Model-Driven Business Management – the Linguistic Perspective

Jon Atle Gulla[α] and Terje Brasethvik

Department of Computer and Information Science
Norwegian University of Technology and Science
Trondheim, Norway
Jon.Atle.Gulla@idi.ntnu.no

Abstract: Traditional business models are primarily used in the initial phases of BPR or ERP projects. They are static representations of systems requirements and business procedures, but have proven less useful in supporting the actual use of the systems. With the Model-Driven Business Management (MDBM) approach, dynamic and adaptable business models constructed as part of the implementation project are afterwards used to access the system and monitor the real business flows. Crucial to the whole approach is the provision of advanced modeling capabilities and linguistic techniques for exposing the semantics of the models and ease the use of potentially complex business models. In this paper, we present the linguistic aspects of the MDBM approach and discuss how the linguistic part and the modeling part of MDBM mutually support each other.

1. Introduction

Enterprise Resource Planning (ERP) systems integrate and streamline the business processes of an organization across departmental and geographical borders. Besides having fully integrated databases of business data, they automate and support the business functions with modules for functional areas like finance and materials management. A characteristic of ERP systems is that they are delivered with pre-implemented modules that only have to be customized to the particular needs of the organization.

In most ERP projects, the organization is reengineered either before the project is initiated or as part of the project. Principles from BPR [HC93] or TQM [Ju88] have been widely used in these projects. Common to these reengineering efforts is the idea that the organizations would benefit from a stronger *process orientation.* The ERP technology makes this possible, and most large and midsize companies today employ cross-functional ERP solutions to support their business processes [GM99, VHW00]. An important activity in these ERP projects is the modeling of current and future business processes. The models facilitate the exchange of information between the project members, and they provide the means for documenting and analyzing the structures and processes of the organization.

[α] Dr. Gulla is also the managing director of Elexir Sprach- und Informationstechnologie, a subsidiary of Fast Search and Transfer, in Munich, Germany.

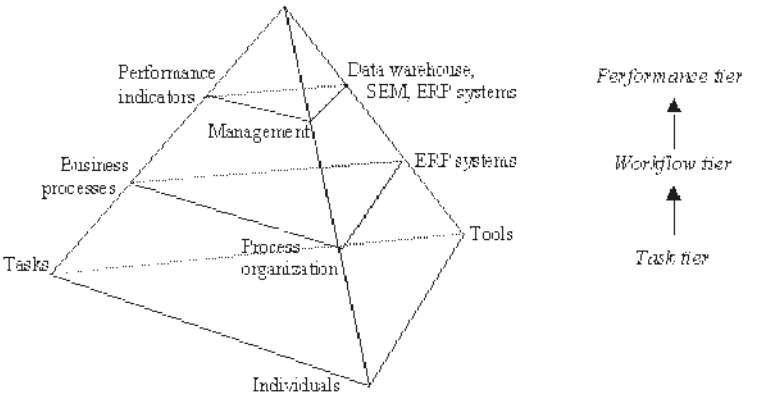

Figure 1. The Business Engineering pyramid

The *Business Engineering pyramid* in Figure 1 illustrates the way information technology penetrates modern organizations. At the bottom tier, individuals use stand-alone tools to perform or automate specific tasks. At the workflow tier, ERP systems enable the organization to implement new and improved business processes. Similarly, various types of information systems are introduced at the performance tier to monitor the performance of the organization and report key indicators to the management. Vertically, the tasks need to fit into the processes and the processes must address the performance indicators and enable their monitoring. For the simplicity of this paper, we will in the following refer to all systems at the two upper tiers as ERP systems.

It should not come as a surprise that business modeling in ERP projects is extremely demanding. The ERP system encompasses several functional areas, and there is usually no single person that has the full overview of the whole system. When modeling the business, both functional and organizational issues need to be taken into account [GAV00]. The models address intricate needs at all three levels of the business engineering pyramid, making them both complex to construct and difficult to use in the subsequent design, testing and training phases. In practice, there are often inconsistent variants of the business model around, and the whole business is not completely modeled [GB00]. Still, it is commonly accepted that modeling is a necessary activity of any ERP projects, and the challenge is then to make the most out of the business models.

In the rest of this paper, we discuss an approach to business management that combines advanced modeling techniques and linguistic resources with ERP environments. Whereas Section 2 explains the main principles of Model-Driven Business Management (MDBM), we present the overall architecture of the linguistic part of our model-driven SAP environment in Section 3. Section 4 is devoted to the linguistic techniques used in our search process. Some preliminary comparisons with other approaches are given in Section 5, and conclusions and directions for further work are found in Section 6.

