

A FRESH LOOK AT INTEGRATED BUSINESS APPLICATIONS - NEITHER PACKAGE NOR CUSTOM-MADE!

Andrea Engelhardt
Bereich Wirtschaftsinformatik I (Department for Information Systems I)
Friedrich-Alexander Universität, Erlangen-Nürnberg
Erlangen-Nürnberg, Germany
Email: engelhardta@acm.org

ABSTRACT

Since integrated business applications demonstrate weaknesses, intermediate solutions positioned in between packages and custom-made software need to be explored. One suggestion for such an adaptable architecture uses a functionally stable core with peripheral applications. This architecture accounts for software market dynamics and interests of different parties involved. The ideas developed stem from a case study in the German market for small business software. The case describes the problems within this business configuration and presents a technical solution that implements a core application approach. The concept could turn out to be useful for other business networks, as first hypotheses hint.

Keywords

Software (CB), Organizational Characteristics (DA), Organizational Functions (DB), IS Management (E), Information Systems (H)

CURRENT PROBLEMS WITH INTEGRATED BUSINESS APPLICATIONS

Businesses have been replacing in-house developed business applications with packaged software for the last three decades. At present, it is fair to state that average German application portfolios are composed of 55% custom-made software² and 45% packages, with a continuing trend towards using more packages (BMFT 1994). Both types of business software hold some disadvantages, i.e.:

- Especially large integrated enterprise packages³ are often overloaded with functionality. An individual business might on the one hand only need certain parts of a package, but is nevertheless forced to buy it completely. On the other hand, the software might very well not fulfil special requirements. An abundance of functionality also contributes to a high degree of complexity, which then necessitates expensive expert advice to customise the system. The inflexibility of systems may cause the organisation to adapt to the package instead of the software adapting to the organisation.
- Individually developed software is expensive simply because there are no economies of scale, both during development as well as during maintenance. It has also been difficult to achieve a balanced skills portfolio within the IT department, covering a broad range of fast evolving technologies.

Packaged and custom-made software form the boundaries of a research field (figure 1): How can the information systems discipline systemise intermediate solutions in between these two extremes in order to facilitate organisations achieving and maintaining a desired level of application integration?

Integration has been a keyword in information systems research. At its core integration means the unification of actions towards organisational goals. It is the role of business information systems to facilitate this. Two aspects, which are instrumental in defining intermediate solutions, have been neglected in the integration discussion so far:

- (1) The fact that there is a dynamic software market. Integration needs to be achieved over time using software from different vendors with changing product releases.
- (2) The fact that there are business networks whose IS-strategies need to be supported. Integration within the traditional "model enterprise" (Becker/Schütte 1996, Mertens 1997, Scheer 1994) seems to be a too limited perspective, especially since business form tends to change - one enterprise might be absorbed by another or dispersed into smaller units.

It is the objective of this research to provide a framework for integrated business information systems that accounts for software market dynamics and the requirements of business networks through a concept unifying the advantages of packaged and custom-made software.

² This figure includes software embedded in products ("embedded systems").

³ Business software packages from vendors such as Baan, JD Edwards, Peoplesoft, SAP, etc.

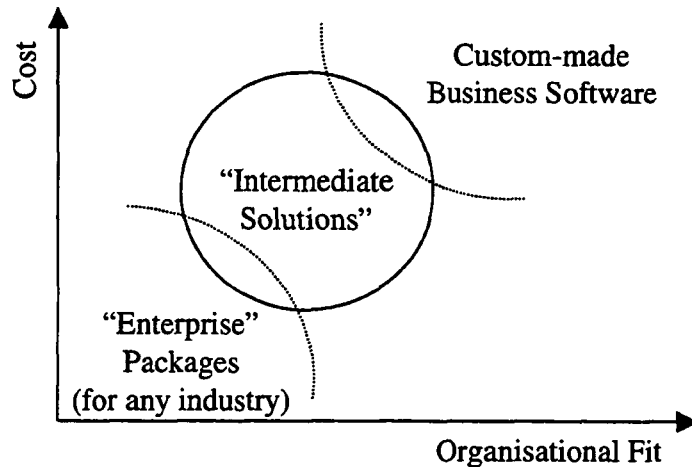


Figure 1: The research field between packaged and custom-made business software (adapted from (Meyer 1993))

Approach

At first the research field displayed in figure 1 is dissected using three characteristics - business size, industry, and integration technology.

