Association for Information Systems
AIS Electronic Library (AISeL)

ACIS 2001 Proceedings

Australasian (ACIS)

2001

Extension and Configuration of Reference Models for Enterprise Resource Planning Systems

Michael Rosemann
Queensland University of Technology, m.rosemann@qut.edu.au

Graeme Shanks
The University of Melbourne, g.shanks@dis.unimelb.edu.au

Follow this and additional works at: http://aisel.aisnet.org/acis2001

Recommended Citation

Rosemann, Michael and Shanks, Graeme, "Extension and Configuration of Reference Models for Enterprise Resource Planning Systems" (2001). *ACIS 2001 Proceedings*. 78.
http://aisel.aisnet.org/acis2001/78

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Extension and Configuration of Reference Models for Enterprise Resource Planning Systems

Michael Rosemann [a] and Graeme Shanks [b]

[a]School of Information Systems
Queensland University of Technology, Brisbane, Australia
m.rosemann@qut.edu.au

[b]Department of Information Systems
The University of Melbourne, Melbourne, Australia
g.shanks@dis.unimelb.edu.au

Abstract

Enterprise Resource Planning (ERP) Systems are comprehensive off-the-shelf packages that have to be configured to suit the requirements of particular organisations. Implementation tools that guide the project team through the configuration process support the system individualisation. A key component of configuration is reference models that describe the functionality and structure of the ERP system. This research in progress paper discusses the shortcomings of current ERP reference models. Proposed modifications to reference models include support for specific model configurations, the interdependence of these configuration decisions on other reference models and the subsequent consequences of these decisions. Reference models, once configured in this way, become enterprise-specific. In addition, configuration decisions need to be documented and used throughout the ERP system lifecycle. In order to facilitate understanding and explanation of reference model modifications, scenarios and design rationale are used to augment the reference models. A program of further research including tool support and empirical studies is presented.

Keywords

Enterprise resource planning systems, conceptual model, reference model, configuration management, design rationale, scenarios

INTRODUCTION

Enterprise Resource Planning (ERP) systems are comprehensive packaged software solutions that integrate organisational processes through shared information and data flows. They have evolved from packaged software to support material requirements planning (MRP) and manufacturing resource planning (MRP II). With additional functionality, including sales, finance, human resources and purchasing, today's ERP systems provide support for most commercial functions of any organisation. ERP systems are generic and have to be configured for specific organisations, industry sectors and countries (Klaus et al. 2000).

ERP vendors experienced rapid growth throughout the 1990s. The 1999 market for ERP software and services was estimated at US$17 Billion, and is forecast to grow to US$21 Billion by 2004 (AMR Research, June 2000). A few vendors dominate the ERP market. Market shares in 1999 were SAP 30%, Oracle 14%, Peoplesoft 7%, JD Edwards 5%, Baan 3%, and "Other" 41% (Gilbert 2000). ERP system implementation costs are often reported to be five to ten times the cost of software licenses (Davenport 2000): therefore organisations worldwide spent something like US$90-180 billion on ERP systems in 1999.

Configuration of ERP systems requires extensive knowledge of the function and structure of the system, together with a detailed understanding of the requirements of the organisation. In order to facilitate configuration and increase understanding of ERP system functionality and structure, ERP vendors provide reference models, typically in the form of business process, function, object, data and system organisation models. However, even the most popular solution for Enterprise Resource Planning systems (mySAP applications) still use classical modelling techniques (extended ER-models and Event-driven process chains) (Curran and Keller, 1998). These techniques do not take the special requirements of configuration management into account.

Though most of the Fortune 1000 companies currently use ERP Systems, the IS literature has ignored the conceptual problems related to the model-based configuration of Enterprise resource Planning systems. Overall, this area of research can be divided into requirements engineering for the *development* of Enterprise Resource Planning Systems (Brinkkemper, 1999; Daneva, 2000) and requirements engineering for the *configuration of Enterprise Resource Planning Systems*. The latter one is the focus of this paper. Theoretical contributions in this field are still the exception. As an example, Rolland and Prakesh (2000) suggest a map including ERP goals and objectives for the identification and evaluation of user needs. Gulla and Brasethvik (2000) suggest three process modelling tiers to manage the complexity of process modelling in comprehensive ERP Systems projects. Their functional tier dimension deals with the functionality of the Enterprise System. However, they do not discuss how to differentiate reference models in this tier.

This paper proposes that reference models be used actively within the configuration process. This requires from the system vendors that they document the system configuration potential in their reference models. To ensure that design decisions taken during actual ERP system configuration process are remembered, it is suggested that scenarios and design rationale are used to augment reference models.

