocesses *Instantiation, Derivation* and *Generation*. The *Instantiation* process expresses the relations between the three levels of genericness. The *Generation* process expresses the relations between the four CIMOSA views. The CIMOSA views may be seen as a filter which focuses, e.g., on the functional aspects or the information aspects of the CIMOSA enterprise model. The relations of the three modelling levels are described by the *Derivation* process. For the CIMOSA users who want to create their particular models, the instantiation process is the most important one. The [Formal Reference Base, 1994] gives a number of rules and constraints which must be taken into account when creating particular models. However, these guidelines are completely unsuitable to help CIMOSA users to create models of their own, already existing systems.

A more concrete methodology is given in the *User Guide on Requirements Definition Model* [AMICE, 1994]. This modelling guide suggests a global top-down modelling process which starts with the identification of the domains, the definition of domain objectives and constraints, and the identifications of relationships between the domains. Afterwards, the domains are decomposed into domain processes. Domain processes are further decomposed into business processes / enterprise activities. The modelling process ends with the operational and information analysis. This course of actions has been demonstrated with an example in [Zelm, 1995]. After applying this procedure to practical examples, we found out that it is inefficient to model existing systems because of the following reasons:

- The top-down analysis cannot be strictly goal-oriented. It is burdened with the collection of irrelevant data since one cannot decide at an upper level, which, e.g., domain/business processes will be relevant for the solution of the given problem.
- The top-down analysis implies a lot of backtracking. Decisions made at an upper level can not be changed at a lower level.
- Top-down approaches are useful for system design from scratch. However, they are not recommended for the analysis of existing systems since the boundaries of the basic building blocks, i.e., the enterprise activities are already fixed.

The existing tools which support the design of particular CIMOSA models like, e.g., GTVOICE [VOICE, 1994] or SEW-OSA [Aguiar, 1995] implement this top-down modelling approach.

In summary, a modelling methodology is missing which allows CIMOSA users to create particular models in an efficient and goal-oriented manner, focusing on the processes and activities which are relevant to solve a given problem.

2.2 CIMOSA Modelling and Simulation Tools

Before using a particular model for problem solving, it has to be proven that the relevant aspects of the system are represented correctly. Simulation is suitable to support this task. The Formal Reference Base refers numerously to simulation. However, it is not described how to translate a particular CIMOSA model into a simulation model. Looking at existing CIMOSA modelling tools with simulation capabilities like, e.g., SEW-OSA [Aguiar, 1995], ARTIFEX [Bruno, 1994] and McCIM [Didic, 1993], we found out that they focus on specific applications. A problem which has not been solved by these tools is the on-line assignment, the so-called late binding, of resources to functions which are competing for the resources. Another problem is the simulation of processes where data is produced by a function and used by a subsequent function as control input.

3 A BOTTOM-UP MODELLING APPROACH

Our bottom-up modelling process starts with the design of a model of the most important process and expands this model by processes of decreasing importance. Since the modelling should be supported by a tool, we also have to take care about the data structures which represent the model and the procedures which are needed to combine the basic model elements bottom-up to more complex model components. According to the *derivation process* [AMICE, 1994], the model of a CIM system is created at three different levels of definition and expression: Each modelling level contains an internal structure consisting of three levels of genericness and can be accessed via four views. The modelling levels are the ***Requirements Definition Modelling (RDM)*** level, the ***Design Specification Modelling (DSM)*** level and the *Implementation Description Modelling (IDM)* level.

The problems described in the introduction of this paper can be mainly solved at the RDM and DSM level. The RDM level is not sufficient since information about operation times and resource capacities is needed to perform an analysis of efficiency of the various design alternatives. This information is given on the DSM level. The result of our modelling task is the selected design specification model.

