

11 Templates: Standardization for Business Networking

Thomas Huber, Rainer Alt, Günter Lehmann

11.1 Introduction	212
11.2 Definition and Approaches to Standardization	213
11.2.1 Definition and Dimensions of Standardization	213
11.2.2 Requirements of Inter-process Integration	214
11.2.3 Approaches to Close the ‘Organization Gap’	215
11.3 Template Handbook	216
11.3.1 Idea of a Template Handbook	217
11.3.2 Components of a Template Handbook	217
11.3.3 Activities in Template Design and Roll-out	218
11.3.4 Who Should Design and Use a Template Handbook?	219
11.4 Template Handbook at the Robert Bosch Group	220
11.4.1 Development of the Template Handbook	220
11.4.2 Overview and Experiences	221
11.4.3 Example Documents	223
11.5 Benefits of Templates in a Pharmaceutical Company	225
11.6 Conclusions	227

11.1 Introduction

Business in the information age depends to a large extent on the availability of accepted standards [cf. e.g. OECD 1996]. They significantly reduce the coordination requirements between business partners and are the basis for establishing integrated information flows. The availability of Internet standards, such as HTML or TCP/IP, has been a main enabler in the evolution of electronic commerce (EC) since the early 1990s. In providing specifications for the interconnectivity of information systems, which stretch across system platforms and user communities, they allowed the integration of heterogeneous sectors within an economy. Generally, we can distinguish two main thrusts towards standardization: business requirements and technological integration [cf. Venkatraman 1991].

Business requirements have changed fundamentally in the information age: large, established conglomerates are breaking up, innovative companies are emerging rapidly, and customer and process-orientation is becoming increasingly critical. Companies such as Walmart or Amazon.com derive competitive excellence from a thorough management of information and goods across all instances in their value chains. The management of these relationships is at the heart of Business Networking (cf. Chap. 1). From a technical perspective, this requires integrated information flows across all partners. This means that planning, order and control information need to be visible in real time within the entire business network.

Although the last decades have brought about significant technological improvements, today's information systems are still a far cry from this network-wide real-time visibility of information. The diffusion of enterprise resource planning (ERP) systems, such as SAP's R/3, Baan's ERP or Peoplesoft's BPCS has led to a higher level of internal integration (cf. Chap. 2.3.1). These systems are now being extended to include external partners. However, taking a closer look at the status quo of internal integration, we find that ERP systems are fragmented and distributed especially within large, multinational companies. The reason is in the high complexity of every ERP implementation project and the investments required in terms of time, capital, and manpower. Consequently, various individual ERP systems have evolved for different production sites, sales offices etc. Although, the distributed systems may use the same software products (e.g. R/3), they usually are configured individually with processes and master data being set-up differently. Consequently, the exchange of information between systems is difficult and specific interfaces are needed for establishing an integrated flow of information.

Standards for the configuration of the individual systems provide significant benefits in two dimensions. Firstly, the standardization of data, functions and processes leads to integrated information flows and permits improved processes, e.g. global credit limit checks, supply chain management and improved drill-down possibilities in financial reporting. Secondly, scale effects in the implementation allow reductions in the cost and time required to configure an ERP system.

11.2 Definition and Approaches to Standardization

11.2.1 Definition and Dimensions of Standardization

A broad variety of standards have developed in the field of information technology. Standards are available at various levels and originate from different standardization bodies. Although it would be desirable from a technological standpoint to have a single standard accepted worldwide, the variety also reflects the heterogeneity of the individual industrial sectors and the different power constellations within industries [cf. Brousseau 1994]. In the following, we will define standards as *objects that are shared and accepted within a specific community*.¹ Therefore, standards have three important dimensions (see Figure 11-1):

- The *objects* of standardization are hardware or software components with the latter comprising standards for data, functions, and processes.
- *Communities* are organizational and geographical in nature. Standards can be accepted within a single organization or between multiple organizations. From a geographical perspective, national, international, and global standards are distinguished [cf. Cannon 1993].
- *Standardization bodies* are closely related to the community and include private companies (e.g. Microsoft, IBM), industrial organizations (e.g. X/Open), national (DIN, BSI), international (ETSI, EWOS) and global (JTC1, ITU, IETF) bodies.