The paper first discusses the process of configuration of ERP systems within the system lifecycle. Several possible extensions to reference models in order to support understanding the interdependence and consequences of configuration decisions are highlighted. The next section describes how configuration decisions may be explained using scenarios and design rationale. An approach to augmenting configured reference models is discussed in the fourth section. A program of further research, including tool support and empirical studies, concludes the paper.

CONFIGURATION OF ERP SYSTEMS

ERP System Lifecycle

The process of introducing ERP systems is significantly different from the well-known classical software engineering lifecycle (Figure 1). After a common phase of defining project objectives, analyzing the current situation and developing a business blueprint (to-be-model), ERP systems management contains the two main phases of selection and configuration (Kirchmer, 2000). Configuration management includes the individualization of the system and the development of add-on functionality. The configuration of an ERP system requires the comprehensive selection of relevant parameters, which easily can contain more than 1,000 alternatives and is in the case of SAP R/3 based on more than 8,000 tables (Gulla and Brasethvik, 2000). Most solutions support the project team with step-by-step guidelines that lead the project team through a configuration process. Similar to the classical development process, the configuration of ERP systems also has a requirements engineering phase. However, instead of developing conceptual models from scratch, requirements engineering for ERP systems consists of a continuous process of selecting and configuring given reference models.

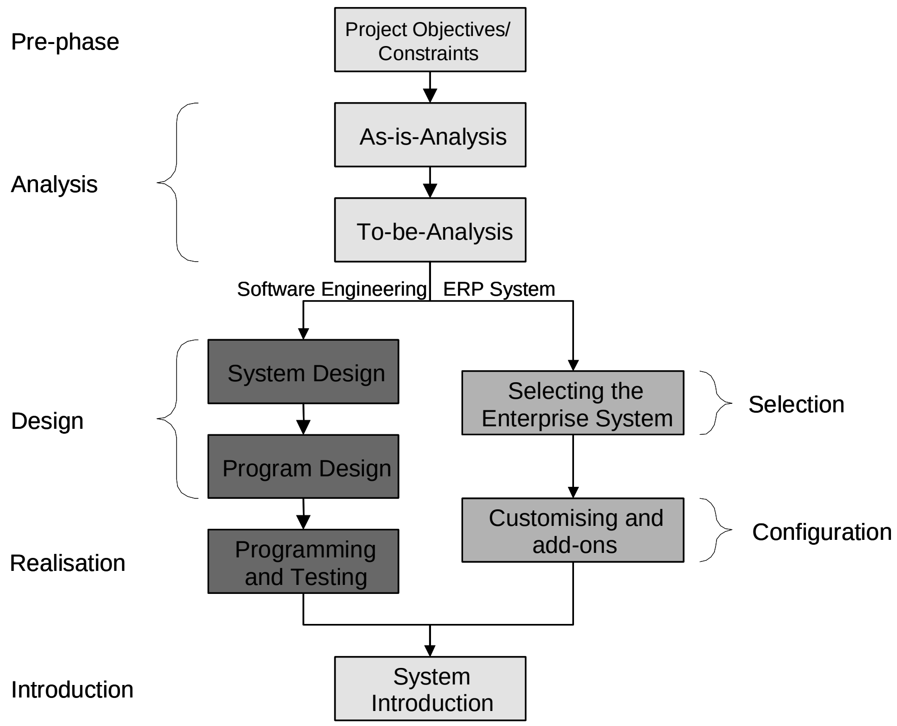


Figure 1: Classical software engineering and ERP systems implementation

ERP System Reference Models

ERP systems offer business solutions for typical functional areas such as Procurement, Materials Management, Production, Sales and Distribution, Financial Accounting, and Human Resource Management. These functions are often further individualized for different industry sectors including automotive, retailing, high tech, etc. Consequently, ERP systems tend to be very comprehensive and complex. As one approach to improve the understandability of ERP systems, since 1992 vendors have developed reference models.

ERP-specific reference models describe on different levels of abstraction the main processes and the data structure of the system. It is usually possible to refer to the relevant part of the online documentation and at the lowest level of abstraction even to the corresponding ERP transaction. It has to be stressed that these models are designed for both the end users of ERP systems and for the implementation team. End users will benefit from these models as they comprehensively and quickly inform about the related software functionality using business terms. The reference models are part of every ERP solution and do not have to be purchased separately.

Though reference models contribute significantly to the understandability of software functionality, they still have core weaknesses.

