

Towards a Design Methodology for Decision Support Systems

J. Zuurbier¹, J. Brinkkemper², M. Offereins², N. Odding¹

¹School of Management Studies, University of Twente
P.O.Box 217, 7500 AE Enschede, The Netherlands

²Department of Computer Science, University of Twente
P.O.Box 217, 7500 AE Enschede, The Netherlands

Abstract

This paper proposes the use of process models for DSS design. The kind of process models suggested are task structures and decision structures with simple graphical syntax and semantics. The process models form the basis for a coherent DSS design methodology, based upon the bounded rationality paradigm. The history of DSS and DSS design is discussed to form our theoretical position. The resulting methodology has been tested and evaluated in a laboratory experiment. The results of this evaluation will be used for continuous improvement of the methodology.

Keywords: Decision Support System, Design Methodology

1: The technological perspective

'The growing preoccupation of operations researchers and management scientists with Management Information Systems (MIS's) is apparent... Enthusiasm for such systems is understandable: it involves the researcher in a romantic relationship with the most glamorous instrument of our time, the computer'. This citation is, contrary to what one might think, not a very recent one. It was written in 1967 by Russell Ackoff [1]. In the paper, Ackoff identifies several problems determining the malfunctioning of MIS. But he also identifies very clearly that there are in fact several types of management decisions, each requiring a different kind of support. He argues that managerial decisions '... can be classified into three types:

(a) Decisions for which adequate models are available or can be constructed and from which optimal solutions ... can be derived. In such cases the decision process itself

should be incorporated into the information system...

(b) Decisions for which adequate models can be constructed but from which optimal solutions cannot be extracted... A simulation of the model will ... permit comparison of proposed alternative solutions.

(c) Decisions for which adequate models cannot be constructed. Research is required here to determine what information is relevant'.

This classification fits into our current distinction between MIS, DSS and EIS. But it was not until 1978 that the first complete text on DSS was published by Keen and Scott Morton [11]. Since then, more and more DSS applications have evolved, thanks to developments in hardware (PCs) and software (4th generation languages). These developments lead to questions about the design of DSS applications. Arinze [2] classifies various DSS design methodologies according to paradigm (decision-driven, systemic or data-driven), structure (stage-driven or context-based) and orientation (normative or descriptive). In the beginning of the 1980s, the questions about DSS design were addressed by describing scenario's for system construction [21, 23]. Therefore, we conclude there was a strong *technological* focus in DSS design theory. The emphasis in this theory lay on the S for "system". A central theme was that an evolutionary or iterative approach was necessary for DSS construction. Such an approach can be seen as an iterative cycle between the DSS generator and the specific DSS [12]. This design strategy was contrasted with the linear approach commonly used in the design of the traditional Management Information Systems. Several reasons were identified for an iterative or prototyping approach as a scenario for DSS design.

1 Users have difficulty stating their requirements in advance. New requirements arise as they use the sys-

tem. An iterative design approach enables the users to go through a learning curve.

- 2 Specifications of user requirements have to address aspects of the form of the system, like the user interface and the response times. This can be done by using prototypes.
- 3 There are communication problems between users and analysts, due to differences in background and experiences. The prototype provides a vehicle for communication.

A prototyping scenario may form part of a larger and more formalized life cycle development process. This will ensure proper validation, documentation and testing. Sprague & Carlson [21], however, advocated the throwaway prototype approach as the most suitable strategy for DSS builders. It lowers their risk and their users' expectations. If the users like the prototype, the evolving prototype is hard to avoid, because the users cannot see any reasons to throw it away.