- Differentiating by size leads to three segments - large, medium-sized and small businesses.
- Trying to contrast the software needs of different industries introduces the idea of horizontal and vertical software markets. Applications vary in their degree of industry neutrality. It is possible to identify core business systems, applicable in all industries; the so-called horizontal applications. Vertical systems subsume applications that are typically used in a certain industry, such as manufacturing, retail, etc.
- Recent software technologies facilitate application architectures displaying better integration quality than possible with traditional technologies.

Secondly, an in-depth case study describes one specific business network, which centres on software requirements for small businesses being served by various suppliers. Experiments with the various parties involved will assist in identifying strengths and weaknesses of the proposed solution.

Finally the concept is extended to other business networks sharing some of the features described in the case study.

CASE: SOFTWARE FOR THE GERMAN SMALL BUSINESS MARKET

Germany's small business count reflects the typical picture in advanced economies - most businesses are small and medium-sized. According to the official statistics⁴ (Bundesamt 1987) there are some 2.1 million businesses in Germany, 99.8 % of which employ less than 500 people and 94.7 % less than 20 people.

Abstracting from the diversity of small business, a few industry verticals are conceptualised. These verticals differ from those used for large companies, which IBM for example uses in order to focus its organisation (IBM 1995). A suggestion for seven small business verticals differentiates in construction, retail, hotel and food & beverage, craft, professional service, commercial service and health care.

The representation in figure 2 implies that there are more similarities of software requirements inside a vertical than across. Some business functions, however, cut across the vertical divisions. Most notably, every business encompasses an accounting function. Personnel administration is another example of a business function that can usually be found in different industry lines.

Mostly large software vendors offer solutions for these business functions, which can also be thought of as core systems. Further business functions, such as inventory management or production planning and control, could also offer a large enough software market volume to allow for core system developers to prosper, even if their

⁴ Recent publications show quite different estimates concerning the total number of businesses. The last reliable count dates from 1987, from which one has to extrapolate and also gauge the effects of German reunification.

core systems would not be useful in each industry vertical.

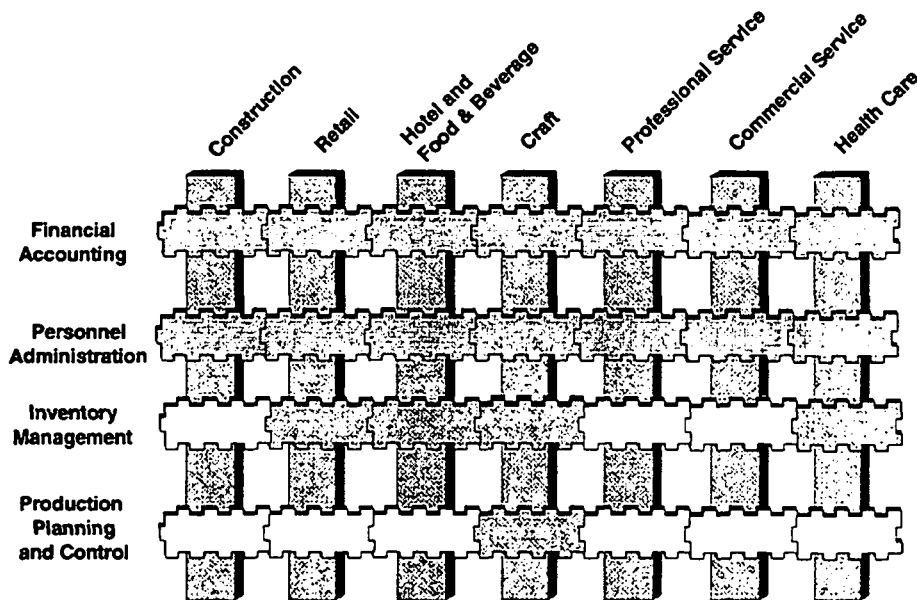


Figure 2: Small business verticals and some horizontal core business functions

The term “core” is to imply lack of completeness. The core needs to be augmented by other applications, in order to fulfil all of the software requirements of a business within an industry vertical.

The following case study presents a loose partnership between a large core system supplier and many small vendors complementing a core application to offer integrated solutions in a variety of industries.

The Role of the Tax Accountant as the Provider of a Core Application: Financials

The majority of small German businesses have outsourced the accounting function to their tax accountant.