2. Model-Driven Business Management

With the *Model-Driven Business Management (MDBM)* approach, the models integrate all technological, organizational and procedural issues in the project and drive the whole implementation. After completing the project, the same models serve as user-adaptable

interfaces to the underlying computerized systems, while monitoring the actual use of the system. As opposed to today's ERP solutions, user access control and business process administration are done at the model level rather than at the transaction level.

MDBM requires that models play a central role throughout the system's life cycle:

- **System development**: During the development process, the organization's current processes and structures are matched against the predefined models delivered with the ERP systems. Each element of the model constructed is linked to one or more user profiles (roles) and scenarios. The user profiles correspond to the authorization objects in ERP systems and are used to give groups of users restricted access to the system. Every flow is divided into one or more scenarios, where a scenario specifies a particular variant of the flow, e.g. the purchasing flow for a particular material of the maintenance flow for critical parts that need to be fixed immediately.
- **System Operation**: The complete ERP system may consist of a number of different applications that are integrated into the same business flows. Logging on to the MDBM environment, the user traverses the business model to access all transactions that are included in his user profile. Double-clicking on the leaf processes in the business model, the user is taken to the corresponding ERP transaction. In a similar vein, the user can select document types in the data model and jump to the management reports defined for these document types.
- **System Maintenance:** With MDBM, new users are given access to the underlying ERP solution by giving them a user profile. If the business processes change after the implementation of the ERP solution, most of these changes will be done at the business model level without changing the customization of the underlying system.

Consider the model view in Figure 2, which is part of the total purchasing flow in the Materials Management (MM) module in SAP R/3. Whereas the MM flow is one of the most complex flows in SAP, the model view is simple and involves just a few functions. This view is for the *purchaser* user profile in the *consumables* scenario. The complete MM model, which is partly shown to the left, is very detailed and is not normally used by end-users. As the view is tailored to purchasers with no management responsibilities, the details of the release procedure is not exposed. Also, since consumables in this case means that there is no master data for the materials to purchase, the purchaser does not need to look for contracts or info records (vendor-material price lists) for the material. The purchaser simply creates a purchase order with reference to the purchase requisition, prints the order, and monitors the ordering until the goods arrive. The function *"Create purchase order"* is highlighted because it is the current function in this execution process.

Each of the lowest level processes in this particular model view is linked to an SAP transaction. The *"Create purchase order"* function, for example, takes you to transaction ME21 in the underlying SAP R/3 system. There are also functions that are not linked to specific transactions, but these specify manual tasks that are not supported by any computerized system. There is also a system monitoring module that logs the actual use of the system for later revisions of the business processes.

Implementationally, the MDBM environment is an integration of a documentation tool, a business analysis tool, a business process execution tool, and a systems administration tool.