At the end of the 1980s however, the effectiveness of DSS was seriously doubted. Therefore, a number of empirical studies [18, 16] were conducted. It was concluded the decision strategy used by the decision maker is very important [16] and that the system should provide decisional guidance based on such a strategy [17]. As a result, the *decision process* gained renewed attention [5]. Research on DSS design acquired a strong *organizational focus*: the D for "decision" was put back in DSS design. This focus recognizes that the interplay between the organization and the information system is extremely important [15]. Various DSS design methodologies have been proposed as a result of this change in perspective. Arinze & Snehemay [3] put the DSS design process in the larger framework of data collection and system maintenance. Bahl & Hunt [4] base their methodology on the analysis of business events, a participant analysis and a decision content analysis. Similarly, three phases of executive problem definition, solution definition and data collection were proposed in a DSS methodology by Guimares & Saraph [9]. Though these methodologies propose tabulations and lists of intermediate design products, none of them includes a simple graphical formalism to analyse the decision process. This is in large contrast to the literature in MIS design where numerous data modelling techniques (e.g. entity relationship diagrams) and process modelling techniques (e.g. data flow diagrams) have been proposed. This motivated our work on the DSS design methodology that we also tested on third year MIS and Computer Science students. A future assessment of our methodology in comparison with other methodologies as was done by Mahmood & Medewitz [14] should explore the advantages of graphical formalisms for the description of decision processes.

This paper is further organized as follows. Section 2 describes the proposed parts of a design methodology. These parts are combined in a coherent DSS design methodology in section 3. The research method is given in section 4, while section 5 describes the research results. These are discussed in section 6.

2: The methodological perspective

The paradigm used in current research on DSS is the satisficing and process-oriented view of decision making. This assumes bounded rationality of the decision maker. A model of such a decision-making process is given by Simon [21]. It consists of three stages: intelligence, design, and choice. The bounded rationality paradigm implies that we have to analyze the current decision making situation, as a starting point for change. The rationale behind this is that these current circumstances determine the possibilities for a decision maker to generate and evaluate alternatives, thus they determine how far the decision maker's rationality is in fact bounded. Application of this approach implies:

- building a descriptive model of the decision process; and
- improving the existing situation, by giving the decision maker the possibility to generate and review alternatives, thus reducing his bounds on rationality.

To this end, Sol [20] describes a model cycle which can be summarized as description, abstraction, modification and reapplication. The first phase in the modelling process consists of analyzing the existing situation, leading to a description of the decision process as it takes place. This description is called the *descriptive empirical model*. On the basis of one or more descriptive empirical models, a *descriptive conceptual model* can be constructed. This model is still a description of an existing situation, but on a higher level of abstraction. It is more generic, which makes it applicable to other, similar organizations. On the basis of the descriptive conceptual model, and the research results or experiences of the designer, the *prescriptive conceptual model* can be constructed. This model is an improved version of the decision process under analysis. A possible design for an automated system can be based on this model. The specification of the improved decision process in its organizational context, is called the *prescriptive empirical model*.

We suggest the use of *process models* to specify the conceptual models. In process models, the sequence of the different steps taken in the decision process and the decision maker's position are subject to analysis. Todd & Benbasat [24] refer to this kind of modelling as global modelling. '*Global modelling involves a direct*