The tax accountant industry is strictly regulated by law and professional institutions much like other professions such as medicine, law or architecture. Differing from most international practices, it is prohibited to run a tax accountancy as a commercial enterprise. The professional needs to apply his or her own expertise to every piece of work that leaves the office, so there is little room for delegation. This provision represents the main reason why tax accounting has been a small business industry in Germany and why there are no large German accounting firms, e.g. as they have developed in the UK or the USA.

The restrictive effects of regulation are offset by some favourable rules. The law stipulates a monopoly for certain types of work that solely a licensed tax accountant can perform. Most importantly this monopoly includes bookkeeping. The German lawmaker reasons that the task of bookkeeping frames a tax declaration. Therefore, in order to further the common good, this task needs to be performed by a trusted professional. This provision leaves small business with the choice of keeping their books themselves or outsourcing it to a tax accountant. For many small businesses, accounting does not merit employing someone full-time. Thus a high level of outsourcing can be observed.

The Role of the Tax Accountants' Data Processing Firm DATEV

At the end of the sixties, tax accountants were faced with the challenge to process the monthly declarations of large number of clients for the newly introduced value-added tax. It was time to give the profession access to data procession capabilities. One tax accountant implemented the idea of a data processing centre commonly owned by the members of the profession. He founded DATEV, a company exclusively serving German tax accountants, in the legal form of a co-operative.

DATEV has grown to be Germany's largest provider⁵ of financial accounting solutions, handling the accounts of some two million business entities, which comes close to almost every small business. Products range from entry-level to full-fledged PC-accounting packages complementing traditional mainframe-based data centre services.

Laws restricting accountants' activities extend to DATEV. This means that DATEV is bound to which business

⁵ 1996 sales of almost DM 1 billion, some 4.700 employees.

functions its software offerings may cover. Unless there is a dramatic change in the law, DATEV's solutions centre on accounting, personnel administration and taxes. This restriction to the accounting function is what makes the DATEV case especially interesting. Other vendors specialising in small business software might rather opt for increasing the scope of their packages to include functionality needed in other business functions. The most obvious example might be a function for invoicing. However, concentration on a specific function may very well be a matter of strategic choice. Integration technologies facilitate multi-vendor business software and allow software suppliers to develop and price competencies in a functional area. An example for this kind of strategic positioning is American software vendor Coda (Coda 1997). The DATEV case allows studying vendor-concentration on one business function without having to address the complexities of packages in the "big" business sector.

DATEV has to face an increasingly competitive software market that caters for the need of the small business sector. Some of the forces are the growing international presence of software vendors, offering personal finance packages that appeal to small business owners, as well as an increasing willingness of these businesses to try and use information technology.

DATEV has reacted to these trends by offering PC-based solutions that accountants can provide to their respective clients. Knowing that an accounting solution would only cover a portion of the potential software requirements, DATEV needs to offer seamless integration between its packages and the ones a client might use for other business functions, such as invoicing or inventory control. Software solutions with DATEV financials at its core should ideally offer an equivalent quality as integrated packages sold by a single source.

The Role of Small Software Vendors

The software industry mirrors the structure of overall German business. There are some large, few medium-sized and many small software companies. Some of the smaller software shops deliver business applications to their clients. Their limited resources make it difficult for them to maintain and update their own solutions, while at the same time tending to their customers' needs.

Small software firms feel the competitive pressures even more than a large business entity such as DATEV. They are looking for strategies to offer complete solutions while keeping their advantage of flexibly reflecting individual customer wishes in their offerings.

Traditional file interfaces that DATEV published to enable third-party software to link up with DATEV's accounting solution, were widely accepted within the small business software market. A DATEV initiative to implement rich interfaces to their PC-based software packages based on newly available technology could experience a similar market reception and develop into a quasi-standard, since it would allow to small developers to quickly integrate an accounting part into their applications.

The Role of the Client: The Small Business As Software User

The small business sector has gained attention, especially, since the studies of Birch (Birch 1987) in the US found that this segment contributes more to job growth than large business. Small business is thought to adapt faster and to know their niche markets better. Even if there are critical voices doubting the foundation of claims made by small business researchers (El-Namaki 1990) special government programs geared towards these businesses are wide-spread world-wide⁶. Current initiatives encourage more software use in general, and access to the Internet in particular.

The software market for the small business clientele is expected to grow. The generation change is one favourable influence, because the future leaders are thought to appreciate information technology more than their predecessors. Another factor is the changing cost structure for information systems – decreasing hardware prices allow for spending more on software and services.