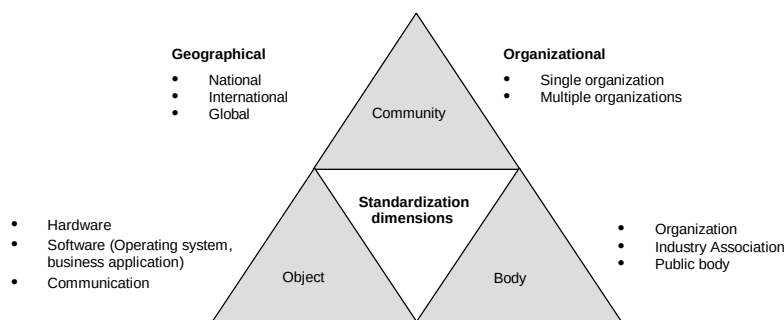


Figure 11-1: Dimensions of Standardization

The standardization dimensions are useful in analyzing standards that are on the marketplace. Standards which have had an impact in the marketplace concern

¹ This definition uses elements of ([Cargill 1989] and [Buxmann 1996]).

proprietary or ‘de facto’ approaches to system platforms, such as the PC or DEC VAX, to operating systems such as DOS or Windows, to communication such as IBM SNA or DECNet, and to business software, such as SAP or Baan. Open standards are not owned by any single business or vendor and include such approaches as X/Open, the OSI model, and the Internet standards [cf. Chesher/Kaura 1998]. Starting from the late 1980s, the focus of businesses towards standardization has changed with the growing diffusion of open standards towards the upper layers (i.e. layers 4 to 7) within the ISO/OSI model (see). Moving to the higher levels mainly implied that interconnections among systems were taken for granted and that applications and the contents exchanged were becoming critical issues.

ISO/OSI-Layer	Examples
7. Application Layer	X.400, SMTP, DCA/DIA
6. Presentation Layer	X.400, ASN.1, FTP
5. Session Layer	X.225, X.224, ISO 8326/27
4. Transport Layer	X.225, X.224, TCP, ISO 8072/73
3. Network Layer	X.25, IP
2. Data Link Layer	X.25, ISO 8802, Arpanet, Ethernet, Token Ring
1. Physical Layer	X.25, ISO 8802, Arpanet, Ethernet, Token Ring

Table 11-1: Standards in the ISO/OSI Model

11.2.2 Requirements of Inter-process Integration

Integrating information systems means establishing communication between these systems. From the perspective of the well-known communication model, a common language is required which has four levels (see Figure 11-2). As long as communication involves human participants, there is a high degree of flexibility in interpreting the intended meaning. However, this form of integration is hardly efficient in the context of ERP where productive and high-volume data has to be re-entered manually in the system. Therefore, the following discussion will focus on inter-process integration, i.e. the direct coupling of applications. An outstanding example of for inter-process communication is EDI. Information is not only transmitted electronically, but also processed automatically in the receiving IS. Other forms of communication, such as human-computer interaction, are not within the scope of this Chapter.

Inter-process integration requires that all aspects of the communication should be identical in both systems. As [Kubicek 1992] showed, the ISO/OSI model has to be extended to include all necessary aspects. In addition to the communication services that are still covered by ISO/OSI, another three layers are required which, in essence, are derived from communication theory [cf. Zbornik 1996, 96]. In the first place, a common syntax is required which defines the order, length and the type of data being exchanged. Semantics add meaning to the individual data fields, e.g. EAN codes not only specify the 13-character syntax but also make sure that each article has a unique identification number. By referring to this unique number, meaning is added since it always leads to the same article name. The pragmatic element is optional and a feature of sophisticated workflow systems. It makes sure that transmitted data has not only been understood but that subsequent actions are triggered. For instance, the ERP would automatically issue an invoice once a product has been delivered.