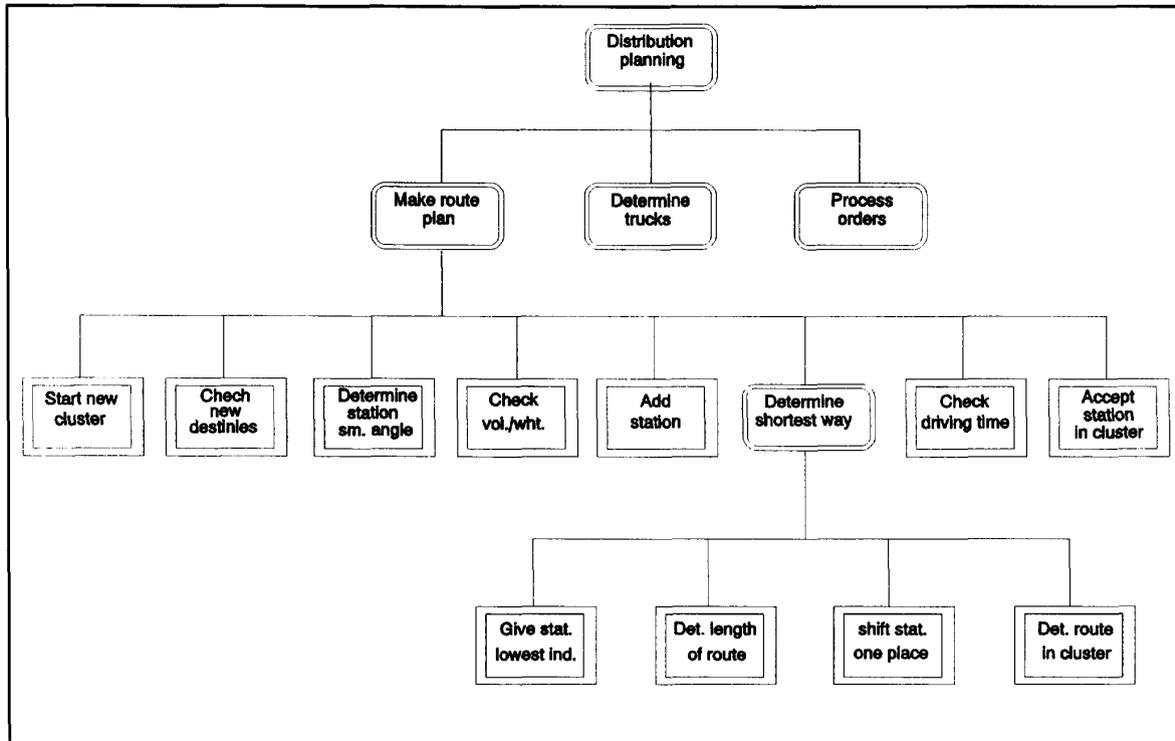


Fig.1: example of a task hierarchy for the gasoline supply case.

examination of the processes of problem solving. This approach shows not only which actions are taken during problem solving, but also the sequence of their execution. A flowchart of the problem-solving process is generally the result of this activity'. The models specify the decision process and can therefore be used to identify where improvements may be made. This makes these kinds of models suitable for use in the design cycle described above. An example of process models for DSS design (in this case a GDSS for knowledge acquisition) can be found in [13]. Examples of process models for MIS design can be found in [6]. Bots & Sol [5] specify a method of modelling decision processes using tasks, decisions and information as separate entities. A task is defined as "solving a problem". The method starts with the description of the task structure. A task structure for a given problem consists of two components. The **task hierarchy**, which shows the hierarchical structure of the task, can be described in terms of subtasks. In figure 1 an example of a task hierarchy is shown for the gasoline supply case (see section 4). We see for instance that the main task "Distribution planning" is subdivided into the tasks "Make route plan", "Determine trucks" and "Process orders". For the first task, the complete task subdivision is shown (see figure 1). The **task flow** defines the

sequence in which the subtasks and coordinating decisions are related. In figure 2 the task flow diagram for the task "Make route plan" is given. Coordinating decisions determine which subtasks have to be executed in which sequence. Figure 2 for instance shows that if an order from a gasoline station doesn't fit in the current volume and weight constraints of a truck, it starts a new cluster of orders. A subtask which cannot be subdivided anymore, is called a leaf task. Each leaf task can be specified in terms of decisions (denoted by circles) and input / output information. This is called a **decision structure**. The whole process of designing task hierarchies, task flows and decision structures, is referred to as "task analysis".

3: DSS design methodology

The prototyping approach, the model cycle as specified by Sol [20] and the task analysis as specified by Bots & Sol [5], can be combined into a coherent DSS design methodology. Observe that we apply a method engineering approach [19] to configure a design method for DSS, by putting together some proven parts of other methods.