Prototype Business Network Configuration

Put together in one picture (figure 3), the four parties need to align their interest in order to achieve integrated solutions for small businesses. Some features of this network can be found in other configurations. Solutions suggested for this configuration might be useful for other configurations (discussed later in this paper).

⁶ For an example refer to **Error! Reference source not found.** (02.08.1997), 5th Framework Programme For Research, Technological Development and Demonstration of the European Union, Section on "Innovation and participation of SMEs".

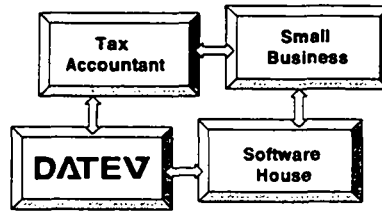


Figure 3: A prototype business network: Configuration with tax accountant, DATEV, software house, and small business

A SEMANTICALLY RICH WRAPPER OF A CORE APPLICATION

Concept

There are other technological approaches to reach intermediate solutions between packages and custom-made software. Two examples are the template-approach (Hofmann/Rockart 1993), which uses case-tool-based software designs to adjust an application to specific requirements or “frame technology” (Bassett 1996), where software consists of adaptable components. The ideas put forward in these approaches are helpful in rethinking the software “moulding” process. However, they address the problems of large, complex systems running within one organisation.

The initiative of the Open Applications Group (OAG), a not for profit organisation of large software vendors, also addresses the problems of large, mostly globally operating companies. The OAG has set out to deliver a standard that facilitates to link the business applications of various large producers. The OAGIS (OAG Integration Specification) defines integration points where applications exchange business object documents, which execute program functions in the receiver application (OAG 1995). An example of an implementation of the OAGIS are the SAP BAPIs (Business Application Programming Interface) (SAP 1997).

The solution described here differs in that it addresses smaller applications bought by smaller clients from a number of vendors in regional market. The proposed technical solution caters for the requirements of a software market. Coming from the case study just described the following “application architecture”⁷ is proposed.

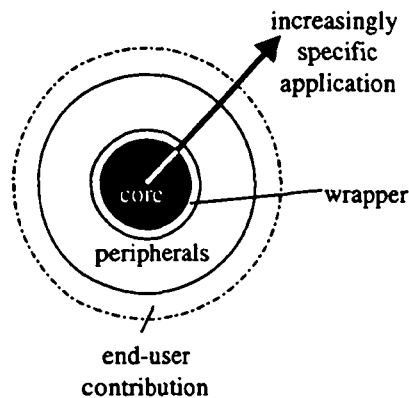


Figure 4: Peripheral applications integrated with a core application having a functionally stable wrapper

A functionally stable core application allows access to its inner workings by exposing a “wrapper”. The wrapper’s semantics stay stable over time, even if there are new releases of the application inside the wrapper, which then provides a frame of reference for independent software developers linking up with the core application.

In this case a functionally stable wrapper is designed for an accounting package taking up the role of the core

⁷ Software development is often likened to architecture. “Software” or “application architecture” could thus refer to the design of one application or to a model for a whole business. Both meanings of the word “architecture” use the analogy of designing a single structure; in this context the author would like to suggest to instead think of the work of a city planner, who would integrate several buildings.

application. The provider of this core application has to decide which elements to expose through the wrapper. Independent software developers supply "peripheral" applications. In this context "peripheral" does not mean less important, but rather expresses a higher degree of individualisation and adaptation to the requirements of a vertical or a specific client.

Next to supplying peripheral components, these independent software developers take up the role of component integrators. At this point in time it can only be speculated if there will be a "division of labour", where some software houses concentrate on supplying components and others specialise in integrating solutions built on them.

The application architecture is further enriched, as soon as end-users develop enough skills to exploit the possibilities of modern office packages and complement their application portfolio with highly individualised elements.

Implementation

Since the users in the presented case study are small enterprises, the proposed solution builds on PC-software platforms. Recently available technologies for application component integration open possibilities for effectively linking the strategies of a package vendor like DATEV with those of the developers of less widely distributed business applications. They allow exposing selected parts of an application package for outside access. These selected parts constitute the wrapper for the application within.