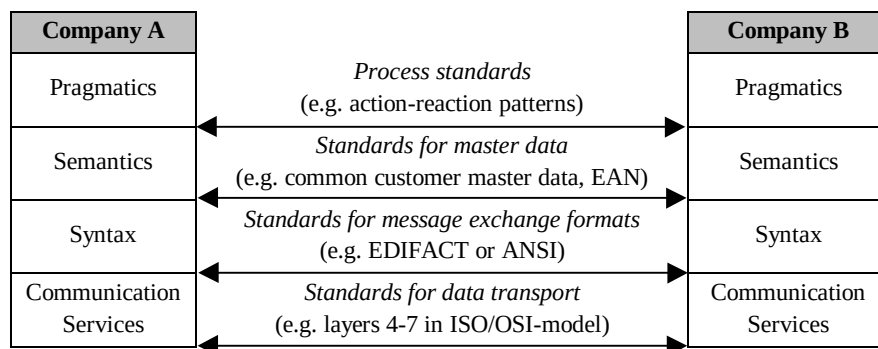


Figure 11-2: Standards in the Communication Model

11.2.3 Approaches to Close the 'Organization Gap'

Standards are the key to inter-process integration. In their early study, [Benjamin et al. 1990] reported that insufficient availability of standards has been the most important barrier to interorganizational integration. Up to date standards are mostly available for communication services and on the syntactical level (see Figure 11-2). Standardization on the semantical level has been set-up in some industries by industrial associations, e.g. the EAN codes in the food industry or the ISBN numbers for publications, or by independent providers, e.g. the Dun & Bradstreet (D&B) company numbers.² Standards on the pragmatical level are only available within companies and solutions which span across multiple organiza-

² Information on the D&B numbers (DUNS) is available at <http://www.dnb.com>.

tions still remain to be established. The neglect of semantical and pragmatical issues in such standards as EDIFACT has been referred to as ‘organizational gap’ [cf. Kubicek 1992].

Attempts to close the ‘organizational gap’ in inter-process communications have to address two aspects in general (see Table 11-2). First, the number and the type of systems involved are relevant. Normally, common semantics and pragmatics are no problem within a single ERP system. Additional systems may be other internal ERP systems or external systems, e.g. the ERP system of a supplier. The second dimension distinguishes two possibilities of standardizing semantics [cf. Scheckenbach 1997]: either the same codes and master data are used or matching tables are established ex ante which interrelate the individual codes. Clearly, using identical objects is a more efficient and pervasive way of integration, but requires the restructuring or re-customization of existing systems as well. A solution for the tight integration of ERP systems will be described in the following (shaded area in Table 11-2). Other possibilities are proposed and are being developed in the EDI community (e.g. basic semantic repository, automated negotiation), but these will not be dealt with in this book [cf. Lehmann 1996].

	Matching Tables	Same Objects
Single ERP System	—	Joint master data
Multiple ERP Systems	Converters	Network code
External Systems	Matching service (e.g. the translation engine from GEIS)	Industry code (e.g. D&B numbers)

Table 11-2: Options to Close the ‘Organization Gap’ in Inter-process Communication

11.3 Template Handbook

A solution for achieving standardized objects among multiple ERP systems is presented in the following. This template handbook is intended to provide a method for the semantical integration of individual ERP systems which is apt to be of substantial benefit in facilitating implementation and improving business process performance. The template handbook was developed in regular workshops and represents a pilot project with Robert Bosch GmbH in Germany. As part of this project, numerous interviews and workshops were undertaken with representatives from both the IS and the application side.

11.3.1 Idea of a Template Handbook

The implementation of ERP systems is a major challenge for companies. For different technical and organizational reasons, multinational companies need many different ERP systems, e.g. different R/3 systems for plants and for sales offices. In view of the complexities of each implementation project, different ERP systems have evolved with incompatible processes and/or master data. Consequently, each business process is limited to a specific area.