- As the models are focused on the description of the process execution and the data structure, it is not obvious what *configuration alternatives* exist. The analysis of a reference model shows what is possible in general, but not what might be recommended alternatives. Reference models represent the entire functionality from the viewpoint that the complete system is used. The models are not designed for configuration. Their modelling techniques do not support constructs that cover possible decisions during the implementation phase, i.e. decisions at buildtime. Thus, they do not differentiate between decisions on instance level and type level.
- Reference models concentrate on the elements that are of importance for a specific Enterprise System. *Enterprise-individual aspects* of the organisation, business objectives or manual tasks cannot be seen in these models. They do not include any references to the involved or required knowledge.
- Besides the missing transparency regarding possible choices during the configuration process, it is also not clear what *consequences* a configuration of one process or data structure has *on other processes or data structure*. An example of the interrelation between the configuration of Enterprise Resource Planning Systems processes can be found in Rosemann (2000).
- Moreover, the models do not have any *link to the actual process execution or database design*. Thus, it is not possible (e.g. in the form of model attributes) to see the process performance expressed in key performance indicators such as processing time or resource utilization.

Overall, reference models are a comprehensive and consolidated description of possible processes and data structures within an ERP system. This paper is concerned with how these models can be extended in order to depict configuration opportunities and explain configuration decisions. The assumption is that this will increase the value of these models and their acceptance. We focus the discussion on reference data models although the principles apply also to other types of reference models, in particular process models.

Configuration of Reference Data Models

ERP reference data models summarize on a conceptual level the data structures in an ERP system. They are typically based on modeling techniques that extend the classical Entity-Relationship approach suggested by Chen (1976) as they include, for example, the generalization-specialization relationship type. One of the most comprehensive reference data models is the SAP reference model, which includes more than 4,000 entity types. More than 180 business objects further cluster this data model (Curran and Keller, 1998).

Reference data models are particularly important for configuration decisions about the system organisational units as they depict precisely the given opportunities of an ERP System. A subset of the reference data model (approx. 30-40 entity types) allows a complete description of the interrelations between system organisational units such as company, factory, distribution channel or division. This facilitates especially configuration decisions that cover more than one module. Furthermore, reference data models are used within projects that aim to develop add-on solutions for ERP systems.

During the implementation of an ERP system such reference data models are customized corresponding to the actual system configuration. The following paragraphs discuss the main configuration decisions that can be made and how they could be depicted in reference data models. Extracts of the SAP reference data model are used as an example. The structure of this analysis follows the main constructs of Entity-Relationship-Models, i.e. entity types, relationship types and cardinalities.

Optional Entity Types

Transparent examples for model configurations related to optional entity types can be found in ERP Systems in the definition of the organisational structure. The Financial Accounting solution in SAP R/3, for example, requires a decision about the optional system organisational unit 'Business Area'. Business Areas in SAP R/3 are defined as "the organisational unit in external accounting that corresponds to a selected area of activity or responsibility within an organisation to which the value movements entered in financial accounting can be assigned." (SAP online documentation, 2000).

In a reference data model, optional entity types such as a Business Area could be highlighted with a dotted line. Figure 2 shows the two entity types Business Area and Cost Center. On the left side in the Figure 2, it is indicated that a Cost Center can refer to a Business Area. However, it is not clear, if this can be configured. The proposed dotted line indicates that a Business Area is not necessarily required. The decision about this organisational unit is compulsory as other processes depend on this decision. For example, the process of entering a Cost Center requires a reference to a Business Area for each and every Cost Center, if Business Areas are used (Rosemann, 2000).

