ERP in the e-commerce era

Paul Oude Luttighuis and Frank Biemans
Telematics Institute
P.O. Box 589, NL-7500 AE Enschede, The Netherlands
{luttighu,biemans}@telin.nl

Abstract

E-commerce developments and requirements will force ERP vendors to continuously innovate their products. Classical ERP system characteristics are inappropriate for a future in which e-commerce is common practice. This paper reviews five of these characteristics, explains why they hamper e-commerce, and discusses solutions. Some of these are indeed already recognised and dealt with by ERP vendors; others aren't. Finally, the paper introduces the research efforts of the Telematics Institute that should help effective and rapid application of ERP in e-commerce contexts.

Introduction

ERP systems are at the electronic heart of day-to-day business operations and planning inside companies. As such, they have been in use for many years. Currently, we witness a new trend that significantly impacts the ERP world: e-commerce. Due to e-commerce developments and requirements, ERP systems face a number of challenges in order to remain suitable candidates for use in an e-commerce environment. In order to sharpen discussion, we deliberately take an overly critical perspective on classical ERP systems.

In our definition, classical ERP systems are solely aimed at planning and managing a single company's internal processes. In the following section, we will elaborate on this by presenting five classical ERP characteristics. In subsequent sections, we will show that these characteristics make classical ERP in appropriate for e-commerce and networked business and discuss how these issues may be dealt with. Indeed, some of these have already been recognised and addressed by the ERP world. Some other points however, have hardly been made so far. Finally, we will translate our findings into two major research lines, currently pursued by the Telematics Institute.

Five classical ERP characteristics

First, the basic function of classical ERP systems involves company-internal planning and management of business functions. In other words, a company's customers and other partners have no direct interaction with the system. Classical ERP systems do not encompass front-office functions for interaction with the business environment. Also, they lack, for instance, functions for negotiating with clients and suppliers. In short, classical ERP is *incomplete* (Figure 1).

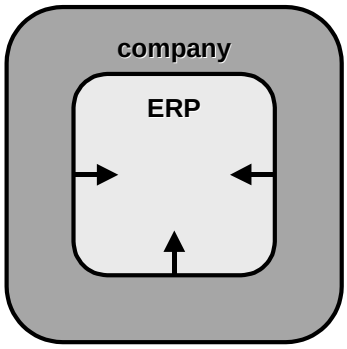

Figure 1 — Classical ERP is incomplete.

Second, modules in classical ERP are strongly interdependent, making it difficult to reconfigure or extend them, or to couple them to other systems, in case of changing requirements or contexts. In short, classical ERP is *monolithic* (Figure 2).

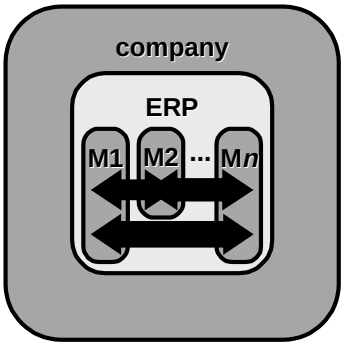

Figure 2 — Classical ERP is monolithic.

Third, classical ERP systems are not coupled. They support a single company's planning and management, without interacting with ERP systems of customers, suppliers, and other partners. In practice, of course, this remark should be relaxed: many ERP systems today incorporate for instance supply-chain management functions, and most of them enable EDI messaging. For sake of discussion however, we state that classical ERP is *non-coupled* (Figure 3).

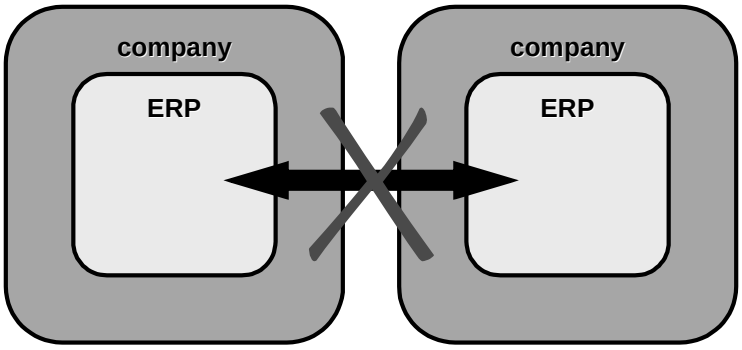

Figure 3 — Classical ERP is non-coupled.

Fourth, classical ERP systems are focussed on the planning of individual companies, rather than on cross-company planning or sharing ERP functions. Of course, in some cases, cross-company functions may reside at a single company, typically a powerful company in a chain or network. Still, whether distributed or centralised, ERP functions are currently only offered at a single company's basis. In short, classical ERP is *single-company* (Figure 4).

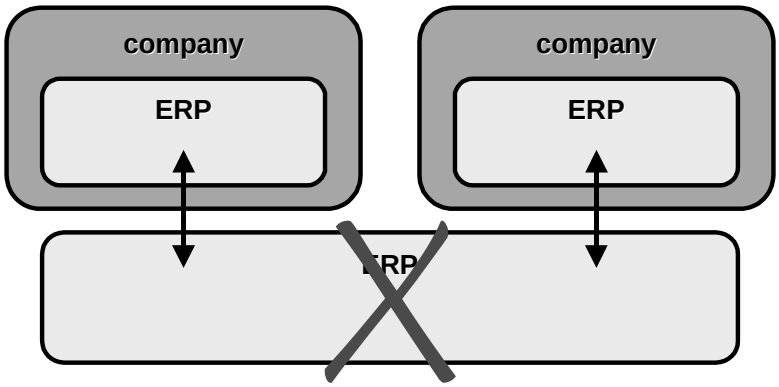

Figure 4 — Classical ERP is single-company.