Insights gained from different practical partner company projects shows that semantical standardization is one of the most important issues when implementing ERP systems in multinational companies. Semantical standards enable an automated information exchange between the many different ERP systems – Bosch, for instance, has more than 30 productive SAP systems – supporting its business processes. The problem is in that there are different methods which support the implementation of ERP systems (e.g. SAP's ASAP method), but not the *standardization*, of such systems. An organizational gap needs to be closed in this area as well.

System templates are a solution for defining semantical standards for various applications. Such templates make sure that all applications are provided with a common customizing and a common set of master data. This is achieved, if templates are developed centrally and then rolled out to the different systems.

11.3.2 Components of a Template Handbook

Templates are well known from office packages such as Excel or Word, and denote predesigned documents that contain formats, styles or formulas. Apart from office applications, there are templates used in two other areas. First, programming languages, e.g. C++, make use of libraries that offer frequently used functions, e.g. pull-down menus. Second, there are systems built with the aid of CASE-tools that are called application templates [cf. Hofman/Rockart 1994]. However, the latter only target the development of new systems and, as a consequence, do not address the organizational gap. Therefore, the standardization of data, especially of customizing data and master data in ERP systems, still remains a topic for research.

For this purpose, the project group has defined ERP templates as *concepts or models for the standardization of processes, functions, and data that could be implemented in a physical (ERP) system*. Templates may as well be a 'piece of software' which can be distributed automatically on different systems, or some paper documents, e.g. manuals, which have to be implemented manually. The major components of ERP templates are:

- Preconditions for the set-up of the ERP configuration settings (e.g. the settings in the SAP R/3 implementation guide),
- Preconditions and recommendations for the common definition of master data,
- Test procedures, demonstrations, training documentation,
- Maintenance procedures, and
- External validation from authorities, customers, ISO 9000 etc. (if required).

With a view to enabling and ensuring multiple re-use of templates from different organizational units, establishing a *documentation* is highly relevant. Such documentation should cover the following aspects:

- ‘Marketing’ documentation (What can the template do?)
- Functional description for the implementation team (documentation of the template components; how does the template work?)
- Implementation guide (For roll-out and localization; how to install the template?)
- Users guide (How to use the template?)

11.3.3 Activities in Template Design and Roll-out

The template handbook embodies a method for the development and the roll-out of templates. The main activities of the method are shown in Table 11-3. For each activity, a result document is produced which can be used in projects, and there are also techniques that help to fill out a result document. Examples of the main result documents and techniques are presented in Chapter 11.4.3.

Template Development	Template Roll-out
1. Organize template development 2. Document conditions for template deployment 3. Document template process 4. Analyze the output exchange of the template process 5. Define customizing settings (overview) 6. Define authorization concept 7. Define customizing settings (detail) 8. Document templates add-ons 9. Define data storage guides 10. Test template 11. Prepare functional description 12. Set up maintenance procedures 13. Write user documentation 14. Prepare implementation guide	1. Initial template selection 2. Implement template locally 3. Set up localization concept 4. Conduct local template tests 5. Local approval of the template

Table 11-3: Major Activities in Template Method

11.3.4 Who Should Design and Use a Template Handbook?

Templates can be used in large multinational companies as well as in smaller companies. However, the development process is very different. In larger companies, templates are designed to ensure minimal standards for the configuration settings in their ERP systems. Meanwhile, the development of a template method (or handbook) is mainly an IT driven project, whereas the development of a specific template (e.g. for the ‘make-to-stock’ processes in plants) calls for a profound understanding of the company’s business processes and the ERP system that supports these processes. Therefore, people from the different business units (for which a template is to be designed) and from the IT department of a company have to work closely together in order to define the contents of a template.

In smaller companies, templates are used in a very different way. Such companies need templates for faster – and therefore cheaper - implementation of the complex ERP software products. In such cases, the term ‘industrial sector templates’ is preferred. Such templates are pre-configured ERP systems that may be implemented in different companies of a specific industrial sector. The development of a template is not normally done in these cases by the company that will use a template, but by a firm of consultants or by the manufacturer of the ERP software.