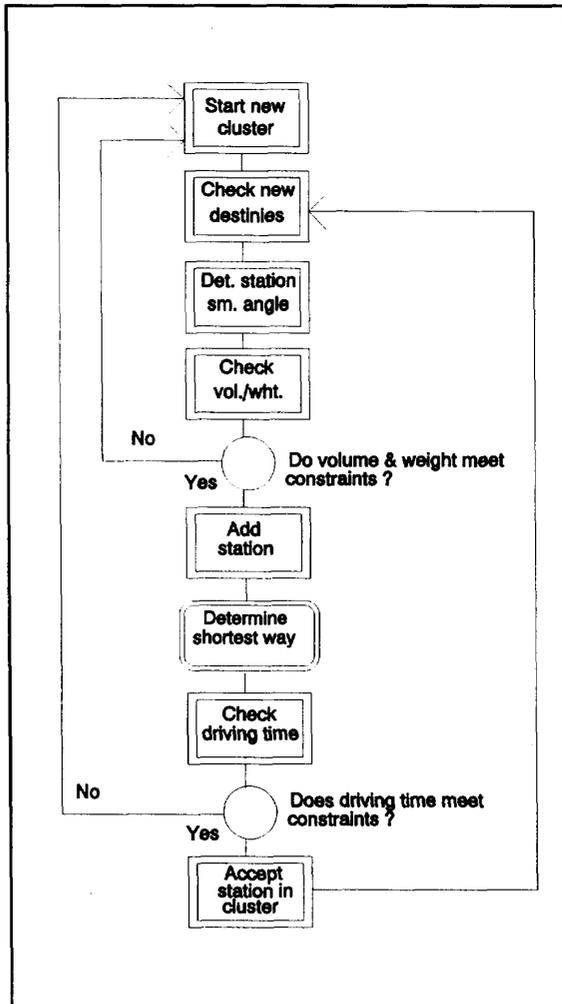


Fig.2: example of a task flow for the task "Make route plan" in the gasoline supply case.

We define prototyping as a method to determine the *complete* specifications of a system, characterized by the use of prototypes in an iterative fashion. We distinguish five stages in the prototyping process:

1. global analysis and specification;
2. design and realization;
3. use and evaluation;
4. adaption;
5. completion of the specifications.

An iterative cycling takes place between the stages 3 and 4. Within the stages 1 and 2, the model cycle of Sol [20] can be applied. The descriptive empirical model as well as the descriptive conceptual model can be specified in the first stage of the prototyping process. The prescriptive conceptual model can be specified during the

design phase. The prescriptive empirical model evolves in the realization phase. The conceptual models can be specified using the process modelling technique discussed in section 2.

The presented design methodology has two advantages. Firstly, a large decision process will consist of many subtasks and decisions. A design methodology should be able to handle this by being able to make distinctions between different abstraction levels. Our proposed design methodology has this ability. Secondly, the methodology should be able to address the bounded rationality of the user. Figure 3 illustrates how bounded rationality is addressed by different aspects of the proposed DSS design methodology. The methodology has been tested in a laboratory experiment.

<i>Simon's phases:</i>	<i>prototyping:</i>	<i>modelcycle:</i>	<i>task analysis:</i>
intelligence	global analysis and specification	descriptive empirical model descriptive conceptual model	tasks gathering information tasks identifying decisions
design	design and realization adaption	prescriptive conceptual model	tasks generating alternatives
choice	use and evaluation completion of the specifications	prescriptive empirical model	decisions

Fig.3: bounded rationality addressed on different DSS design levels.

4: Research method

In the laboratory experiment 40 pairs of students were given a case for which a DSS had to be designed and constructed. The students had already followed courses in Information Management, Systems Analysis and Computer Programming. The case described a gasoline supply company delivering different types of gasoline to different customers. The DSS support in this case was intended to concern the route planning of each gasoline truck. A truck has different sections for different types of gasoline. Each truck is different in size. Customers are located on different locations and order different amounts and types of gasoline. The planning process should be based upon the purchase orders of customers (gasoline stations) and should minimize the driving distance for each truck. To this end, an operations research model was supplied as part of the case description.