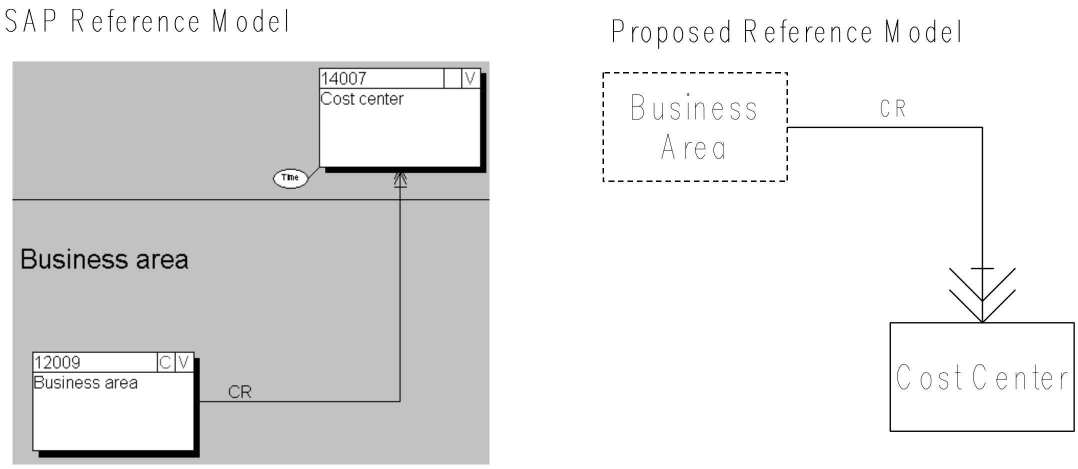


Figure 2: Optional entity types with a required decision

Other important examples for such optional entity types are specializations of organisational units. During the system configuration process, the relevant specializations of an organisational unit have to be selected. Other organisational units are optional and the entire configuration process does not necessarily require a decision about these constructs. In these cases, the system typically sets one instance of this organisational unit as a hidden default. If a decision is made to use many instances of this organisational unit, an entity type in the corresponding data model is required. An example in SAP R/3 is the Dunning Area. A Dunning Area groups in Financial Accounting-Accounts Payable all customers that are treated in the same way when in comes to sending out reminder notices. No other process outside the dunning sub-module depends on this decision. All customers are treated identically, if Dunning Areas are not used. The SAP reference model (Figure 3, left side) only highlights that a Dunning Area existentially depends on a Company Code. In order to highlight the configuration potential, it is proposed to highlight optional entity types that do not require a decision during the system individualization with two dotted lines (Figure 3, right side).

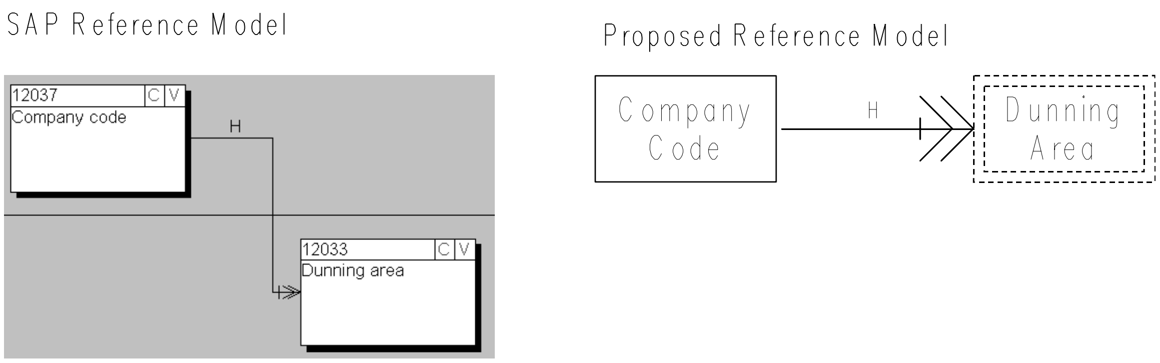


Figure 3: Optional entity types with an optional decision

Configuration of Relationship Types

The configuration of relationship types includes two decisions. First, if the relationship type is required at all. Second, what cardinalities the relationship should have. Optional relationships in ERP systems support an particular design alternative of cross-references. The more cross-references exist, the more intensively different modules and sub-modules are linked. On the other hand, intensive cross-referencing demands a good understanding of these interdependencies from the system users and can be perceived as restrictive in daily processes.

Optional Relationship Types

Consequently, examples for optional relationship types can be found in the accounting modules of ERP systems. An example is given on the left side in Figure 4, which shows the interrelationship between Profit Center and Cost Center in SAP R/3. The arrow and the 'CR' (conditional-referential) indicates that a 0,m-1,1 relationship exists, i.e. each Cost Center can refer to exactly one Profit Center. Each Profit Center consolidates the cost of zero to many Cost Centers. However, it does not become clear, if the decision to link a Cost Center to a Profit Center can be made at buildtime for all Cost Centers or only at runtime for each new Cost Center separately. The proposed model (Figure 4, right side) indicates clearly that this is a buildtime (configuration) decision. The dotted line indicates that this is an optional relationship type.

