The BOS-Method Architecture:

An Improved Structured Approach for the Development of Distributed Information Systems

M. Breu[1], G. Leonardi[2], B. Percie du Sert[3],
L. Barengo[2], M. Pfeiffer[1], J.-C. Utter[3]

[1]Siemens Nixdorf Informationssysteme AG, EMSC, Otto-Hahn-Ring 6, D-8000 München 83
[2]Olivetti Ricerca, CASE Technology Group, Via Campi Flegrei 34, I-80072 Arco Felice
[3]Bull S.A., rue Jean Jaurès, F-78340 Les Clayes-sous-Bois

Abstract. A crucial point in all system development methods is the consistency problem of analysis at different abstraction levels. This point become even more important in the development of widely distributed information systems which needs to handle the feed back from technical details to the general distribution issues early in the analysis.

Although a structured methodology ([Yourdon 89]), by allowing the study at different levels of abstraction in parallel, is able to handle this feed back problem, it does not overcome the difficulty of managing and monitoring the development process. In order to solve this problem a rigorous life cycle model, structured in phases and stages, is needed. These two requirements appear to be contradictory.

This paper shows an improved structured methodology based on event-oriented modelling which harmonizes the parallel analysis on different levels of details with a life cycle based on phases and stages. It presents a methodology using a hierarchy of abstractions of models, which, although based on phases and stages, does not follow the idea of waterfall development.

1 Introduction

The complex distributed systems, now emerging, make new demands on the development methods. Although applied in many projects, traditional approaches like SSADM ([SSADM V4 90]), or Merise ([Tardieu et al. 89]), DOMINO ([PHB 91]) etc. need to be improved further to meet these demands.

Distributed information systems typically consist of several data-processing systems that communicate with each other, in order to achieve the objectives of the system. Thus technical issues like fault tolerance in case of break down of a remote system component, or autonomy of data processing systems in case of communication failure, will deeply influence the more abstract descriptions of the information system's tasks. Development methods must be able to address these problems.

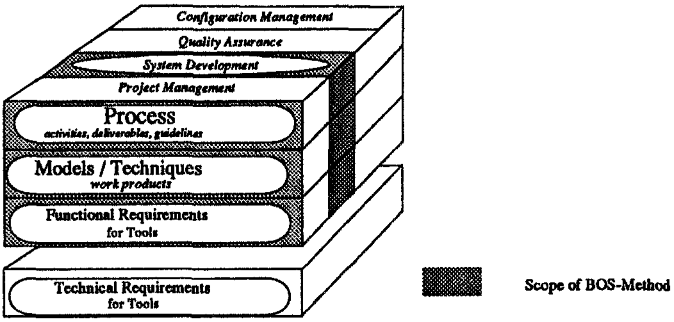
A solution is the parallel development of an information system on several levels of abstraction. Industrial system development methods also need a strong structuring into well defined stages in order to allow efficient project control. Only one of these two demands is solved by the structured approach and the classical waterfall model. BOS-Method defines the levels of abstractions for the system description independently from the levels of decisions to be taken in the project. Both perspectives are linked in a life cycle that resolves the feed back problem in system development.

1.1 History, Scope, and Area of Application of BOS-Method

The three companies, Bull, Olivetti and Siemens Nixdorf are engaged in a joint initiative (BOS) for the development of trans-european information systems.

The European Methodology and System Center (EMSC) as the technical center of this initiative took over the definition of the necessary concepts. As part of the EMSC the working group on methodology analysed the opportunities and necessities of a common information system engineering method. As a result a harmonised method based on those of the three companies (Omega®/Merise from Bull, MOiS® from Olivetti and GRAPES® /DOMINO® from Siemens Nixdorf) was developed: the BOS-Method.

Development projects are structured into project management, system development, quality assurance, and configuration management, each of which is based on an individual process and individual models and techniques. The activities are supported by tools and interact with each other.



BOS-Method is a method for system development. It thus encompasses the definition of a development process, a system model and modelling techniques. In addition the interfaces to the other project activities are defined.

The functional requirements for tool support are determined by the definition of BOS-Method and its interfaces to project management, quality assurance and configuration management. The current version of BOS-Method concentrates on the requirements definition and design phases of the system development process and the related models and modelling techniques.

The main concern of the current version of BOS-Method, in the area of system modelling, is the definition of the contents of system models that describe requirements and the design of an information system from the business and from the technical points of view. In the development process main emphasis was put onto the definition of the required milestone deliverables.