3.1 The bottom-up modelling process

Our proposed bottom-up modelling process can be described roughly with the following procedure:

1. Goal definition
2. Definition of system boundaries and initial object
   3. Identification of functions applied to the initial object from its "birth" to its "death"
   4. Definition of the In/Outputs of the functions
   5. Identification of further relevant objects
   6. Completion of processes applied to further identified objects according to 3.-5.

Termination: No more relevant objects

7. Synchronize the processes and determine the triggering conditions

Each step of this procedure will now be described with an example. The example is taken from an industrial test case: A machine tool producing company has severe problems with the delivery of their products because of delays in the parts production. To find out the reasons of the delays, we performed a simulation study of the material flow [Roessel, 1995]. Apart from this conventional approach, we designed a CIMOSA model of the „as-is“ situation.

The first step of our modelling process is the definition of the goal according to the problem we want to solve. In our example, the goal is to increase the due-date dependability of the parts production. In step 2, the boundaries of the system and the initial object are defined. In our case, the initial object is a single part or a lot of single parts. The system we want to improve is the mechanical work shop for parts manufacturing. The mechanical work shop is a job shop which consists of about 20 machine tools. The production of a part starts when its raw material and its work plan is available in the input buffer of the mechanical workshop. From here the object and the work plan are transported to the input buffer of the machine where the first operation will be performed. After the execution of this operation, the machine operator places the object and the work plan in the output buffer of the machine. Next, the object and the work plan are transported again to the subsequent machine and so forth until all operations of the work plan have been executed. Afterwards, the part is buffered in the output store of the mechanical workshop. Therefore, following step 3 of our modelling process, all functions which are applied to the object between its birth and death are transport and manufacturing operations. In general, the birth of an object may be either its entering of the system (in our example the raw material) or its first appearance, i.e., the creation of the object. On the other hand, the death of an object may be its leaving from the system (example: finished parts stored in the output buffer from where they are handed over to the assembly department) or its disappearance because of, e.g., dismantling. The ordering of the functions according to their order of execution results in the initial process (Figure 1). The function transport has two ending statuses. Only if the goal of the transport operation is the output buffer of the mechanical workshop, the process terminates.

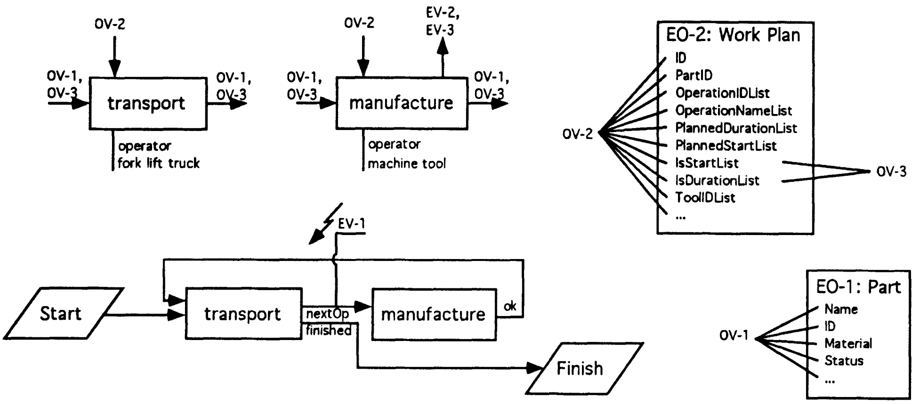

Figure 1 Part Process.