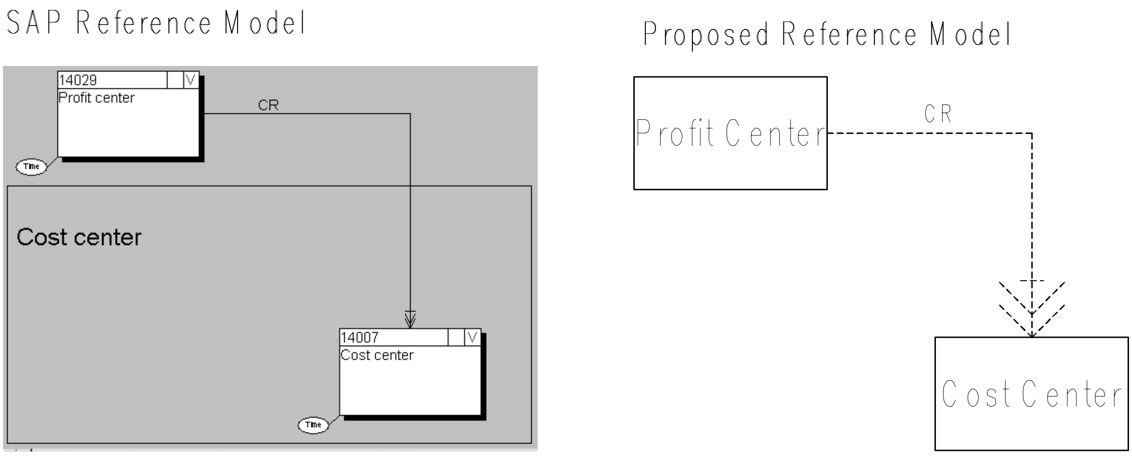


Figure 4: Optional relationship type

Further model extensions could be made that differentiate the default value for this relationship type in the case that no decision is made during the implementation process. This could be either the relationship does not exist or it exists and all Cost Centers have to refer to exactly one Profit Center. Optional relationship types that do not require a decision during the configuration process could be depicted with two dotted lines and would indicate the default value.

Configuration of Cardinalities

Decisions about the configuration of cardinalities can take place in conjunction with optional relationship types or independent from them. An example related to Figure 4 would be that after a decision that the relationship type exists, a decision has to be made, if a Cost Center definitely has to refer to a Profit Center or not. This difference can be expressed by the minimum cardinalities 0 and 1 and expresses a possible decision at runtime. In general, alternatives in ERP system configuration can be related to minimum and maximum cardinalities. Optional cardinalities that do not have to be defined during the customizing process should have a default.

Regarding the minimum cardinalities, a decision has to be made between 0 and 1, whereas 0 indicates that at runtime an entity of the involved entity type does not have to take part in the relationship. The typical alternative for the maximum cardinality is 1 and many. An example for such a configuration alternative is given in Figure 5. This example from SAP R/3 shows the interrelation between Company Code, the highest reporting unit in Financial Accounting, and Controlling Area, the highest reporting unit in Cost Management (Figure 5, left side). A variable is proposed to represent the maximum cardinality and in order to express that a Controlling Area either corresponds with exactly one Company Code ($x = 1$) or it covers more than one Company Code (x = many). This is a mandatory decision during the configuration of SAP’s accounting solution (see attached screenshot from the SAP configuration of the Controlling Area). Again, the existing SAP data model only includes the maximum case, i.e. x = many. The actual configuration opportunity does not become obvious.

Interrelationships between configurations

All the configurations above were local customizing decisions, i.e. each configuration could be made in the local context of the involved entity types, relationship types and cardinalities. Further complexity is added, if also interrelated configurations are analysed. These interrelations can again be differentiated in mandatory and optional interrelationships. They require a combination of the suggested modifications of existing reference modeling techniques. Pointers would be useful in these cases in order to indicate existing dependencies. Selected examples in the SAP R/3 system are the following.