The development process is currently limited to start after global enterprise analysis (which aims to detect possible areas for the introduction of information systems) and to end with the installation and acceptance of the information system. Thus enterprise analysis and the maintenance process is not covered in this version.

The systems which are developed with BOS-Method are within the business and administration area. They can be highly distributed and are based on the common system software platform of the BOS companies. The distribution is a property that is not tackled in detail by existing methods and is of special interest for BOS-Method.

A typical example for distribution is a car rental information system where information about cars must be exchanged between rental stations.

1.2 The Structured Approach

BOS-Method, in line with other ***structured*** approaches [Yourdon 89], or [SSADM V4 90] offers a number of advantages:

- Top-Down approach: the method allows a start at a high level, initially showing a broad picture, and then allowing gradual and controlled decomposition into increasing levels of detail.
- users involvement through all levels of the development process: users contribute to the development of the system that meets their needs.
- regular and formalised walk-through procedures: all work done on the project is critically reviewed for quality, completeness and applicability.
- standard framework for development staff to follow and for management to control.

These features, although important are the typical benefits that one would expect when using any structured method.

In the following the architectural structure of BOS-Method is outlined. For a complete description see [BBPU 92]. The architecture and concepts of the system model are presented in the next chapter. A description about the principles and architecture of the development process follows. Finally, further principles of the structured approach are discussed and are related to BOS-Method.

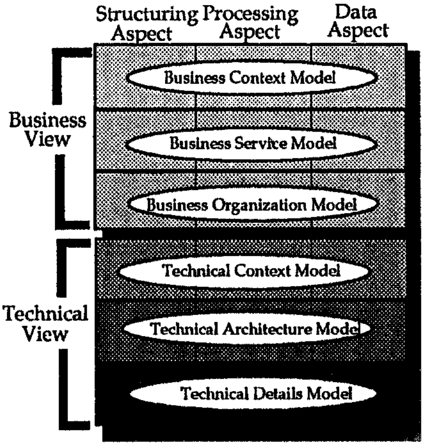
2 The System Model

The system model of BOS-Method is an abstraction of the current or planned real world information system. It documents all relevant facts that are necessary to analyse, design and construct that information system and shows the relationship between these facts. It must serve as a mean to communicate between different types of people involved in the development project.

The information system is described under two main viewpoints: the ***business view***, containing facts representing the business and management rules relevant in the business area of the information system and the *technical view*, containing all facts relevant for the technical construction of the data-processing systems supporting the information system.

Each view is divided into three abstraction levels focusing on different details of the information system. Each level is a self-contained submodel of the system model. On each level the facts are grouped into the different aspects of

- *structure*, identifying the components of the system and their relationship
- *process*, describing the dynamics of the system and
- *data*, describing the information the system components have to store and to exchange between each other.



In the following the purpose of each level, the major concepts documented and the work products associated with each level are described. The following icons on the left side, show which parts of the system model are addressed by each section.

2.1 The Business View

In the business view the information system is understood as a reactive event processing system that steers the execution of enterprise services and keeps track of services the enterprise calls on. The control of the services is described by business transactions (see [Aue, Breu 92]). In the business view manual and automated activities are treated in the same manner. The business view only defines *what* has to be automated but not *how* it is automated.

2.1.1 The Business Context Model

The business context model documents the precise scope of the information system.

The events the information system has to react on or has to emit define the scope of the information system. These events may be messages sent from/received by individual persons or other information systems (automated or manually) interacting with the considered information system (*external partners*), or the arrival of a certain point in time when a business transaction has to start.

The business transactions are identified by the events that trigger them and the events emitted by them.

2.1.2 Business Service Model

The purpose of the business service model is to define conceptually the services the information system provides or monitors. It specifies the behaviour of the information system and its information structure. The information system is viewed as a set of interacting information resources. Information resources are independent objects that interact via message exchange.

To be more precise, *information resources* encapsulate a subset of the information structure together with the actions that manipulate this information. For each information entity of the information structure exactly one information resource has the update right for it. Technically information resources are documented by a list of actions and an E/R-model showing the accessed entities and their relationships. The sum of all these E/R-models give the complete information structure. Thus information resources are data capsules in the sense of abstract data types.

The information resources are subject to distribution on different data processing systems on the next abstraction levels. Information resources may be grouped to information subjects to ensure the comprehensibility of the description. This builds a conceptual hierarchy.