11.4 Template Handbook at the Robert Bosch Group

11.4.1 Development of the Template Handbook

The Robert Bosch Group is headquartered in Stuttgart, Germany, and has operations in over 130 countries worldwide. In its four business sectors, viz. automotive equipment, communication technology, consumer goods, and capital goods, Bosch generates approx. DEM 50 billion in sales and employees approx. 189,000 people (1998 figures). At Bosch, many different ERP systems support the business processes. Most of them, more than 30, are using SAP's R/3. ERP systems are in place for different organizational units, such as manufacturing facilities, sales offices, financial departments as well as for different geographical regions (e.g. Germany, USA, and Asia). With a view to enabling new business processes and reducing the costs for ERP implementation projects, Bosch decided to use templates for the:

- initial implementation of new ERP systems, and
- the deployment of new processes on existing ERP systems.

Regarding the dimensions of standardization (Table 11-1) the Bosch templates were defined to support the global operation within Bosch (standardization community). The standardizing body was QI (Querschnittsbereich Informatik) which is Bosch's central IT department and serves all its divisions. The objects to be standardized were configurations of the ERP systems as well as data structures and codes.

Because of all the experiences that Bosch had gathered in respect of its ERP systems, they were aware of the complexities besetting templates. Therefore, there was no doubt that an effective method would have to support the template development process. The development of the 'template handbook' (which in fact is the name given the method) was the subject of a joint project which involved people from CC iBN and up to seven people from Bosch. The people from Bosch came from its information management department and had a good knowledge either of the ERP systems or the business processes within different departments at Bosch. The template handbook was developed using the method engineering concept (cf. Chap. 14.1.3). The basic components of method engineering are:

- *Activities* which describe what has to be performed. The goal of an activity is to deliver one or several defined results. The procedural model of the method determines a sequence of activities.
- *Results* are produced or changed by activities. Various types of documents can be distinguished, such as tables, graphs or diagrams.

- *Techniques* which describe how to obtain a result. Every task has at least one corresponding technique. Techniques are guidelines for the production of results and can include best practices and lessons learned from prior projects.
- *Roles* which describe the responsibilities for a specific task. Within a method, people and organizational units assume certain roles which are a summary of activities from an actor's point of view.

11.4.2 Overview and Experiences

For Bosch, the two primary objectives in developing the template handbook reflect the motivations described above (cf. Chap 11.1). On the one hand, ERP systems should be capable of supporting Bosch's Business Networking requirements. An integrated flow of information between the individual ERP systems was the key to addressing this business need. On the other hand, the time needed for R/3 implementations was to be reduced through re-use of know how.

Table 11-4 gives a detailed view of the contents of the template handbook. Section A lists all the activities, techniques and result documents that were needed to develop a template. Section B comprises all activities, techniques and result documents needed to roll out a specific template to different ERP systems.

Activity	Technique	Result Document
A. Template Development		
1. Organize template development	Organization template development	Overview organization template development
2. Document conditions for the template deployment	Conditions for template deployment	List of conditions for template deployment
3. Document template process	Activity chain(s) template processes)	Activity chain diagram
4. Analyze the output exchange of the template process	Context diagram template process	Context diagram of the template process
5. Define customizing settings	Settings for customizing (overview) and authorization	Preconditions for customizing (overview)
6. Define authorization concept		Authorization concept

7.	Complete customizing settings	Settings for customizing (detail)	Preconditions for customizing (detail)
8.	Document templates add-ons		Overview template add-ons
9.	Define data storage guidelines		Documentation of storage guidelines
10.	Test template	Test of configuration settings	Test report
11.	Prepare functional description	Functional description	Documentation of template functionality
12.	Set up maintenance procedures	Maintenance procedures	Documentation of maintenance procedures
13.	Write user documentation	User documentation	Documentation for end user
14.	Prepare implementation guide	Implementation guide	Check list conditions for template deployment
			List of template add ons
			List of settings to be localized
			Change list for original objects
B. Template Roll-out			
1.	Select templates for local use	Local selection of templates	List of selected templates
2.	Implement template locally	Local template implementation	Experience report
3.	Adjust template locally	Localization	Documentation of localization
4.	Conduct local tests	Local tests	Local test report
5.	Accept and release template	Acceptance and release	Configuration settings