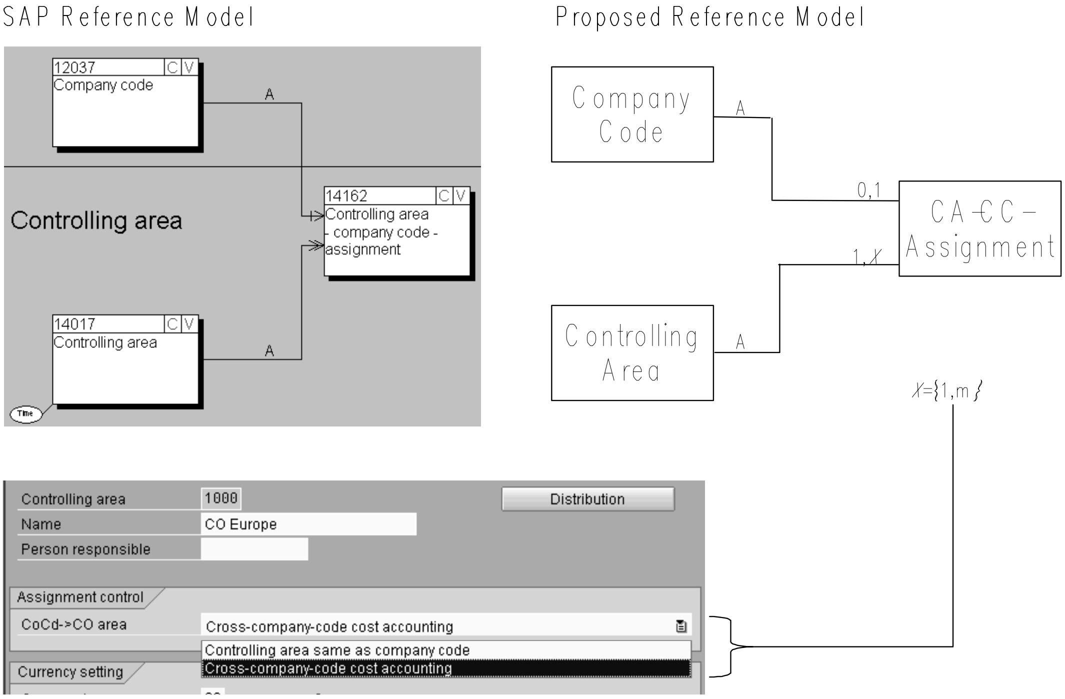


Figure 5: Configuration of cardinalities

- The decision to use Profit Centers in the Cost Management module can lead to not using Business Areas in Financial Accounting as Profit Centers have comprehensive reporting opportunities (optional interrelation). In this case, a decision about one entity type influences the existence of another entity type.
- Company Codes in Financial Accounting and Purchasing Organisations as the highest reporting units in Materials Management can be linked in SAP R/3 either directly or via individual Plants. In this case, a decision about one relationship type influences the existence of another relationship type.
- A key configuration decision regarding the linkage of Materials Management and Financial Accounting has to be made regarding the level on which material should be evaluated. This can be either the level of the legal entity (Company Code) or each individual factory (Plant). Thus, the decisions about the Company Code-Material Valuation Area relationship type and about the Material Valuation Area-Plant relationship type have to be made simultaneously. One of these cardinalities will be 1-1, the other one 1-many. In this case, a decision about one cardinality influences another cardinality.

The suggestions above proposed various ways of modifying the syntax of existing reference modeling techniques in order to capture the configuration potential. Besides an increased transparency about mandatory and optional configuration opportunities it is also necessary to adequately document the actual configuration decisions. The following section proposes how the pragmatic quality of reference models can be increased using argumentation-based design rationale and scenarios. The company code-controlling area-assignment will again be used as an example in order to show how configuration decisions can be described.

UNDERSTANDING CONFIGURED REFERENCE MODELS

A key issue with reference model configuration is that many different stakeholders in the ERP systems implementation process need to clearly understand both the reference models and the configuration choices made to the reference models. The configured reference model is the critical documentation that records how the ERP system has been configured and should work. Understanding of conceptual models has been long recognized as a key issue in the usability of conceptual models. *Explanation* and *visualisation* are two means for improving stakeholder understanding of conceptual data models (Shanks and Darke 1999). Gulla and Brasethvik (2000) discuss comprehensively the importance of the pragmatic quality of business process models in the context of a large SAP implementation.

Important knowledge about design decisions, assumptions, and argumentation, and about the details of how particular stakeholders intend to use the data represented in ERP reference models is gained during the configuration process. Although this may remain in the memories of those who participated in the configuration process, it is usually not recorded. This knowledge can be captured and used to assist with explanation of the model. Argumentation-based design rationale and scenario-based analysis are two mechanisms for capturing and retaining knowledge.

Capturing Configuration Decisions Using Argumentation-based Design Rationale

A number of alternative reference model configurations are generated, discussed, and evaluated during ERP system configuration, which is a creative design process. The models and associated discussions and design decisions constitute the design reasoning or argumentation. Design rationales are typically represented as explicitly structured discussions about the design artefact and are "... representations of the reasoning behind the design of an artefact" (Buckingham Shum and Hammond, 1994). They support the building of cumulative design knowledge and aid reasoning, communication, and critical reflection about the process and the design, and they are an important resource for reuse and redesign processes (Fischer et al. 1991, MacLean et al. 1991).

Structure-oriented and process-oriented techniques constitute the two main categories of design rationale (Dix et al. 1993). Structure-oriented techniques are intended to be used *after* the design process. They focus on the logical structure of the space of all design alternatives. Process-oriented design rationale techniques focus on maintaining an historical record of design decisions and are intended to be used *during* the design process. The Issue Based Information System (IBIS) and its descendants are examples of process-oriented design rationale techniques (Conklin and Burgess Yakemovic 1991). There are two types of process-oriented design rationale approaches: those that represent the design discussion only, and those in which the design rationale is integrated with the artefact itself as it evolves. Empirical studies suggest that integration of the design discussion with the artefact is preferable as design is focused to the task at hand and large and unusable documentation is avoided. Shanks and Darke (1999) found that design rationale was an effective means of explaining the evolution of concepts in the design of a corporate data model. Simple design rationale notations are preferred, as more expressive notations with sophisticated computer support are too difficult and time-consuming to use. An example of a design rationale fragment, represented using Question/Answer/Reason notation (Shanks and Darke 1999) is shown below in Figure 6. This example refers to design decisions taken about the configuration of cardinalities as shown in Figure 5.

Q Does a Controlling Area correspond to many Company Codes?

A Yes, the financial controller recommended that we select this configuration and hence we have a cardinality of many on the relationship.

R Our accounting standards recommend that all companies within our organisation are consolidated for cost management purposes.