Each *business transaction* is split up into smaller activities to define the precise execution of the business transactions inside the information system. To link the processing and the data aspect, activities are refined further to the actions on the information structure.

2.1.3 Business Organisation Model

The purpose of the business organisation model is the mapping of the activities in the information system structure onto its physical components.

The organisational structure of the information system consists of the geographical locations and the actors working in these locations. Business transactions are executed as a chain of activities of several actors. The activities are synchronised by the exchange of internal messages between actors. If the business transaction spans over several locations these messages must be transmitted between locations.

For each information resource it is decided, whether to automate it or not.

For each actor procedures are defined that document his contribution to the business transactions. In these procedures manual, automated interactive, and batch processing parts are identified.

For each location the data referenced is described in a *location data view*. It documents the subset of data that is accessed, or manipulated by the actors in each location.

2.2 The Technical View

The technical view defines *how* the automated parts of the information system are implemented.

2.2.1 Technical Context Model

The technical context model defines the requirements for the data-processing systems (dp-systems) that automate the particular part of the information system that is to be automated.

In general a distributed information system consists of more than one data-processing system. Geographical locations are typical candidates for the installation of a dp-system.

The external requirements to the behaviour of each dp-system are specified by the structure of the dialogue between the users and the dp-system and the communications with the other dp-systems.

The data storage for each dp-system is specified by the selection of data entities that are actually stored locally. This selection also determines which data entities have to be accessed remotely (via explicit communication protocols or a distributed data base management system).

Each dp-system is assigned the set of information resources that are needed to support the activities of the actors.

2.2.2 Technical Architecture Model

The technical architecture model describes the hardware and software architecture of the data-processing systems, including physical communication aspects.

The architecture of the interactive and batch applications including the definition of the implementation base (specification of hardware components, data base system, transaction monitor, etc.) Also the user interfaces and the electronic data exchange protocols are defined.

The data base schema is optimised according to the selected data base system and the required data accesses and updates.

2.2.3 Technical Details Model

This model is the last abstraction level before code production. It defines the precise implementation specification for each dp-system.

On this level the software modules and the functions are specified by pseudo code techniques, decision tables, etc..

3 The Development Process Architecture

BOS-Method provides structural standards for carrying out system analysis, system design and system implementation.

The development process comprises a requirements definition phase, a design phase and an implementation phase. Each phase is composed of several stages. During a stage, development activities are performed to produce the work products.

Phase and stages are characterised by the documentation to be delivered at their end. This documentation mainly consists of a set of work products describing different parts of the system model. Thus there are clearly defined interfaces between phases and between stages in the form of documents delivered for review and project reporting.

Each of the phases terminates at a baseline, at which the contents of a defined set of work products are frozen. At the baseline, these products are submitted to change control. Thus the baselines provide a link from system development to change control.

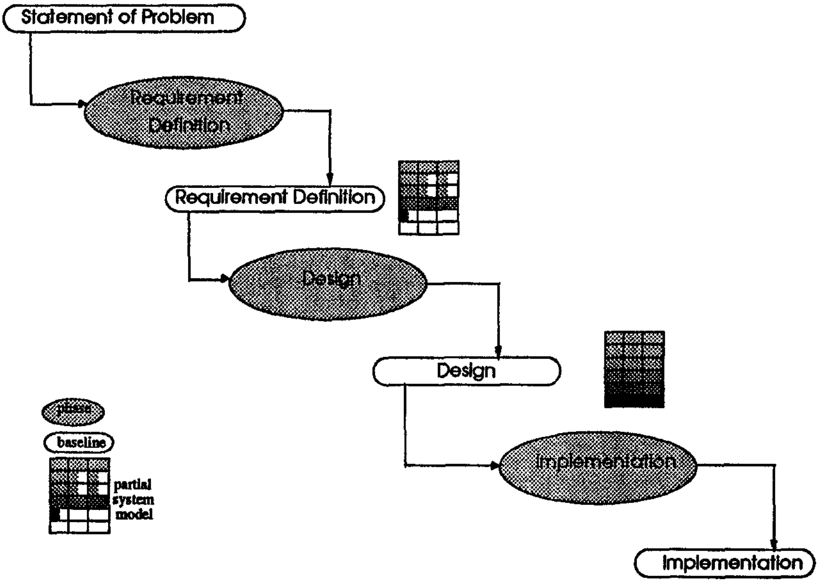
Each of the stages terminates at the milestone. At the milestones, i.e. at the end of each stage, the products required at that milestone are monitored, for project and quality control, and management decisions are taken. The stages and milestones provide a link from system development to project management.