Table 11-4: Template Handbook Overview

The whole development process for the template handbook took about 7 months and is now (Spring 99) in the phase of practical testing with processes – that should run on different ERP systems within Bosch – being defined and documented using the template handbook. The main challenge was to define all the aspects of a template that must be documented in some kind of result document. The goal was to document the entire template life cycle, thus covering development, roll-out and distribution, maintenance, release management, and re-use. The second challenge was to align all the activities that were necessary to produce the

result documents. It took about ten group meetings to define the content of the result documents and the activities needed to build these documents.

11.4.3 Example Documents

As mentioned above, the template handbook specifies a result document and a technique for each activity. Both will be illustrated using the first activity of the method ('organize template development' Table 11-4). The document 'overview organization template development' is shown Table 11-6. The guidelines required to complete the result document fields are specified in the technique which is termed in analogy to the activity 'organization template development' (see Table 11-5). The following paragraphs show the constituents of this technique. All other activities, result documents and techniques of the method are documented as in the example discussed in this Chapter. Altogether, the template handbook now consists of approx. 70 pages.

Technique: 'Organization template development' The first step of the method deals with the identification of the organizational framework needed for the development of a template. The following information is important: • Template Name: Definition of a unique name for the template. The name should reflect the future function and the scope of the template. • Kernel System: Definition of the operating environment for the template. Here the decision is made on what types of systems the template will run (e.g. plant or distribution systems). • Application Type: Detailed specification of the type (manufacturer and application name) of the application for which the template is to be developed (e.g. SAP R/3). • Release: It is important also to specify the exact release number of the application type (e.g. 3.1.h). • Process-/Function Number: Use, if available, the identification numbers for functions provided by the manufacturer (e.g. the SAP function numbers). • Development Board: In this row are defined the persons (and organizational units they belong to) responsible for the development of the template. • Review Board: Here the review board is defined (employees and organizational units). The review board has to crosscheck the contents of the template against the overall conventions that exist in a company (e.g. naming or programming conventions). • Authorization Board: This row defines responsibilities for official authorization (final acceptance) of the template. Accordingly, the names of the persons appointed to the board and their organizational units are indicated in this row. • Pilot Customer(s): Definition of the pilot customer(s) of the template. Again, the names of the persons appointed must be indicated and the organizational units they belong to. • Template Roll-out: Definition of the organizational units that must or could use the template. For each of these units, a decision is needed on whether the template must be implemented locally (enter 'M' for must) or whether the template could be implemented in a local system (enter 'C' for could) - if the unit wants to do so. • Localization: A decision is needed for each organizational unit that must/could implement the template on whether a localization of the template (e.g. changes in the customizing definitions in the template) is allowed (enter 'Y' for yes) or not (enter 'N' for no). • Copying Model: In a next step, a decision is made for each organizational unit on whether the template can be used as a copying model (enter 'Y' for yes) or not (enter 'N' for no). Copying model in these context refers to templates for an initial system set-up (this field is rather Bosch-specific). • Authorization Date: At the end of the development process, the template needs to be formally authorized (accepted). If accepted, the place and the date of the authorization are entered in this document, the authorization board signs the document and the template is ready for roll-out.
--

Table 11-5: Technique 'Organization Template Development'

Document: Overview - Organization Template Development **Date:** 12.12.99
Status: Completed **Author:** THU
Activity: Organize Template Development **Page:** 1