In step 4 the in/outputs of each function are defined (Figure 1). Both functions of our example manipulate the object part. Therefore, we defined an enterprise object part (EO-1) and an object view (OV-1) on this enterprise object which is used as input and as output of the functions. Additionally, both functions need a work plan. Consequently, an enterprise object „work plan“ (EO-2) and corresponding object views (OV-2, OV-3) are defined. In the following this *core model* is expanded from the inside to the outside by adding further relevant processes. When we specified the functions, we found out that a suitable tool must be also available before a manufacturing operation can be started. Since the lack of available tools may influence the due-date dependability of the parts production, we decided in step 5 that tools are relevant objects (EO-3). We also created a model of the tool process (Figure 2). A tool process starts with the birth, i.e., the assembly of a complex tool out of its basic parts and ends with the death, i.e., the dismantling of the tool. After the assembly, the tool is transported to the machine where the manufacturing operation is performed. This is done by the operator of the machine. After executing the manufacturing operation, he brings back the tool to the central tool assembly work place. The dismantling of the tool is equal to its death. For synchronizing the tools and the parts process events are needed (EV-1, EV-2, EV-3). Event EV-1 is sent out after the tool has been transported to the machine. In conjunction with the ending status nextOp of the transport operation, this event triggers the execution of a manufacturing operation. EV-2 and EV-3 are used to trigger the "use" and the "transport back to work place" functions of the tool process.

In our example, we did not add any further processes. Other objects like, e.g., NC-programs are available when they are needed. Therefore, they are not considered to be relevant objects.

The final step in the design of the "as-is" model is the definition of triggering conditions of the processes (Figure 3). The part process starts, when the raw material and the work plan is available. After finishing the part process, a signal is sent to the domain "assembly". The assembly of tools is triggered by the planning department. They send the assembly plans of the tools to the mechanical work shop independently of the work plans.

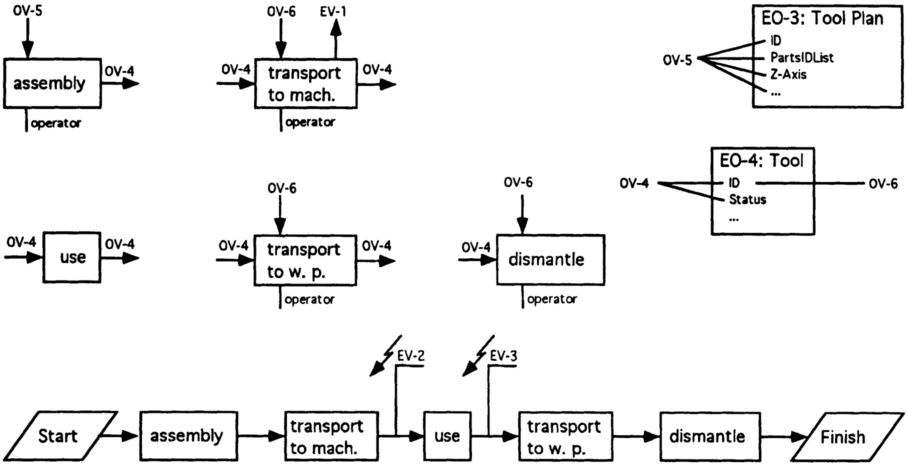

Figure 2 Tool Process.

We applied the proposed modelling methodology also to another example at the Rodenstock group [Limberger, 1996]. Here, by comparing the existing with the expected inputs of the modelled functions we discovered an error in the raw material order process. Its elimination led to a significant improvement of the raw material supply.

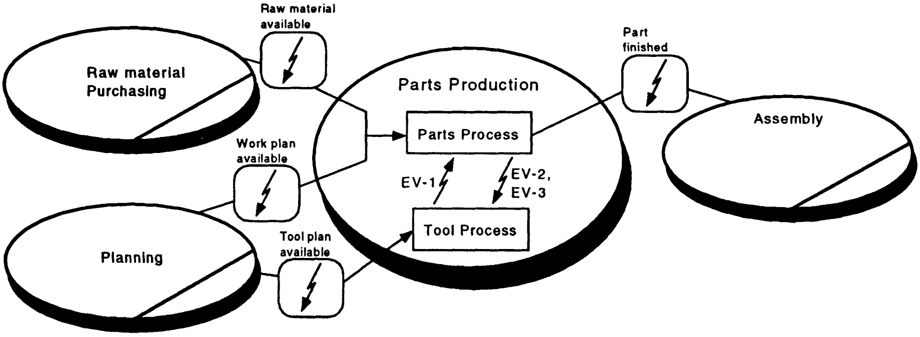

Figure 3 Triggering Conditions of the Processes.