The advancement through the stages of the development process is not strictly coupled with a progress through the abstraction levels. At some points in the development lower levels may be anticipated to assess the effects of a design decision. Also technical decisions may have a feed back to higher levels.

This chapter will describe the development process. The description is organised in hierarchical levels. An overview of the development process as structured in phases with their baselines is first given, and then every single phase will be decomposed in stages with their milestones.

3.1 Phases and Baselines

The development process defined by BOS-Method goes through three phases. Each phase starts and ends with a baseline. The starting point of the development process is the statement of problem which is considered as a baseline too. Then the process goes through the requirements definition baseline, design baseline, and it ends with the implementation baseline: at this point the maintenance process starts.



The aim of the **requirements definition phase** is to elicit the actual requirements of the client, starting from an informal and perhaps incomplete description of the requirements for the information system provided by the client. In the requirements definition phase, the scope of the required information system, its structure, and the approximate processing and data are specified. Additionally the parts of the information system that are implemented by data-processing systems are identified. The activities of this phase are performed in close co-operation with the client.

The result of the **design phase** is the complete specification of the structure, processing, and data stored the data-processing systems. Starting point is the formal requirements specification produced in the requirements definition phase. This phase implies less involvement of the client than the requirements definition phase as here mainly internal details of the system are defined.

The purpose of the **implementation phase** is the production and installation of the data-processing system as specified in the design phase. The development process ends with the acceptance of the final product by the client.

Statement of Problem

At the beginning of the development process, a statement of problem is provided. This is the initial document and serves as starting point for the first phase of the development process.

The statement of problem gives an informal specification of the information system to be developed. It is not required to be precise or complete.During the requirement definition phase a complete and formal description of the requirements will be elaborated. Therefore the statement of the problem may only address a subset of the following topics: *List of problems to be solved by the new information system, Scope of the new information system, Other topics (e.g. planned time frame, budget)*

Requirements definition baseline

The result of the requirements definition phase is the basis of a contract between client and supplier for the design of the information system and the production of the related data-processing systems.

Therefore the business context model and the technical context model go under change control. Thus after the requirements definition baseline, the formal specification of the scope and the requirements for the information system and the data-processing systems, are frozen.

Design baseline

At the end of the design phase the design baseline is reached. At this point the complete system model, i.e., the complete hierarchy of models go under change control. Thus after the design baseline, the complete formal specification of the required system, is frozen.

Implementation baseline

At the end of the implementation phase, the implementation baseline is reached. At this point the installed and accepted final product is delivered to the customer.

3.2 Stages and Milestones

Each of the phases of the development process is decomposed into stages:

Requirements definition phase
- Preliminary study
- Information System Requirements Specification
- Data Processing System Requirements Specification

Design phase
- Information System Specification
- Data Processing System Specification

Implementation phase
- Qualified Data Processing System
- Installed Information System

Each stage starts and ends with a milestone. The work products of the stage is identified in a milestone which represents the starting point of the next step.

A more detailed overview of the development process, including the description of stages and milestones, is now given. The attention will be focused in the aim of the stages and in the contents developed at the respective milestone.

Preliminary study

The purpose of the preliminary study is the analysis of the problems in the current information system and the elaboration of requirements and constraints for the new information system.

The preliminary study provides a basis for decision on the scope of the information system and the data-processing systems to be built. It starts from an informal and probably incomplete specification of the required information system in the statement of problem.

- The decision is between options showing possible scopes of the information system and possible degrees of automation.

- The decision is based on estimations of profit, cost, and capacity for development and operation of each option.

In the following the work products to be delivered at the end of the preliminary study are stated. The degree of detail and completeness of some of the work products depends on management decisions.

Products to be delivered at the milestone*

- For each option : Draft business context model
- For each option: Estimations of the cost and effort for development and operation of the option (This may require deeper elaboration of options)
- Draft glossary fixing terms relevant for the production of the system
- Informal requirements catalogue containing a detailed list of requirements, problems to be solved, and other constraints (imposed or chosen)

Information system requirements specification

In this stage all functional requirements for the information system are defined. The main design decision during this stage is which business transactions the information system has to support and which are out of its scope.

To validate the system scope, a first cut of the structure of the business services and the business organisation is developed.

Products to be delivered at the milestone

- Complete business context model
- First cut business service model
- Updated glossary
- Informal documentation of additional organisational and technical requirements.