Template Name: Purchase Demand Release												
Kernel System	• Plant systems (all plants within Europe) • Sales systems (multi-divisional sale systems)											
Application Type	SAP R/3											
Release	3.1. h (and higher)											
Process/Function Number	BANF											
Template Boards	Organizational Unit				Name							
Development Board	• IS department • Plant representatives • Sales representatives				• A. Miller, F. Smith, C. Barns • H. Wilson, W. Walters • S. May, K. Muller, B. Gerber							
Review Board	• IS department • Plant representatives • Sales representatives				• H. Hansen • K. William • P. Klein							
Authorization Board	• CIO • Vice-president, plant management				• U. Lee • P. Guinness							
Pilot Customer(s)	Strasbourg Plant, France				J. Bartoli							
	Philadelphia Plant, USA				W. Clark							
	Munich Sales Office, Germany				B. Gross							
	Hong Kong Sales Office				E. Wu							
Template Roll-out (M = must, C = can)	Unit 1 Org.	Unit 2 Org.	Unit 3 Org.	Unit 3 Org.								
	M	C	M	M								
Localization (yes/no)	Y	Y	N	N								
Copying Model (yes/no)	N	N	N	N								
Authorization Date	Frankfurt, January 24, 2000											

Table 11-6: Example of a Result Document

11.5 Benefits of Templates in a Pharmaceutical Company

Another example of the deployment of templates in multinational companies is taken from the pharmaceutical industry. In the case of this company, the pressure to develop templates was a direct function of the growing globalization of its business. The company is organized in multiple divisions which are located in various parts of the world. In order to better manage the divisional businesses, the company decided to move from existing, historically grown, local systems to global systems. Contrary to the former, which are cross-divisional, the latter provide global information flows within a specific division. The main objectives of this strategy were to:

- Establish consistency and integrity of group reporting,
- Ensure consistency with divisional business strategies (process view, integration, globalization),
- Enhance project flexibility (improvement in terms of time, costs and risks), and to
- Foster opportunities for sharing and re-use of knowledge.

In order to achieve these results, establishing standards for processes, data and systems configuration's settings were a prerequisite. The necessary templates were developed by divisional competence centers for two areas. First, the company developed a company core that defines the common elements for all SAP R/3 implementations across geographical sites and divisions, e.g. group master data, chart of accounts, organizational elements, etc. This core guarantees the support of the group reporting rules, definitions and accounting principles and ensures consistent and efficient reporting. The uniqueness of data is ensured by assigning different number ranges to the divisions.

Based on this core, the company developed further templates for each division which were called 'divisional kernels' (see Figure 11-3). These templates are pre-configured ERP systems (mainly SAP's R/3) and reflect common processes in a division based on best practices. They are used for implementing systems within the different regional and local businesses of a division. The objectives of the divisional templates are to:

- ensure globally/regionally homogenous systems which facilitate different global/regional processes,
- support local process improvements, and to
- reduce implementation time and costs.

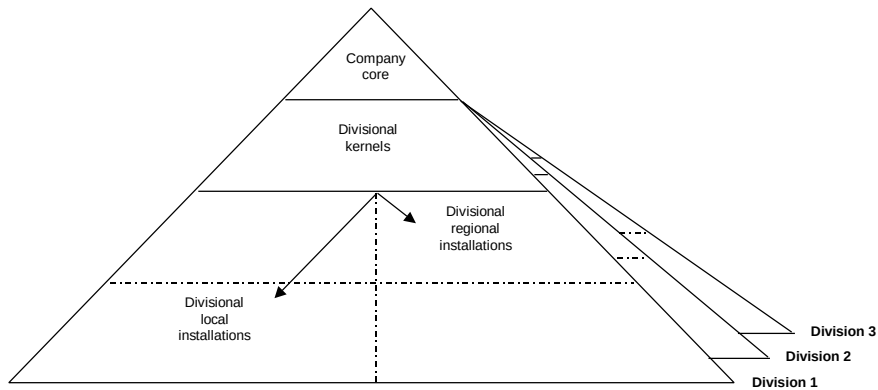


Figure 11-3: Distribution of Company Core and Divisional Kernels

As Figure 11-4 shows, between 40% and 80% of a local system's configuration is defined by templates (FI: financials, SD: sales and distribution, MM: material management, PP: production planning; FI, SD, MM and PP are modules of the R/3 system). Thus, templates were the important instrument to achieve global standardization among the different ERP systems. Having common, standardized systems in place tends to increase the efficiency of business processes and sustain strategic control and process improvements. Common systems and standardized implementations also tend to make for faster implementation and consequently, lower implementation costs.