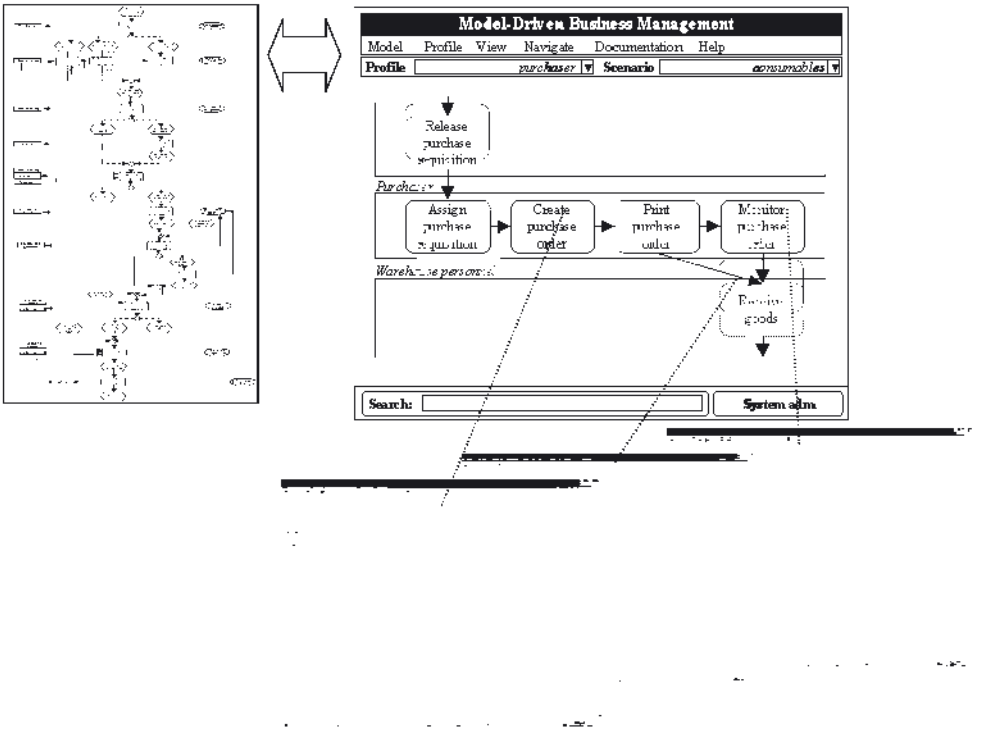

Figure 2. The Model-Driven Business Management environment

3. The MDBM SAP Linguistic Component

SAP R/3 is one of the most complex ERP systems on the market. More than 800 pre-defined business processes across all functional areas are supported, and several thousand transactions are implemented. The system includes 12 customizable modules for finance, human resources and logistics, as well as a number of industry-specific modules [He97]. According to AMR Research's figures from 1999, SAP R/3 has the largest share of the ERP market with 31%, way ahead of Oracle's 14% and PeopleSoft's 7%.

The functionality of SAP R/3 is normally described using an extended process modeling formalism called EPC. For our purposes, though, the modeling language had to be extended with concepts for view modeling, resource modeling, business rules modeling, and report modeling. The complete business model for an SAP solution, thus, gets very complex with several thousand leaf processes and up to 3-5 hierarchical levels. In total there may easily be something like 20,000 elements in the business model.

The business model gives you access to transactions in the SAP R/3 system as well as business data (master data and transaction data) entered into the system. In a midsize SAP solution something like 4-5,000 transactions may be used on a regular basis. The business data includes master data for objects like materials, customers, vendors, plant

structures, and organizational units. The transaction data document all the transactions carried out in the system.

The linguistic component included in the MDBM concept gives you an alternative to traversing the business model. The user can ask the MDBM environment to display a certain model fragment, go directly to an SAP transaction, or display some business data. The interface looks like a normal search interface, as shown at the bottom of the Model-Driven Business Management window in Figure 2. All queries posted by the user are interpreted according to the active user profile and the active scenario.

Consider the model view shown in Figure 3(a). For the user profile "purchaser" and the scenario "consumables", the query "purchase orders" makes the system display the view of the structural model containing the object *purchase order*. Double-clicking on the operation *create* of this object, the user is taken directly to the SAP transaction for creating purchase orders. A query like this is useful if the user would like to know which transactions he is allowed to run on purchase orders. Only the operations relevant to his user profile (and active scenario) are displayed.

In Figure 3(b) the search query "create new orders" is by the linguistic component interpreted as a request for the *Create purchase order* function in the business model. The system displays a view of the process model and highlights the desired function. If a different user profile or scenario had been active, a different part of the model might have been displayed to the user. For example, if user profile *sales representative* had been active, this query would have led the system to display a view of the process model containing the function *Create sales order*.

The linguistic component is built up like a traditional search system, where model elements and business data are indexed like documents in document management systems. In addition, two classifications are used to distinguish between the types of elements indexed. The *phrase type* attribute is set to act for functions in the process model and transactions in the SAP system, structure for structural model elements and master data, and report for reporting structures and transaction data. The *level* attribute is set to model for business model elements and instance for business data elements.