For the construction of the DSS the RDMS Paradox 4.1 was used. This software can be classified as a DSS

generator [22] because it is a user-friendly 4th generation language, offering both graphical and modelling facilities, apart from the database facilities. Research results were collected in a qualitative and a quantitative fashion. The qualitative material consisted of two reports on the design of the system. The first report was made by each group before building the system, the second report was made after the realization of the system. Each report consisted of a description of the prescriptive conceptual model in terms of task hierarchies, task flows and decision structures. The quantitative material consisted of closed questionnaires filled in by each individual group member. The questionnaire was constructed with a five points scale, and contained questions about the use of the design methodology.

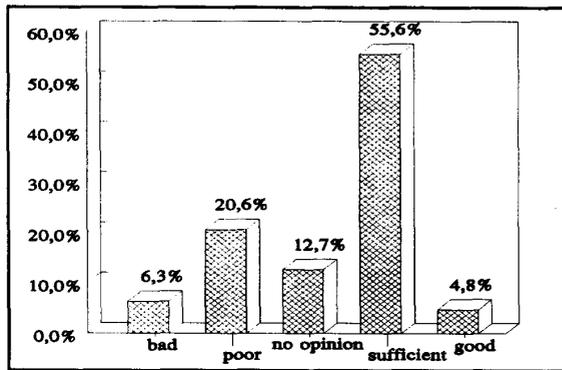


Fig.4: user interface design.

The research approach is comparable with the approach followed by Dennis et al. [8] in their evaluation of Group DSS use. Both approaches evaluate the user satisfaction on the basis of quantitative and qualitative material, in order to identify possibilities for improvement. Notice that we don't use the students as surrogates for managers, which is an often criticized approach [10], but as surrogates for DSS designers. Using MIS and Computer Science students as surrogates for DSS designers makes sense because DSS designers will often have the same background as the students we use as their surrogates.

5: Research results

In this section, we review the results of the qualitative and quantitative material jointly. The questionnaire has been filled in by 80% of the participants. The research results describe the usefulness of the DSS design methodology for system design and construction. This has been tested for three separate areas: menus (the

dialog component), tables (the data component), and algorithms (the model component). The complete questionnaire contained 19 questions. The most important ones are discussed here.

Firstly, the relationship between the modelling process and the menus which have been used has been evaluated. The questionnaire contained the following question: 'Does task analysis offer enough grip for the design of a menu structure (user-interface) for a DSS?'. The results of this question are given in figure 4. A majority of 60.4% thinks the methodology is sufficient or good. Furthermore, it appears from the reports the participants wrote, that the implemented menu-structure has indeed been derived from the task hierarchy.

Secondly, the relationship between the tables used in the DSS and the modelling process has been investigated. For this relationship the questionnaire contained the following question: 'Does task analysis offer enough grip for the design of an information structure (tables) for a DSS?'. This question leads to the results as described in figure 5. A majority of the respondents (46.1%) think that the task analysis is sufficient for its purpose. However, 38.1% considers that the methodology not sufficient. In the reports the respondents indicate that the decision structures, which should describe the data requirements, offer no support in designing the tables. The fact that the necessary algorithm and its data requirements have been given beforehand, could be an explanation for this result. But the application of task flow diagrams in the area of meta CASE tools [7] confirms the outcome in figure 5. The modelling of a requirements engineering process involving tasks, decisions and documents required an additional document modelling technique based on entity relationship diagrams.