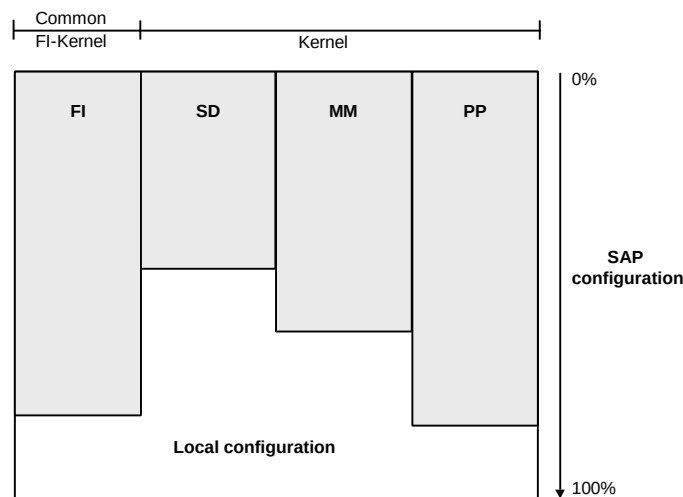


Figure 11-4: Pre-configuration Rates for Different Templates (in % of the Whole System)

11.6 Conclusions

In the present chapter, we emphasized the role of standardization in Business Networking and presented templates as instruments to achieve standardization of distributed ERP systems. As our project activities have shown, templates are apt to generate major benefits in two areas. First, templates sustain integrated information flows and foster new process perspectives such as one face to the customer as well as enhanced controlling and reporting possibilities. Second, templates are instruments to reduce the costs of implementing and coordinating distributed ERP systems inside large multinational companies. However, introducing, assessing and rolling out templates are complex processes which call for a systematic approach. This method we have termed 'template handbook'.

Besides the complexity of template development, other factors were found to have an impact on realization. First, templates presuppose the existence of application architecture planning. For example, a corporate decision would be needed as to whether local cross-divisional or global divisional systems are to be implemented. This degree of (de)centralization governs the design of templates. Templates with strong cross-divisional components, for instance, may differ from divisional templates. This may be due to the probability of different production strategies (make-to-order, make-to-stock etc.) being used greater in cross-divisional than in divisional production set-ups.

Second, templates are associated with significant scale economics since the costs of standardization are higher for pilot users. Every instance of the template being implemented will decrease its development cost and, due to learning curve effects, the cost of implementation. Consequently, organizations should strive for an as large a number of implementations as possible, making sure at the same time that pilot users are not penalized for participating in the development of templates. It is quite important, for the promoters of templates to consider creating incentives schemes which reward the pilot users.

Third, as all standards, templates have a strong political component. As [Brousseau 1994] pointed out, standards tend to be confronted with the dilemma as to whether universality or individuality should prevail. On the one hand, single, universal standards tend to yield the higher returns in terms of efficiency, whereas on the other hand, the different businesses within a company require individualized systems for maximal flexibility. For example, financial managers and forecasting departments would prioritize standardized and centralized systems. People in sales and production usually have a strong bias towards decentralized and individualized processes. In point of fact, the amount of individual fields and codes to be added to a template depends to a large extent on the political power of the individual business units.

Given the political power to implement standards and a strategy for the development and re-use of templates, the templates designed and the systematic method

applied have shown a practicable way of how to close the organization gap. At the same time, they will enable the course of further research to be charted out. First, continued research will have to focus on how future (ERP) application architectures that are geared to Business Networking will look like. A specially interesting aspect will be to determine which parts of these architectures will lend themselves to the use of templates. Second, the method is proposed to be amplified to assist in answering the question as to what the optimal degree of standardization will be that a specific template should provide. Finally, in view of the growing autonomy and interdependence of business units, it should be borne in mind that the possibilities of a hierarchical roll-out of templates are likely to diminish. Instead, it will be necessary to convince users to develop and use templates, and this will involve not only an appraisal of the costs but also a transparent demonstration of the benefits to be gained from investing in templates.