Figure 6: Design Rational Fragment

Capturing and Explaining Configuration Decisions Using Scenarios

A scenario is "... a concrete description of an activity that the user engages in when performing a specific task" (Carrol et al. 1995). Scenarios are informal representations of specific instances of work driven tasks. These may be used in various forms (e.g. text descriptions, cartoons, videos) at any level of detail. They are useful for relating abstract, generic concepts to the everyday activities with which the user is familiar.

Scenarios may take either the *envisioner* role or the *evaluator* role (MacLean and McKerlie 1995). In their envisioner role, scenarios can be used during both the “as-is” and the “to-be” phases of the ERP system lifecycle. They can be informal, vague, open and inconsistent in order to support development of an understanding of the Business Area and the relevant users' requirements. In their evaluator role, scenarios can be used to assist the evaluation of configured ERP reference models and to help explain their meaning to all potential stakeholders. These scenarios need to be clearly and carefully grounded in the detail of the ERP reference models. They are an important component of the documentation of the models and of training programs that explain them.

Empirical studies have shown that scenarios are effective for both the design and explanation of conceptual models (Potts et al. 1988). Shanks and Darke (1999) found that envisioner scenarios were particularly effective in establishing requirements for corporate data models. They also found that evaluator scenarios were very effective in explaining the context in which business users would use the information in the models. Both types of scenario were readily understood by the business users and facilitated communication between them and the information technology staff. An example of an evaluator scenario is shown below in Figure 7.

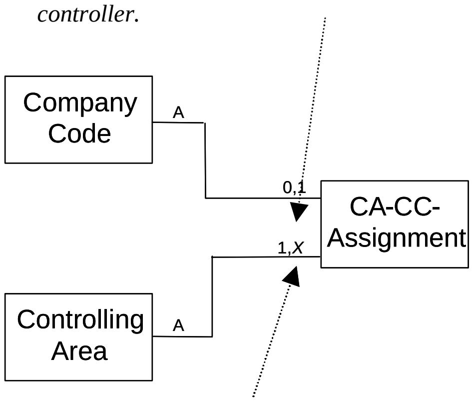
> *When monthly financial accounting reports are produced, we consolidate accounts for each company and distribute to the financial controller and analysts at each of these companies. However, for cost management purposes we provide an additional report that consolidates all companies and send the report to the group financial controller.*

Figure 7: Example of Evaluator Scenario

AN APPROACH TO AUGMENTING CONFIGURED REFERENCE MODELS

We propose the use of design rationale and scenarios to augment ERP reference models during configuration. Scenarios in the form of simple text blocks that describe desired system behaviour could be added to configured reference models to aid understanding by all types of stakeholders. Design rationale fragments in the form of "question/answer/reason" fragments can also be added to reference models to explain the reasons for configuration of the reference models. This additional information is expensive to collect and store (Shanks and Darke 1999), but as configured reference models are intended to be the shared documentation of the configured ERP system, the expense may be justified if reference model understanding is improved. In order to be most effective, both design rationale fragments and scenarios need to be linked to elements in configured reference by hypertext links. Figure 8 below shows how design rationale and scenarios could augment the ERP reference model fragment from Figure 5.

When monthly financial accounting reports are produced, we consolidate accounts for each company and distribute to the financial controller and analysts at each of these companies. However, for cost management purposes we provide an additional report that consolidates all companies and send the report to the group financial controller.



Q Does a Controlling Area correspond to many Company Codes?
A Yes, the financial controller recommended that we select this configuration and hence we have a cardinality of many on the relationship.
R Our accounting standards recommend that all companies within our organisation are consolidated for cost management purposes.

Figure 8: Example of Augmented Reference Model

LIMITATIONS AND FUTURE RESEARCH PROGRAM

This research in progress paper has focused on reference *data* models. It argues that current ERP reference models are not able to depict the configuration potential of the underlying ERP package. Suggestions for required extensions to support better understanding of the interdependence and consequences of configuration decisions have been made. These suggestions covered the main constructs of ER-models, i.e. entity types, relationship types and cardinalities. The second part of this paper proposed scenarios and design rationale to capture configuration decisions. Thus, this paper covered the design (first part) and the configuration (part 2) of reference models.

At this stage, this part of our research is concentrated on vendor-specific methodologies as it uses reference process and data models in mySAP applications. This represents a major *limitation* of this work, though the general recommendations can easily be adapted to techniques used by other Enterprise System vendors.