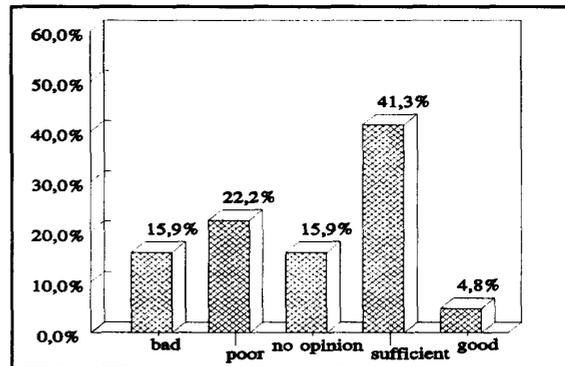


Fig.5: information structure design.

Thirdly, the relationship between the modelling process and the algorithm used has been evaluated. Primarily the task structures are thought to be useful for

mapping the algorithms. The question asked in the questionnaire was: 'Does task-analysis offer enough grip for designing algorithms (models) for a DSS?'. Figure 6 shows the results.

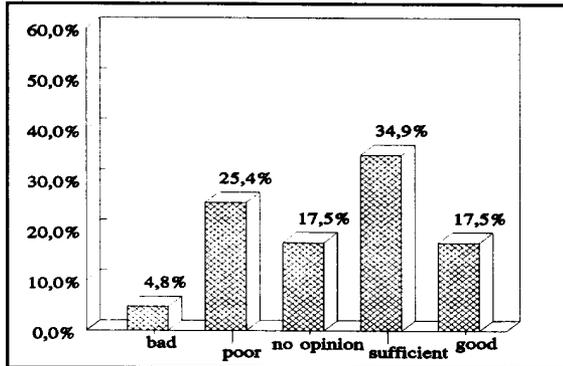


Fig.6: algorithm design.

To evaluate the application of the methodology for the design of the procedures (algorithms), we have also compared the procedures written by the respondents with their task structures. In most of the cases there was a good resemblance between the task structures and the procedures written. As can be seen in figure 6, most of the respondents also think that task analysis is useful for the design of the algorithm (52.4%).

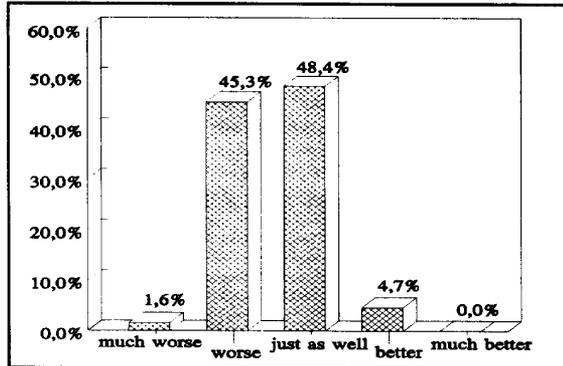


Fig.7: added value of the methodology.

The respondents have also been asked *how the modelling process would have been if they had not used the design methodology*. Figure 7 shows the results of this question. 46.9% of the respondents think that the modelling process would have been worse or much worse without the task analysis. In the reports a lot of the participants indicate that the analysis helped them to oversee the problem and structure their thoughts and the design process. The relative small number of respondents indicating the positive effects of the task analysis on the

design process, is in our opinion due to the pre-structuring of the case. Restrictions on the educational infrastructure did not allow an unstructured case description with a lot of open questions.

According to the respondents, *the design methodology also has a positive impact on the quality of the DSS*. Figure 8 shows that 42.2% of the respondents think the quality of the system would have been worse without using the task-analysis. However, 48.4% of the respondents indicate that the quality would have been just as well. Again, the degree of structure in the case description might be an explanation for this result.

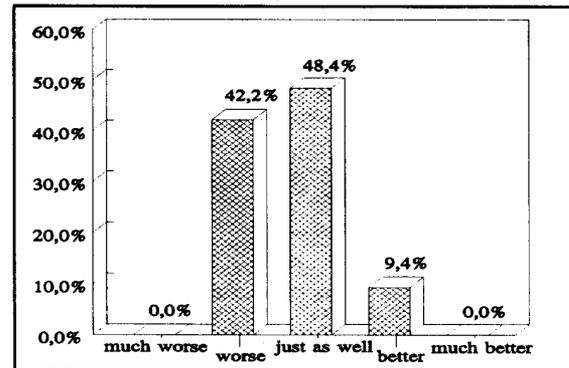


Fig.8: DSS quality.