OLE-Automation lends itself as an integration mechanism. The part of the accounting application that is to be accessible to independent developers is compiled as an OLE-server, exposing its objects in a type library. An OLE-controller can then display the information in an object catalogue, supporting the developer in his integration work. The interface description of the accounting application takes up the form of an OLE-object model. Seen from a developer's perspective, an object catalogue is available within the programming environment, which allows access to the elements of the object model in their application.⁸

For a core application as described above, it is important to expose a semantically stable object model. This means in particular that the naming of the classes as well as their methods and properties has to be consistent and easily understandable. It also needs to capture the functional essence of the core, which remains the same even if the implementation of the application might change, such as the move to a different data base system. For the core application in the case described above, this leads to defining a reference model for the accounting function (figure 5).

A reference model can then serve as a standard should market participants accept it. Strong market presence of the core application supplier helps to exert de facto standardisation power.

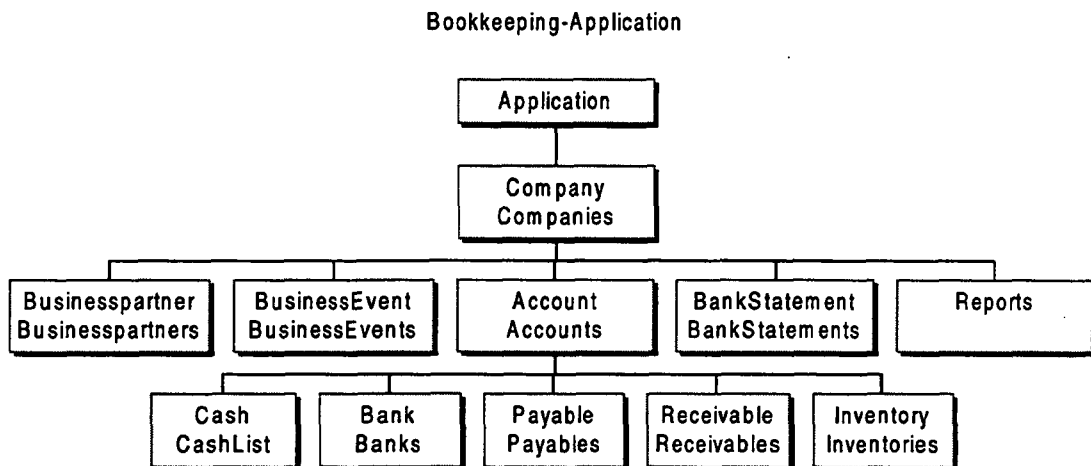


Figure 5: A reference model for the accounting function in a small business represented as OLE object hierarchy

Testing

The suitability of the approach has undergone a few testing phases.

- Internal - development of a prototype
 1. Prototype peripheral application integrated with an entry-level accounting product

⁸ For an in-depth discussion of OLE (Brockschmidt 1995).

2. Transferral of the prototype to demonstrate integration with larger products in the accounting product line
 3. Consolidation of the experiences into one accounting object model
- External - testing with selected partners
 4. Independent software houses
 5. Students

The first step was to show the technical feasibility of the approach. This was accomplished by integrating a bookkeeping application at that time under development in DATEV, with some interfaces a third-party product might include. At first customer data of a prototype retailing application was exchanged with the bookkeeping application. Secondly, the prototype sent invoices to the bookkeeping product. These two examples address the same issue as the business object documents of the OAGIS "SyncCustomer" and "LoadReceivable" (OAG 1997).

Both examples used the OLE-automation interface of the bookkeeping application. This step led to some suggestions on how to change the design of the bookkeeping product in order to supply a better object model in order to facilitate programming work of external software producers.

The graphical representation of the object model (similar to the one in figure 5) was used in negotiations with selected software houses supplying peripheral applications for certain industries. This phase is ongoing, and will ideally result in a few prototypes that expose the strengths and weaknesses of the approach, as well as a preliminary description of the co-operation process.

This phase will be complemented by experiments in the university, where students take up the role of independent software suppliers integrating their "solutions" with the DATEV core application.

These tests will help to establish the semantic stability of the wrapper, i.e. what should be exposed of an accounting application.

INTEGRATION WITHIN OTHER BUSINESS NETWORKS

If one looks at the business network constituted by small businesses, small software houses, tax accountants, and DATEV, the question arises whether lessons learnt can be extended to other business configurations involving small businesses. The following paragraphs outline scenarios of how one could transfer the concept of a standardised, well-defined software core supplemented by peripheral applications.