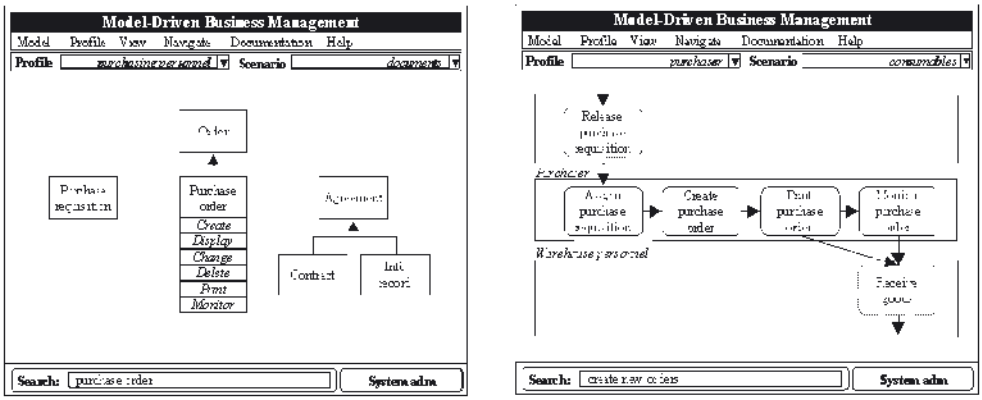

Figure 3. (a) The query "purchase orders" takes you to the structural part of the business model. (b) The query "create new orders" takes you to the purchasing process.

The search process is divided into the following six steps (see Figure 4):

- **Phrase recognition:** First, a tagger is used to determine the phrase type of the query. This information is important for anticipating the intention of the user. Whereas the query "create new orders" is found to be a verb phrase (VP), "new orders" is analyzed as a noun phrase (NP). The phrase type attribute of the query is deduced from this analysis.
- **Level recognition**: Somewhat more difficult is the determination of conceptual level. The system needs to know whether the user is searching for a model element or data in the SAP solution. The hypothesis is that that a model element should be returned unless the query is in the past tense, refers to master data or transaction data, or contains temporal or spacial coordinates. The level attribute of "create new orders" is assumed to be model.
- **Query reduction**: Unnecessary words are deleted from the query, and stemming is used to replace all word forms with their canonical forms. In the example above, "new" in "create new orders" is deleted and "orders" is replaced with the canonical form "order." The deletion strategy today is simplistic and based on the part of speech tags returned by the tagger.
- **Query expansion**: Both the business model and the domain-adapted lexicon contain semantic relations that are used to expand the search. Synonyms are taken from the lexicon, and generalizations and specializations are found both in the business model and in the lexicon. Obvious synonyms like "make" for "create" are added to expand the search. Generalized or specialized terms are included depending on phrase type and conceptual level.
- **Search**: The search engine returns a ranked list of elements that match the query terms and are of the correct phrase type and conceptual level. For the query "create new orders", all document creation functions like *Create purchase order, Create sales order, Create plant maintenance order*, and *Create transport order* are returned.
- **Filtering**: Finally, the system needs to make sure that only elements visible to the active user profile and scenario are shown. If the highest ranked element is not available for the active user profile in the active scenario, the system goes down the ranked list until a possible element is found. For the query "create new orders", the user profile *purchaser* ensures that the function *Create purchase order* is chosen before the other possibilities.

The MDBM architecture is a multi-layer architecture that takes into account the openness and modularity of Web applications. SAP provides access to the company's SAP solution through various Web services, such as the MySAP.com user interface and the server-side solutions like SAP Internet Transaction Server, SAP Web applications servers or business connectors and ALE (Application Link Enabling) interfaces. Hence, we can implement our MDBM solution on top of the SAP installation, rather than as part of it. MDBM components need only access SAP system data through the appropriate interfaces.

The end-user client is built as a separate application, as shown in figures 2 and 3. The business models are stored in a repository and accessed through a Web-server as XMI files. The model repository provides a Model Access API for navigating, presenting

and searching the models. This API is in turn used by the linguistic component, which is implemented as a server side API - denoted "Query Analyzer" (QA). The QA has access to underlying linguistic tools, such as a morphological tagger, as well as to the query context provided by the active user profile and scenario. The final query is sent to the search API of the model repository.

Query results are extracted from the model repository and presented either in the MDBM model window or in a regular Web browser window. The former presentation is used if the query results are references to model fragments or the query needs to be refined by the user. If the query results refer to transactions or business data, the system may present the results in a normal browser window or just activate the desired transaction or report in the SAP system.

4. Using the Linguistic Component

Unlike document management systems, the search facilities here do not return a list of ranked elements. Only one element is chosen, and the user needs to reformulate his query if this was not what he intended.

In many cases, the user will specify a query right in the beginning to jump directly to the relevant part of the business model. He will then click himself through the business model to run SAP R/3 transactions and check the business data in the system. Whereas searching is used to move to entirely different parts of the model, model clicking is often preferable for running business processes.

With the strategies adopted so far, we interpret the queries as follows:

- **"Create purchase orders"**
  This query does not pose any problems. The phrase type is act, and the conceptual level is model. The term "orders" is replaced with "order", and the search is expanded with "make" as a synonym of "create." The search engine returns the model function *Create purchase order*.