In order to apply the task analysis, the respondents only got a short description of the modelling process. The reports show however that task analysis is an easy to learn modelling process. The first reports of the respondents were already quite good, but in the final report, the task analysis was in most of the cases sufficient to good. The learning time for task analysis has also been evaluated by a question: 'Do you think you can model decision problems yourself, with the help of task analysis?'. The results of this question are shown in figure 9.

A final remark can be made on the amount of detail of the analysis done by the respondents. From the reports and from the systems build it appears that task-analysis can be done on quite different levels of detail. Further guidelines will be incorporated in the methodology assisting the creation of task models of sufficient detail.

6: Discussion

The research findings clearly show the advantages of a structured design methodology for DSS involving graphical modelling techniques. Notations for tasks and decisions, that are easy to learn by designers and easy to

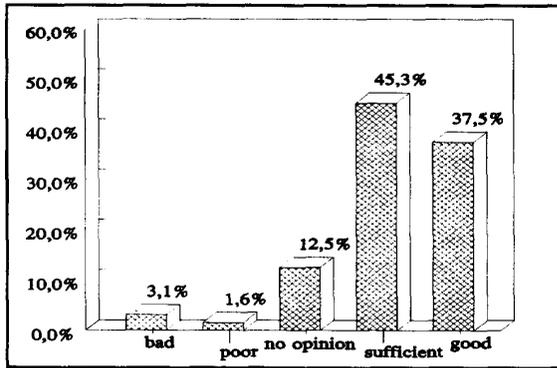


Fig.9: comprehensiveness.

read by users, are a necessity for a simple, but structured design process. It is our intention to use the research results for improvements and expansions upon the suggested DSS design methodology. The results presented here are satisfying, but they also indicate the need for further research. This concerns the use of the decision structures. The results show that the respondents don't like to use this technique to model the data requirements. This could be due to the degree of structure in our case material for the students. However, it is also possible that decision structures aren't suitable for data modelling in a DSS environment. The incorporation of a suitable information structuring technique in the methodology is currently in progress. Furthermore, this design methodology for DSS construction needs to be evaluated in real business practice. At this moment, the methodology is being applied to build a personnel planning DSS for a Rank Xerox service department. The results of this project will be published later. In a later stage, it is our intention to apply the design methodology also on the construction of Group DSS. However, the methodology might have to be expanded for this purpose. Suggestions to do this have already been proposed by Zuurbier [25]. At this moment, the first exploratory cases are being done with the expanded methodology. When the expansions prove to be useful for Group DSS design, we will apply the Group DSS design methodology on a larger scale, to test its applicability.

Eventually, this research into DSS design methodology must, together with on-going research concerning design methods for other applications, contribute to a *design methodology for information systems*. That is: a theory about the construction of methods suitable for information systems design [7, 19].