3.2 Use of Particular Model Components (PMCs)

This section describes how we work with Particular Model Components (PMCs). The combination of PMCs is fundamental for the implementation of a tool which supports the bottom-up design of particular models. We defined five *PMC types*:

- **Enterprise Activity Type:** PMCs of this type represent the functions of an enterprise. They are the lowest level building blocks.
- **Business Process Type:** Is a sequence of Enterprise Activities / Business Processes.
- **Domain Process Type:** Represents a process which consists of Enterprise Activities and/or Business Processes.
- **Domain Type:** Represents the considered system.
- **Resource Type:** Occurrances of this PMC types actually perform the functions.

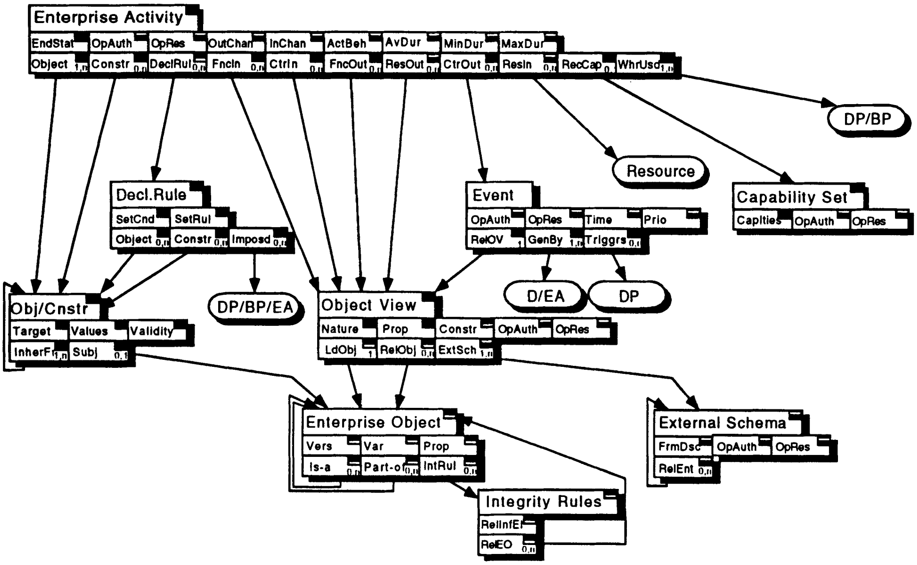

Figure 4 PMC Type Enterprise Activity.

Each PMC contains attributes and references. The attributes do not cause any problems. However, every reference is a pointer to another CIMOSA object which itself contains references to other objects as shown in Fig. 4 for a PMC of type enterprise activity.

The graph (Figure 4) contains three types of elements:

- **CIMOSA objects** (displayed with shaded polygons),
- **subgraphs** representing other PMCs (displayed with ovals), and
- **references** (displayed with arrows).

The *CIMOSA objects* are characterized by three rows. The first row gives the name of the object, the second all attributes of the object and the third all references to other objects. Every item of the second and third row is displayed within a rectangle. Two small rectangles are located on the right side above the item. The upper one indicates if the item exists (rectangle filled black) or does not exist (rectangle filled white) on the RDM level, whereas the lower one gives the same information for the DSM level. The items of the third row also contain numbers which give the cardinality of the references.

A PMC contains two types of references: *Internal references* point to an object which is a part of the considered PMC, whereas *external references* point to objects outside.