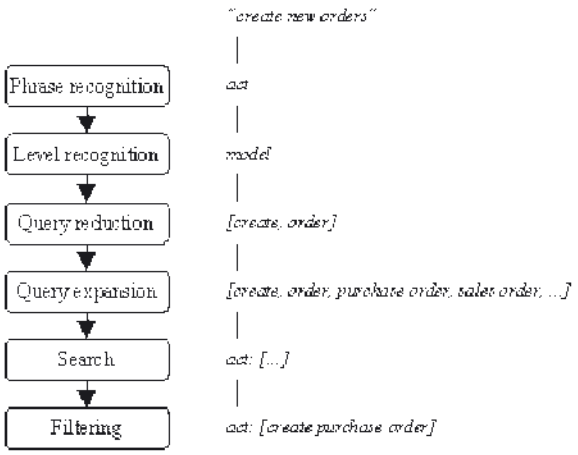

Figure 4. The search process.

- **"Make purchase order"**
  "Make" is a synonym of "create". The returned elements are the same as for "create purchase orders."
- **"Purchase material"**
  This query is of phrase type act and level model, but does not refer to any low-level function of the business model. However, there are several purchasing scenarios that include high-level processes with texts like "Purchase raw material" or "purchase consumable". The term "material" is expanded with possible specializations like "raw material", "maintenance part", "commercial product," and "consumable." The system then displays the high-level process that is most appropriate for the active user profile and active scenario.
- **"Vendor for material X"**
  This query is of type structure and conceptual level instance. We are looking for some master data about vendors that deliver material X. The system takes the user directly to SAP R/3's transaction for displaying vendors with certain characteristics.
- **"Created orders for plant Y"**
  Even if this query is a verb phrase, we do not take the user to some create order transaction. As the phrase is in the past tense and there is a reference to a particular plant, we assume that the phrase type is report. We then search for reports that combine orders with plants, where "orders" is interpreted in light of the user profile and the current scenario.

We are also considering giving the users a list of alternatives, as is done when people search for documents in document management systems. The search process would be the same, but we add a separate window listing all elements relevant to the particular query. This makes it easier for the user to find the intended transaction or report, but also adds one more step to the whole process of accessing the ERP system.

5. Related Work

The Model-Driven Business Management environment is related to CASE tools with executable models (e.g. [HP98]). Whereas these environments generate executable code from user-defined models, our models are linked to pre-implemented executable modules. Since the MDBM approach is restricted to the domain of pre-implemented, customizable software packages, it has been possible to employ user-friendly business models that can afterwards serve as the user interface to the ERP solutions. This model-based user interface is an alternative to the Web ERP platforms provided by TopTier and others [Ma99, To01].

For the ERP market, there are some modeling environments that also concentrate on building up business models of the underlying ERP system. The ARIS toolset is widely used in SAP projects and offers standard modeling languages for modeling functions, data

structures, organizational structures, and user interface aspects [Sc98]. LiveModel from Intellicorp has many of the same features, though Intellicorp also allows you to instantiate business models from the SAP R/3 log. However, their modeling basis is too weak for full-fledged business modeling and is today limited to customization activities.

Many of the ideas of this component stem from Brasethvik's document management system [BG01]. A lexicon is combined with model representations of the relevant domain. The search is raised to a conceptual level by taking into account stemming, synonyms and linguistic relations like generalization and specializations. Our system has a different purpose, as the users are looking for a particular transaction or report. Our system should enable them to do something, not just inform them about something. We need to distinguish between different result types (transactions, business data, help information, etc.), which requires a deeper approach to the mood of the sentence. Recognizing phrase types and intended instantiation levels allows us to predict why the user is posting a particular query.

There are several approaches to semantic retrieval that can be compared to our solution [Ar99, Sp99, St99]. With the use of WordNet [Fe98], for example, documents may be indexed with synsets that encompass certain semantic meanings (e.g. [Go98]). For our application, though, the semantic relations in WordNet are far too general to be of any use. The ERP terminology is very specific, and the domain-specific relations are far more important than the general ones. More promising is the FrameNet project at Berkeley, which tries to define domain-specific frame-semantic descriptions of English lexical items [BFL98]. Parts of their lexicon overlap with our extended business model. Whereas their lexicon is a general purpose lexicon for a particular domain, our model is geared towards the use of particular computer systems. Hence, we do not need to go into all the semantic details, and our model does not intend to be as complete as a real semantic lexicon.