references

- [1] Ackoff, R.L., 'Management misinformation systems,' *Management science*, 14 (1967), 4, pp. B-147 - B-156.
- [2] Arinze, B., 'A contingency model of DSS development methodology', *Journal of management information systems*, 8 (1991), 1, pp. 149 - 166.
- [3] Arinze, B., and B. Snehemay, 'A framework for effective data collection, usage and maintenance of DSS', *Information & management*, 22 (1992), pp.257 - 268.
- [4] Bahl, H.C., and R.G. Hunt, 'A framework for system analysis for decision support systems', *Information & management*, 14 (1984), pp. 121 - 131.
- [5] Bots, P.W.G., and H.G. Sol, 'An environment to support problem solving', *Decision support systems*, 3 (1987), 3, pp. 225 - 231.
- [6] Brinkkemper, S., N.A. Brand, and J. Moorman, 'Deterministic modelling procedures for automated analysis and design tools', in: Olle, T.W., A.A. Verrijn Stuart, L. Bhabuta, *Computerized assistance during the information systems life cycle*, Amsterdam etc., 1988, pp. 117 - 160.
- [7] Brinkkemper, S., A.H.M. Ter Hofstede, T.F. Verhoef, and G.M. Wijers, 'A meta-modelling based CASE shell to support customized domain modelling', in: Iscoe, N., G.B. Williams, and G. Arango (eds.), *Proceedings of the ICSE workshop on Domain modelling*, Austin, Texas, 1991.
- [8] Dennis, A.R., A.R. Heminger, J.F. Nunamaker, and D.R. Vogel, 'Bringing automated support to large groups: The Burr-Brown experience', *Information & management*, 18 (1990), pp. 111 - 121.
- [9] Guimares, T., and Saraph, J.V., 'The role of prototyping in executive decision systems', *Information & management*, 21 (1991), pp. 257 - 267.
- [10] Hughes, C.T., and M.L. Gibson, 'Students as surrogates for managers in a decision-making environment: An experimental study', *Journal of management information systems*, 8 (1991), 2, pp. 153 - 166.
- [11] Keen, P.G.W., and M.S. Scott Morton, *Decision support systems: An organizational perspective*, Reading, etc., 1978.
- [12] Le Blanc, L.A., and M.T. Jelassi, 'DSS software selection: A multi criteria decision methodology', *Information & management*, 17 (1989), pp. 49 - 65.
- [13] Liou, Y.I., and E.S. Weber, J.F. Nunamaker jr., 'A methodology for knowledge acquisition in a group decision support system environment', *Knowledge acquisition*, (1990), 2, pp.129-144.

- [14] Mahmood, M.A., and J.N. Medewitz, 'Impact of design methods on decision support systems succes', *Information & management*, 15 (1985), pp. 137 - 151.
- [15] Ruyter, K. de, and J.J. Zuurbier, 'Customer information systems: Approaching a new field in information systems from a new perspective', *Information & management*, 24 (1993), pp. 247 - 255.
- [16] Schaik, F.D.J. van, *Effectiveness of decision support systems*, Delft, 1988.
- [17] Silver, M.S., 'Decisional guidance for computer-based decision support', *MIS Quarterly*, 1991, March, pp. 105 - 121.
- [18] Sharda, R., S.H. Barr and J.C. McDonnell, 'Decision support system effectiveness: A review and an empirical test,' *Management Science*, 34 (1988), 2, pp. 139 - 159.
- [19] Slooten, C. van, and S. Brinkkemper, 'A method engineering approach to information systems development', in: Prakash, N., C. Rolland, and B. Pernici (eds.), *Proceedings of the IFIP WG 8.1 working conference on The information systems development process*, Amsterdam etc., 1993, forthcoming.
- [20] Sol, H.G., 'Information systems to support decision processes', in: H.G. Sol, and J. Vecsenyi (eds.) *Environments to support desicion processes*, Amsterdam etc., 1991, pp. 115 - 127.
- [21] Sprague, R.H., and E.D. Carlson, *Building effective decision support systems*, Englewood Cliffs, 1982.
- [22] Sprague, R.H., and H.J. Watson (eds.), *Decision support systems: Putting theory into practice*, 2nd ed., Englewood Cliffs, 1989.
- [23] Thierauf, R.J., *Decision support systems for effective planning and control: A case study approach*, Englewood Cliffs, 1982.
- [24] Todd, P., and I. Benbasat, 'Process tracing methods in decision support: Exploring the black box', *MIS Quarterly*, 1987, December, pp. 493 - 511.
- [25] Zuurbier, J.J., 'On the design of group decision support systems', in: Jelassi, T., M.R. Klein, and W.M. Mayon-White (eds.), *Decision support systems: Experiences and expectations*, Amsterdam etc., 1992.