Data-processing system requirements specification

The goal of this stage is the definition of the complete requirement specification for the data-processing systems that automate the information system.

To achieve this, the business service model and the business organisation model are developed in more detail. On the basis of this selection a first cut architectural structure of the data-processing systems is given.

Products to be delivered at the milestone

- Complete business context model
- Draft business service model
- Draft business organisation model
- Complete technical context model
- First cut technical architecture model
- Updated glossary

Information system specification

The purpose of this stage is to provide a complete specification of the business view of the information system. This is done in parallel with the further elaboration of the technical view, i.e. of the data-processing systems. Thus the organisational details of the information system can be adapted according to technical needs.

Products to be delivered at the milestone

* Work products can go through different degrees of maturity from milestone to milestone: *first-cut*, *draft* and *complete*

- Complete business view of the information system
- Complete technical context model
- Draft technical architecture model
- Draft technical details model
- Updated glossary
- First cut manuals and tutorials

Data-processing system specification

During this stage, the system model is completed. Thus after this stage the data-processing systems and the manual procedures to be implemented are specified completely.

Products to be delivered at the milestone

- Complete business view of the information system:
- Complete technical view of the information system:
- Complete glossary
- Complete test and qualification plans
- Draft installation plans

Data-processing system production and qualification

In this stage the software for the data-processing systems is produced, tested, and checked against the initial requirements.

Products to be delivered at the milestone

- Complete data-processing system
- Complete test and qualification documentation
- Complete manuals and tutorials

Information system installation

In this stage the hardware and software for the data-processing systems are installed and put into operation. The stage - and with it the development process - terminates with the acceptance of the information system by the client.

Products to be delivered at the milestone

- Installed information system
- Client declaration of acceptance

4 An Improved Structured Approach: BOS-Method

Whilst BOS-Method approach is a structured one, there are a number of factors that together make BOS-Method a unique and particularly effective way of developing information systems.

- The distinction between the business service level and the business organisation level allows to model and develop geographically wide distributed information systems.
- BOS-Method, although structured in phase and stages, solves feed back problems in the waterfall model by taking into account several abstraction levels in one stage.
- BOS-Method separates business views from technical views, introducing an abstraction level hierarchy, both allowing accurate business definition of system data and processing, and covering technical design including definitions of files or databases, programs and control, and resource usage estimation and optimisation.

- BOS-Method is not driven by a single pet technique, e.g. data flows, but allows the parallel development of three crucial system aspects, data aspect, structuring aspect and processing aspect.
- BOS-Method can be used both by inexperienced developers, providing, as far as possible, a cook-book approach, as well as by experts, allowing them to vary the sequence of techniques to make them fit more closely to the projects needs.
- BOS-Method provides a way for identifying and checking system logic before programming.

The BOS-Method definition in [BBPU 92] gives a complete description of the method. In the following we shortly discuss the benefits of this method.

4.1 Separation of Business and Technical Views

BOS-Method separates the business views from the technical one. While the conceptual stages of analysis and design are being undertaken, the developer and user are not concerned with any aspect of technical systems. The business view comprises all facts that are relevant for management, organisation and decision making. It is independent of the degree of automation of the information system. The technical view defines all facts that are needed to construct the data processing system and to establish their communication. Each view is structured by different abstractions. An abstraction emphasises certain facts of the information system and is documented as a dedicated model.

4.2 Checkable/Provable System Logic

A major problem in the analysis and design phases of development is the lack of a systematic way of ensuring that all the necessary system logic has been adequately defined. Another problem is proving that unusual sequences of events or error conditions will be identified and appropriately dealt with by the system.

Most errors in system logic, as opposed to programming logic, are discovered during the following stages:

- during the writing of program specifications, when the author should consider all possible error conditions and write procedures to cover them,
- during programming, when the programmer discovers that not all eventualities have been catered for,
- during program testing, when tests result in unexpected sequences or errors during system and/or acceptance testing,
- during operation, when actual events from the outside world result in transactions occurring out of sequence and causing inconsistencies and system errors.

As shown in [BBPU 92] BOS-Method provides a method of checking and proving system logic during the analysis and design phases. It provides a systematic approach to

- identify the events that trigger changes to the system's data
- identify the correct sequence(s) of those events,
- specify the processing required to handle each event,
- produce a simple-to-use description of processing, error handling and valuation requirements for each event.