A research program based on the proposed approach to ERP reference model configuration and documentation is planned. Various aspects of the program include:

- analysis of the completeness of the suggested configuration opportunities. Using a main sub-module in SAP R/3 (e.g. configuration of the enterprise structure or Accounts Payable), the consequences of the required system configurations on the reference data models will be studied.
- detailed specification of the process of ERP system configuration and the notations to be used in capturing and representing scenarios and design rationale;
- current research analyzes the configuration of reference data models and reference process models separately. Future work will investigate in linking both configurations and increasing the consistency and integrity of model-based configurations;
- design and development of tool support in integrating reference model documentation with scenarios and design rationale;
- analysis of different media types for representation of reference models and associated design rationale and scenario fragments;
- the acceptance and usefulness of extended and documented reference models will be studied with internal and external project members of current Enterprise Resource Planning Systems implementation and upgrade projects. These empirical results might motivate changes or further extensions of the suggested model-based configurations, scenarios and design rationale.

REFERENCES

AMR Research, Inc., “AMR Research Predicts Enterprise Applications Market Will Reach $78 Billion by 2004”, AMR Research, June 12, 2000, http://www.amrresearch.com/pressroom/files/00613.asp (accessed 21 May 2001)

Brinkkemper, S. (1999) "Requirements Engineering for ERP: Requirements Management for the Development of Packaged Software", *Proceedings of the 4th International Symposium on Requirements Engineering.* Limerick, Ireland, 7-11 June.

Buckingham Shum, S. and Hammond, N. (1994) "Argumentation-based Design Rationale: what use at what cost", *Int. Journal Human-Computer Studies*, Vol. 40, 603-652.

Chen, P. (1976) "Entity-Relationship Model: Towards a Unified View of Data", *ACM Transactions on Database Systems*. 1:1, 9-36.

Curran, T. and Keller, G. (1998) *SAP R/3 Business Blueprint*. Prentice Hall: Upper Saddle River.

Carrol, J.M. (1995) "Introduction: The Scenario Perspective on Systems Development" in J.M. Carrol (ed.) *Scenario-based Design*, John Wiley and Sons, New York, 1-17.

Conklin, J. and Burgess Yakemovic, K.C. (1991) "A Process-oriented approach to design rationale", *Human-Computer Interaction*, Vol. 6, Nos. 3 & 4, 357-391.

Daneva, M. (2000) "An Assessment of the Effects of Requirements Reuse Measurements on the ERP Requirements Engineering Process", *Proceedings of the 4th International Conference on Requirements Engineering.* Schaumburg, Ill., 19-23 June.

Dix, A. J. Finlay, J., Abowd, G. and Beale, R. (1993) *Human-Computer Interaction*, Prentice-Hall.

Fischer, G., Lemke, A., McCall, A.C. and A.I. Morch, A.I., (1991) "Making Argumentation Serve Design", *Human-Computer Interaction*, Vol. 6, 393-419.

Gilbert, A., “ERP Vendors Look For Rebound After Slowdown” *InformationWeek* February 14, 2000, http://www.informationweek.com/773/vaerp.htm (accessed 21 May 2001)

Gulla, J.A. and Brasethvik, T. (2000) "On the Challenges of Business Modeling in Large Scale Reengineering Projects", *Proceedings of the 4th International Conference on Requirements Engineering.* Schaumburg, Ill., 19-23 June, 17-26.

Kirchmer, M. (2000) *Business Process oriented Implementation of Standard Software*. 2nd ed., Springer-Verlag, Berlin, Heidelberg.

Klaus, H., Rosemann, M. and Gable, G.G. (2000) "What is ERP?", *Information System Frontiers*, 2:2, 141-162.

MacLean, A., Young, R., Bellotti, V. and Moran, T.P., (1991) "Questions, Options and Criteria: Elements of design space analysis", *Human-Computer Interaction*, Vol. 6, 210-250.

MacLean, A. and McKerlie, D. (1995) "Design Space Analysis and Use Representations" in J.M. Carrol (ed.) *Scenario-based Design*, John Wiley and Sons, New York, 183-207.

Potts, C. and Bruns, G., G, (1998) "Recording the reasons for design decisions", *Proc. 10th Intl. Conf. on Software Engineering*, 418-427.

Potts, C., Takahashi, K. and Anton, A., (1994) "Inquiry-Based Requirements Analysis", *IEEE Software*, March, 21-32.

Ramesh, B. and Jarke, M. (2001) "Toward Reference Models for Requirements Traceability", *IEEE Transactions on Software Engineering*, 27: 1, 58-93.

Rolland, C. and Prakash, N. (2000) "Bridging the Gap between Organisational Needs and ERP Functionality", *Requirements Engineering*, 5: 3, 180-193.

Rosemann, M. (2000) "Using Reference Models within the Enterprise Resource Planning Lifecycle", *Australian Accounting Review*, 11:2, 19-31.

Shanks, G. and Darke, P. (1999) Improving Understanding of Corporate Data Models, Information and Management , 35, 19-30.

COPYRIGHT

Michael Rosemann and Graeme Shanks © 2001. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.