Setting up case studies in order to research these scenarios will deepen the understanding what business functions could be supported by a standardised core packaged by a wrapper. Cases to study this issue could be found in the following three configurations. Two configurations are presented describing a market situation, since they involve fairly independent business units. Then the concept is carried over to the hierarchy of a larger business, emulating a small business environment.

Configuration within Markets

Two different industries are selected in order to demonstrate how the core application concept could be extended.

Automobile Industry: Car Retailing

There are different business models of how car manufacturers implement their retailing function. One configuration used in practice consists of four parties - the car manufacturer, a retailing unit (owning a certain percentage of each dealership), the car dealers, and software companies, offering solutions (figure 6).

The manufacturer is interested in receiving timely, accurate, and comparable data from their retail outlets, as well as in ensuring a similar quality of customer service. This is the reason, why this party tries to influence IS used in dealerships.

The retailing unit is interested in keeping the cost of IS low. Also they want a solution that fits the different characteristics of each car retailer.

And retailers want to decide themselves what IS to use as well as when to introduce and change it.

Software suppliers for these types of solutions tend to be larger and often provide systems to the manufacturer's IT department. The competitiveness of offered solutions suffers from the fact that the circle of bidding software vendors is rather small.

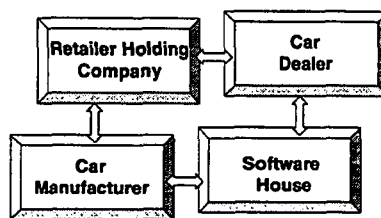


Figure 6: Configuration within the car industry

The involved parties could agree on a functionally lean core system that is complemented by independent vendors:

The core in this case could contain - next to a financial accounting solution - cost accounting and inventory management functionality and a customer service component. The inventory part would allow the car dealer to download part information from the car manufacturer. The support for reordering would help to keep inventory to a minimum in the whole supply chain. A common cost accounting component permits bench marking of car dealers. The customer service component ensures comparable treatment of clients regardless where they bought their car.

Car dealers might opt to individualised peripheral components for invoicing, the administration of their repair activities and their payroll system.

Insurance Industry: Insurance Agencies

Application architectures for the insurance industry have been under development for quite some time. IAA (insurance application architecture) by IBM or insurance reference models by KPMG can be quoted. These architectures tend to concentrate on the insurance enterprise and deliver blue prints for the corporate IS department how to proceed in constructing their specific IS architecture.

Small agencies represent the point-of-sale of many insurers. Their information systems need to be strengthened in order to exploit the advantages of an agency-based business model in the insurance industry, which is facing more competition e.g. from banks selling insurance products or direct insurers renouncing on an expensive agent network.

The parties involved in this configuration (figure 7) are the insurance company, the IS department of the insurance, the agencies and independent software suppliers. Again a core system provided by central IS used in all insurance agencies could ensure consistency and maintainability. The agencies achieve individualisation by enriching the raw solution by hiring smaller software houses.

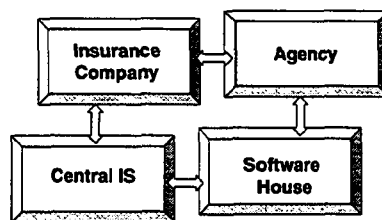


Figure 7: Configuration within the insurance industry

In this case the core would contain an accounting solution, a contract management component, damage administration, as well as a terms and conditions application. These are parts of the application portfolio of agencies that have links to the insurance company and that need to be functionally stable as well as up-to-date in each agency. Individualised parts could be personnel administration, which only agencies with several producers would need or an extended customer database with detailed information about the client-agency relationship.

Configuration within a Hierarchy

Large businesses have been emulating the advantages attributed to operating in a "small business" mode by decentralising (Frese 1993). For these types of businesses a similar problem arises to what has been described in the cases above (figure 8). There are small units that need to be co-ordinated to some extent by central IS within the whole corporation. There is a need for some standardisation. Also, central IS cannot forecast the different requirements arising in the various units and its modus operandi is geared towards large systems. There is a need for flexibility and for practices emerging in the decentralised units.

In particular, within a business using a profit centre organisation the individual units will want to decide themselves what type of software will allow them to perform and achieve set goals. A core application approach might help to achieve the standardisation needed to achieve IS efficiency within the organisation as a whole and allow for the emergence of decentralised IS-strategies helping to reach the effectiveness pursued with decentralisation⁹.