6. Conclusions and Further Work

The MDBM concept is still under development. Even though some preliminary work has been carried out, there are many problems at both the modeling level and the linguistic level that have to be addressed. We are in the process of setting up an SAP laboratory that will allow us to start integrating our components with SAP R/3. Before doing that, though, we need to work out business models that accurately reflect the whole functionality of R/3. Domain-specific semantic relations must also be added to our lexicon.

The linguistic part of the environment is currently being validated. Provided that the overall approach explained in Section 4 works satisfactory, we will start building up the search engine and the indexer independently of the rest of the application.

At the moment, we concentrate on semantic lexicons and very simple query interpretations. This is sufficient for invoking ERP transactions and traversing the business model. For more complex retrieval of ERP business data, however, we need to parse the queries and match the resulting structures with the underlying structures of the ERP database. This is a non-trivial task, as a query can easily involve a number of tables and the checking of several field values. For example, the query *"give me a vendor that can deliver material X to plant Y and is rated among the 10 best vendors"* maps onto 4 different tables and a verification of the vendors' evaluation points.

Bibliography

[Ar99] Arampatzis, A. T., T. P. van der Weide, et al. "Linguistically Motivated Information Retrieval." University of Nijmegen Technical Report CSI-R9918, 1999.

[BFL98] Baker, C. F., C. J. Fillmore, and J. B. Lowe. "The Berkeley FrameNet project" In *Proceedings of the COLING-ACL*, Montreal, Canada, 1998.

[BG01] Brasethvik, T. and J. A. Gulla. "Natural Language Analysis for Semantic Document Modeling." *Data & Knowledge Engineering*. Elsevier Science, 2001 (forthcoming).

[Fe98] Fellbaum, C. *WordNet - An Electronic Lexical Database*. The MIT Press Books, 1998.

[GAV00] Gordijn, J., H. Akkermans, and H. Van Vliet. "Business Modelling Is Not Process Modelling". In Liddle, Mayr, and Thalheim (Eds.), *Conceptual Modeling for E-Business and the Web, ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and the World Wide Web and Conceptual Modeling*, October 2000, Salt Lake City, pp 40-51.

[GB00] Gulla, J. A. and T. Brasethvik. "On the Challenges of Busines Modeling in Large-Scale Reeningeering Projects." In Proceedings of the Forth International Conference on Requirements Engineering (ICRE'00), pp. 17-26, Schaumburg, Illinois, June 2000.

[GM99] Gulla, J. A., and R. Mollan. "Implementing SAP in a Multi-Cultural Organization." In Eder, Maiden and Missikoff (Eds.), Proceedings of the 1st International Workshop on Enterprise Management and Resource Planning: Methods, Tools and Architectures (EMRPS'99), 127-134, Venice, November 1999.

[Go98] Gonzalo, J. , F. Verdejo, I. Chugur, and J. Cigarrán. "Indexing with WordNet synsets can improve text retrieval." In *Proceedings of Coling-ACL'98*, Montréal, August 1998.

[HC93] Hammer, M. and J. Champy. "Reengineering the Corporation." HarperCollins, 1993.

[He97] Hernandez, J. A. *The SAP R/3 Handbook*. McGraw-Hill, 1997.

[HP98] Harel, D. and M. Politi. *Modeling Reactive Systems with Statecharts: The Statemate Approach*. McGraw Hill, 1998.

[Ju88] Juran, J. M. *Juran on Planning for Quality*. The Free Press, 1988.

[Ma99] Marshall, M. "Web Applications Servers Give Green Light to ERP." InformationWeek Online, April 1999.

[Sc98] Scheer, A.-W. *ARIS - Business Process Modeling*, 2nd edition. Springer, 1998.

[Sp99] Sparck-Jones, K. "What is The Role of NLP in Information Retrieval?" in *Natural Language Information Retrieval*. T. Strzalkowski, Kluwer Academic Publisher, 1999.

[St99] Strzalkowski, T. "Natural Language Information Retrieval", Kluwer Academic Publishers, 1999.

[To01] TopTier Software. "An eBusiness Unification Platform for Enabling the Virtual Enterprise." From www.toptier.com/resources/eup2.pdf (accessed February 2001).

[VHW00] van Everdingen, Y., J. van Hillegersberg, and E. Waarts. "ERP Adoption by European Midsize Companies." Communications of the ACM, April 2000, pp. 27-31.