4.3 Flexible Methodology

The problem of defining a too strictly rules-based methodology, useful for a beginner but too binding for an expert, or a loosely rules-based methodology, useful for experienced developers but easily opened to free interpretation for the developer without experience, is here resolved. The versatility of BOS-Method allows it to be used with an extreme degree of flexibility, from the inexperienced developer to the expert.

The method provides in general, objective, rules-based techniques for analysis and design; where this cannot be done, guidelines and check lists take the place of rules.

At all times, the developer can choose to have a detailed task list and to adopt (and how to adopt) the sequence of those tasks: the philosophy behind BOS-Method, allows the experts to vary the sequence of these techniques to make them fit more closely to the project needs.

4.4 Methodology for Distributed Information System

BOS-Method is also addressed to develop large scale distributed information systems.

The development of these distributed information systems has to tackle an enormous quantitative and qualitative complexity. Therefore BOS-Method recognises it as strategic to give the possibility to specify the behaviour of the information system early in the analysis, and to provide a feature to handle problems and aspects related to the temporal and spatial distribution.

The concepts of Business Transaction, Information Resources, Actors and Locations are identified to facilitate these tasks.

In particular, Business Transactions are thought as an adequate form to communicate with the user to capture their points of view of understanding of the business. In order to distinguish, indeed, the facts for the temporal and spatial distribution BOS-Method bases its strength in the separation of these aspects (temporal, spatial) into abstraction levels.

In the business view information resources are defined independently from distribution aspects, just by identifying the relationships between them and the relationships to business transactions. In the technical view the information resources are the components that are distributed onto different locations, depending on performance issues, as access frequency of the actors in the information system.

4.5 Going through the Waterfall Model

The main problem of the waterfall life cycle model is to handle feedback, i.e. changes of products of earlier phases. BOS-Method shows how to solve this problem without raising project management problems inherent in spiral models.

The separation of milestones and abstraction levels is important. This allows to look ahead for judgement of risks and reliability of decision that have to be taken at high levels of abstraction.

However this separation is not enough. The collection of knowledge during analysis and design naturally proceeds in parallel at several abstraction levels. The products describing the separate levels depend on each other. To allow for a controllable engineering process in spite of this, BOS-Method explicitly recognises the state of maturity of a product. It allows to go deeper in the refinement of the model (ie. deeper in the hierarchy of the model) without having the work products of the more abstract model level being definitely completed, but nevertheless also allows to control the development process effectively.

5 Conclusions and Way Forward

BOS-Method has its roots in the harmonisation of the software engineering methods Omega/Merise, MOiS, and DOMINO/GRAPES. It took over the outstanding features of each of these methods and combined them in a powerful state-of-the-art method-architecture. A first project demonstrated and improved its practical applicability.

But the development of BOS-Method is not finished. Still the integration of software production and maintenance phases, the formalisation of the connection to other issues of project development, and the development of a well defined modelling language are in preparation.

References

[Aue, Breu 92] A. Aue, M. Breu: Distributed Information Systems: An advanced Methodology, EMSC Report, 1992

[BBPU 92] L. Barengo, M. Breu, M. Pfeiffer, J.-C. Utter: Definition of BOS-Method - A Method for the Development of Trans-European Information Systems, EMSC Report, München, Ivrea, Paris, 1992

[OMEGA 87] Bull: OMEGA Reference Manual (5 Volumes); Paris 1987

[SSADM V4 90] CCTA: SSADM Version 4 Manuals; Norwich 1990

[De Marco 78] T. De Marco: Structured Analysis and System Specification; New York 1978

[GRAPES 91] G. Held (ed.): GRAPES - Language Description; Munich 1991

[Held 91] G. Held (ed.): Informations- und Funktionsmodellierung mit GRAPES; Munich 1991

[Hopkins/Duschl 92] N. Hopkins, R. Duschl (eds.): The SSADM-GRAPES Comparison Study; Springer, Berlin, 1992

[PHB 91] Siemens Nixdorf: Process Engineering Handbook for Application Software Manufacture and Project Implementation (PHB); Munich 1991

[MOiS 91] Olivetti: MOiS - La Metodologia OIS per i Systemi Informativi - versione 1; Ivrea 1991

[Tardieu et al. 89] H. Tardieu, A. Rochefeld, R. Coletti, G. Panet, G. Vahee: La Methode Merise (3 tomes); Paris 1989

[Yourdon 89] E. Yourdon: Modern Structured Analysis; New Jersey 1989