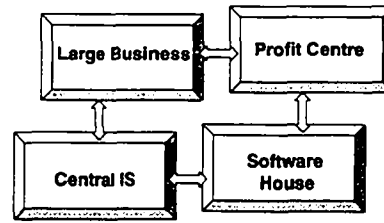


Figure 8: Configuration within a large business

The determination of what should make up the standardised core system and what parts of the application portfolio could be left to the individual business units is a reasoning process that would come up with differing set-ups depending on the purposes pursued in the configuration. Organisational goals will shift the lines between core system and its peripherals.

NEXT STEPS AND PRELIMINARY CONCLUSION

The lines between business functions and the border between what should be standardised and what should be left to the individual business unit are shifting as the business environment and strategies change. By extending the core application concept to other business networks, the process of defining what belongs to a core application and what constitutes peripherals can be more deeply discussed. Software technology needs to support these shifting domain definitions. And business administration needs to provide an understanding for the processes that lead to a sensible division between core and peripheral, which may happen in a dialogue involving interested parties or within a standardisation process. This way, application architectures can integrate the advantages of packages and custom-made software.

One example of how this can be achieved from a business perspective as well as from a technical point of view was discussed for a loose business network configuration. Switching standpoint to the segment of small business can yield fresh insights for information systems research and deliver frameworks for other business configurations.

REFERENCES

- Bassett, P. (1996) **Framing Software Reuse: Lessons From the Real World**, Upper Saddle River.
- Becker, J. and Schütte, R. (1996) **Handelsinformationssysteme** (Wholesale Trade and Retailing Information Systems), Landsberg/Lech.
- Birch, D.L. (1987) **Job Creation in America: How Our Smallest Companies Put the Most People to Work**, New York.
- BMFT (1994) Bundesministerium für Forschung und Technologie (Federal Ministry for Research and Technology) **Initiative zur Förderung der Software-Technologie in Wirtschaft, Wissenschaft und Verwaltung** (Initiative for the Advancement of Software Technology in Commerce, Science and Administration), Bonn.
- Brockschmidt, K. (1995) **Inside OLE**, 2nd ed., Redmond.
- Bundesamt für Statistik (1994) (Federal Bureau of Statistics) **Unternehmen und Beschäftigte am 25.05.1987** (Businesses and Employees on 25.05.1987), **Statistisches Jahrbuch** 1994, Wiesbaden, 137.
- Coda (1997) Why Coda? Error! Reference source not found., 02.08.1997.
- El-Namaki, M.S.S. (1990) Small Business – The Myths and the Reality, **Long Range Planning**, (23) 4, 78-87.
- Frese, E. (1993) Geschäftssegmentierung als organisatorisches Konzept - zur Leitbildfunktion mittelständischer Strukturen für Großunternehmen (Business Segmentation as Organisational Concept – the Lead Role of Medium-Sized Structures for Large Businesses), **ZfbF (Zeitschrift für betriebswirtschaftliche Forschung)**, (45) 12, 999-1024.

⁹ The balance between deliberate vs. emerging strategies (Mintzberg 1989) needs to be addressed for the IS-function as for all other business strategies.

- Hofmann, D. and Rockart, J.F. (1994) Application Templates: Faster, Better, and Cheaper Systems, *Sloan Management Review*, (36) 3, 49-60.
- IBM (1995) Die IBM geht neue Wege zum Markt (IBM Takes New Roads to the Market), *IBM Nachrichten*, (45) 321, 70-75.
- Mertens, P. (1997) **Integrierte Informationsverarbeitung I** (Integrated Information Processing), 11th ed., Wiesbaden.
- Meyer, J.-A. (1993) Semiindividuelle Software (Semi-Individual Software), *DBW (Die Betriebswirtschaft)*, (53) 1, 129-130.
- Mintzberg, H. (1989) **Mintzberg on Management**, New York.
- OAG (1995) Open Applications Group, **White Paper**, Chicago.
- OAG (1997) Open Applications Group, **Open Applications Group Integration Specification**, Release 5, Document Number 970630OAG, Chicago.
- SAP (1997) **The BAPI Catalog**, **Error! Reference source not found.**, 02.08.1997.
- Scheer, A.-W. (1994) **Wirtschaftsinformatik – Referenzmodelle für industrielle Geschäftsprozesse** (Information Systems Research – Reference Models for Industrial Business Processes), 4th ed., Berlin.

ACKNOWLEDGEMENTS

This research has been funded by DATEV eG, Nürnberg, Germany (<http://www.datev.de>).