After the definition of PMC types, a set of rules is needed to define the combination of PMCs. For example, if a number of enterprise activities have been identified in the third step of our bottom-up modelling approach, they have to be ordered to form a process. This is done by instantiating a process template and updating it according to the modelled process. To allow the reuse of PMCs from one or several existing enterprise models for the, e.g., design of a model of a virtual enterprise it is necessary to implement the *export* and *import* functions for PMCs. These functions can be compared with identically named functions in a CAD system, but they are much more difficult to implement since they work on very complex structures, as shown in Figure 4. When joining PMCs, the following issues have to be taken into account:

- Unambiguous object identification: Each object of the model requires a unique identifier. Therefore, it may be necessary to change the identifiers of some objects.
- PMC completeness: To avoid the lack of referenced objects after PMCs have been combined, each PMC should be complete, i.e., it must not contain external references. External references can be avoided by either removing the reference (e.g. in case of an enterprise activity type component, the „Where used“ reference is removed) or including the referenced object in the PMC.
- Consistency: The model representation in CIMOSA is ambiguous, i.e., objects can be modelled in different ways. This fact causes problems if one wants to detect objects which should have the same semantic. For example, inserting an enterprise activity into a business process may result in references pointing to different object views which should actually describe the same object. To ensure the semantic consistency of the designed model, these doubled objects have to be compared and matched or in the case that they differ considerably from each other, the join operation must be refused. Semantic inconsistency occurs if, e.g., two enterprise activities should be connected, whereby the output of the first one is not the expected input of the following one.

3.3 Simulation of particular CIMOSA models

Simulation is a suitable way to validate the behavioral aspects of CIMOSA models. For example the model of the parts manufacturing system which has been created in section 3.1 can be validated as follows: After translating it into a simulation model it is fed in with recorded data, e.g., real work plans, tool plans, etc. of the machine tool company. Afterwards, simulation experiments are executed. The result of these experiments, i.e. the simulated start and end dates of the manufacturing operations are compared with the real system's behavior. Based on this comparison, one can decide whether the model is valid or has to be improved.

Our approach for simulating CIMOSA models is depicted in Figure 5. A pattern for each process instance in the CIMOSA model is derived. In our example we have two process patterns: The parts process and the tool process. Furthermore, every resource is represented by a corresponding instance. During run-time of the simulation, an occurrence of each process instance is created for every order. Each active process tries to progress. However, a process is delayed if the resources needed are not available. All active processes compete for the resources.

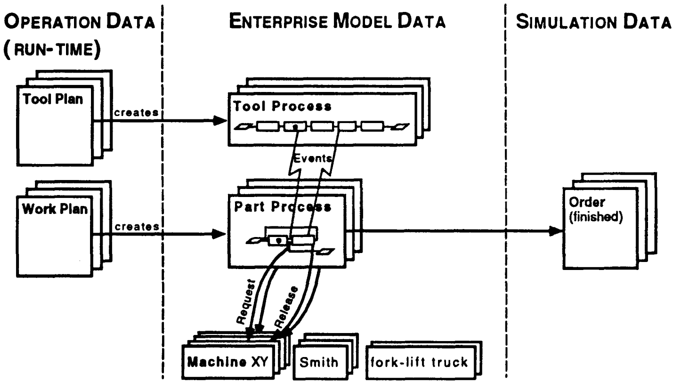

Figure 5 Simulation approach.

4 VIRTUAL FACTORY LAB KARLSRUHE

Victor consists of a modelling environment for the design of CIMOSA particular models and a simulation system for model execution (Figure 6).

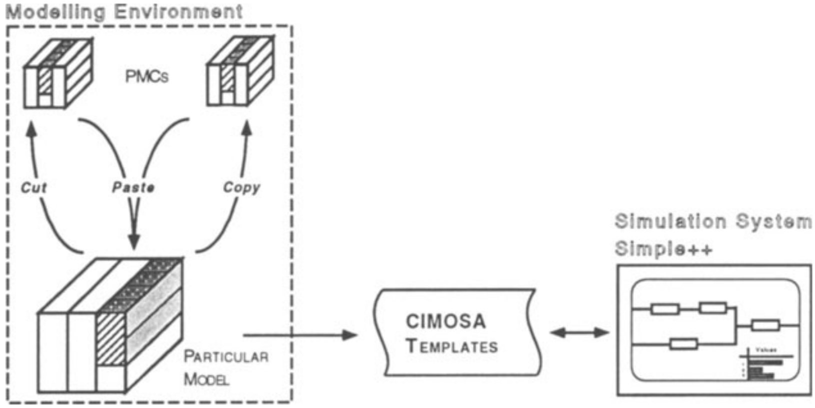

Figure 6 Victor Architecture.

It is implemented as an application specific building block library for the simulation system SIMPLE++[AESOP, 1994]. The building block library consist of building blocks for the CIMOSA objects Domain, Domain Relationship, Domain Process, Business Process, Enterprise Activity, Event, Declarative Rule, Integrity Rule, Enterprise Object, Object View, and Capability Set. We have also implemented a enterprise modelling frame (Figure 7).

For designing a particular CIMOSA model, an instance of this empty enterprise model is created. Afterwards, instances of building blocks representing the several CIMOSA objects are inserted directly into the subnets depicted with the Σs. The modelling can start anywhere. It is possible to start with the definition of domains as well as with the modelling of enterprise activities or enterprise objects. The user is not forced to follow any modelling methodology. The referential consistency of the model is guaranteed by the tool. When defining a link between to objects (e.g. „comprises", „where used"), all references are updated automatically. To make the tool as flexible as possible, we implemented methods to convert an enterprise activity into a business process and a business process into a domain process and vice versa. It is also possible to aggregate and decompose elements of a process. The model can also be accessed via the four CIMOSA views (see Figure 7, buttons at the lower left corner). The function view has two modes: ***specification*** and ***decomposition***. In specification mode, the opening of a building block allows to specify the properties of the object, i.e., to edit its template whereas in decomposition mode all objects and processes included are shown.

Particularly for the design of models of virtual enterprises and for reusing PMCs we implemented export and import functions.

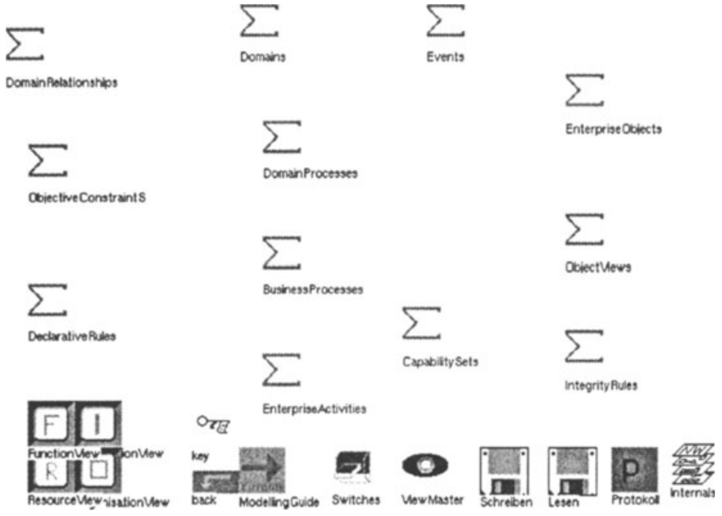

Figure 7 CIMOSA Modelling environment with Simple++.

Finally, Victor offers the possibility to transform the designed enterprise models into executable simulation models.

5 CONCLUSIONS

Although CIMOSA is one of the most powerful approaches in enterprise modelling, it is not generally accepted in industries. For them, the creation of a particular CIMOSA model is burdened with the collection of an immense amount of data. This makes the model creation process an unaffordable effort. The reason is that the CIMOSA business modelling process is not suitable to create a model in a goal-driven, efficient manner. Therefore, a new modelling methodology is proposed. The bottom-up modelling process starts with the definition of the goal. According to this goal, the system and the initial object are defined. In a third step, all functions applied to the initial object are identified and arranged in the core process. Other processes for further relevant objects are added to the core process. The advantage of this procedure is, that only relevant aspects of the system are modelled and that the model is created from inside (the core process) to outside (less important processes). To support this modelling process with a tool, rules for particular model component management are needed. These rules define how particular model components of five basic types can be linked together. Furthermore, a brief overview about the simulation of the created model is given. Finally, the Virtual Factory Lab Karlsruhe (Victor) is presented to demonstrate the implementation of the proposed approach.

6 ACKNOWLEDGEMENTS

This research was performed at the Institute for Real-Time Computer Systems and Robotics (IPR), Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, Computer Science Department, University of Karlsruhe. We would like to thank Prof. Rembold for his support.

7 REFERENCES

AESOP (1994) *SIMPLE++ Uebersicht 05/94.*, Product Description.

Aguiar, M.W.C. (1995) An Approach to Enacting Business Process Models in Support of the Life Cycle of Integrated Manufacturing Systems. Doctoral Thesis, Loughborough University of Technology

Browne, J. and Sackett, P., Wortmann, H. (1994) Industry Requirements and Associated Research Issues in the Extended Enterprise. European Workshop on Integrated Systems Engineering, Grenoble.

Bruno, G. and Agarwal, R. and Reyneri, C. and Chiavola, B. and Varani, M. (1994) Making CIMOSA Operational. Proceedings of the European Workshop on Integrated Manufacturing Systems Engineering, Grenoble.

N.N. (1994) An evaluation of CIM modelling constructs - Evaluation report of constructs for views according to ENV 40003. Computers in Industry 24(1994) 159-236.

ESPRIT Consortium AMICE (1994) *Open System Architecture for CIM.* Springer-Verlag.

Didic, M. and Neuscheler, F. and Bogdanowicz, L. and Klittich, M. (1993) McCIM: Execution of CIMOSA Models. Proceedings of the 9th CIM-Europe Annual Conference.

Didic, M. (1994) Cimosa Model Creation and Execution for a Casting Process and a Manufacturing Cell. Computers in Industry, 24(2-3).

N.N. (1994) *CIMOSA - Open System Architecture for CIM - Technical Base Line (Version 3.0).*

Gehrsitz, Th. (1995) Entwurf und Implementierung eines Werkzeugs zur Unternehmensmodellierung gemaess CIMOSA. Master Thesis, University of Karlsruhe.

Goldman, S.L. and Nagel, R.N. and Preiss, K. (1995) *Agile Competitors and Virtual Organizations.* Van Nostrand Reinhold, New York.

Limberger, M. (1996) Bottom-up Modellierung mit CIMOSA. Studienarbeit, University of Karlsruhe.

McMenamin, St. and Palmer, J.F. (1988) *Strukturierte Systemanalyse.* Verlage Carl Hanser / Prentice-Hall International.

Rembold, U. and Lueth, T.C. and Ogasawara, T. (1994) Intelligent Manufacturing Components for Future CIM Systems Proceedings of the IMS Conference, Vienna.

Roessel, S. (1995) Modellierung und Simulation eines Fertigungssystems. Master Thesis, University of Karlsruhe.

van Rillaer, D. and deVries, M.C. (1994) CIMOSA / RD User's Guide. VOICE Consortium Document R94.176, Siemens Nixdorf - VOICE Cooperation.

Zelm, M. and Vernadat, F. B. and Kosanke, K. (1995) The CIMOSA business modelling process. Computers in Industry 27, 123-142.

8 BIOGRAPHY

Walter Reithofer obtained his diploma in computer science in 1993 from the University of Karlsruhe, Germany. In his master thesis he developed a new micro stereolithography process for rapid prototyping. Prior to his computer science studies, he received a diploma in mechanical engineering in 1987 from the Profession Academy Heidenheim, Germany. Since 1993 he has been a research assistant at the Institute for Real-Time Computer Systems and Robotics at the University of Karlsruhe. His research interests are in computer integrated manufacturing, especially in enterprise modelling and simulation.