Fifth and finally, classical ERP systems impose rigid structures onto data and processes. A considerable degree of configurability is offered, but in many cases the business process has to be tuned to the system to some extent. Once in use, changes in environment or requirements, and does not support flexibility, nor freer forms of interaction. In short, classical ERP is *restrictive* (Figure 5).

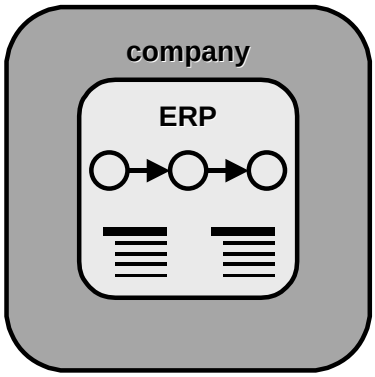

Figure 5 — Classical ERP is restrictive.

Below, we will address these characteristics in more detail, explain why they make classical ERP insufficient for e-commerce, and discuss solution directions. Finally, we will show how these issues are translated to the Telematics Institute's major research lines in the area of e-commerce and networked business.

From incomplete to extended ERP

E-commerce encompasses electronic business interactions with business partners. Hence, the gap between classical ERP functions and a company's borders should be bridged. Typically, this involves front-office functions, such as electronic shops, electronic catalogues, client-initiated order tracking, electronic payment, and customer-relationship management. From an ERP perspective, the addition of this type of functions is often defined as extended ERP, or XRP. However, at the back-end of a company, additional e-commerce functions may be added to ERP as well, such as electronic procurement and supply-chain management functions (Figure 6).

ERP vendors are starting to provide sell-side e-commerce modules, but many others (such as Microsoft, IBM, Netscape, Broadvision) already provide electronic commerce functions as well [SJO99].

To fully benefit from the advantages of XRP functions, they should be adequately coupled to the original internal ERP system. For instance, orders received electronically should be fed electronically into

the ERP system. And, client-initiated order tracking should not be carried out with human intervention. As Forrester recently reported however [Gre99], coupling between e-commerce applications and ERP systems is often still carried out manually, that is by re-entering data by hand. And if coupled electronically, this coupling is mostly implemented by means of hand-written patches. In case of e-commerce modules and ERP modules from the same vendor, there will of course be a better fit.

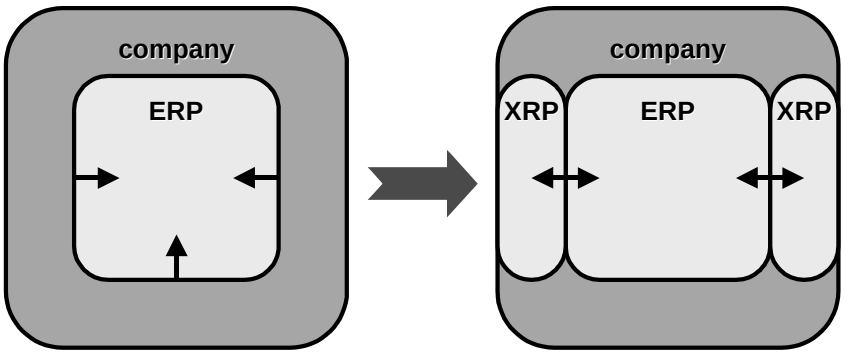

Figure 6 — From incomplete to extended ERP.

The inclusion of negotiation functions in (extended) ERP will allow ERP systems to do business with electronic commercial brokers or negotiating agents.

From monolithic to component-based ERP

With customers and other company's continuously changing a company's business environment at a high pace, there is a growing need for flexibility in IT systems. With classical ERP's monolithic nature, this flexibility was not provided. Therefore, ERP vendors are today re-engineering their ERP product into a series of subsystems and components. Baan is one of the early movers in this area, with its Baan Series portfolio of products. These components can then be supplied on a component-by-component basis and assembled and configured to fit the existing needs of specific users.

In order for the components to be interoperable, they should adhere to a component-based architecture. The components run on a component framework, such as DCOM, CORBA, or Enterprise Java Beans (Figure 7).

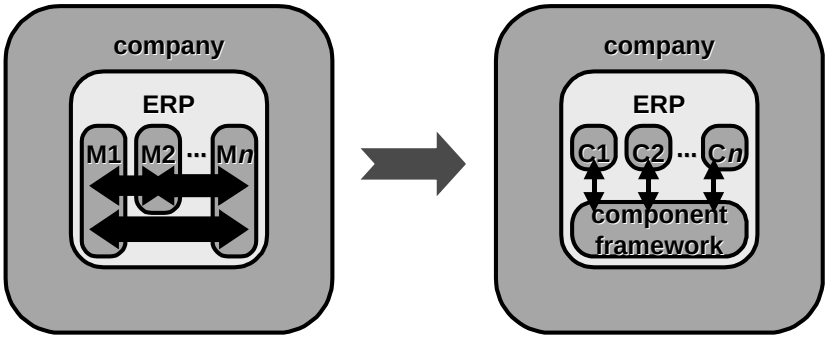

Figure 7 — From monolithic to component-based ERP.

From non-coupled to coupled ERP

Extending ERP with e-commerce functions is but one step. In order to conduct e-commerce, ERP or XRP functions should be coupled across company borders (Figure 8). Traditionally, this coupling has been implemented through EDI messaging technology. The history of EDI however has been one of

costly and time-consuming standardisation and implementation. Also, EDI just provides a data coupling, not a process coupling.

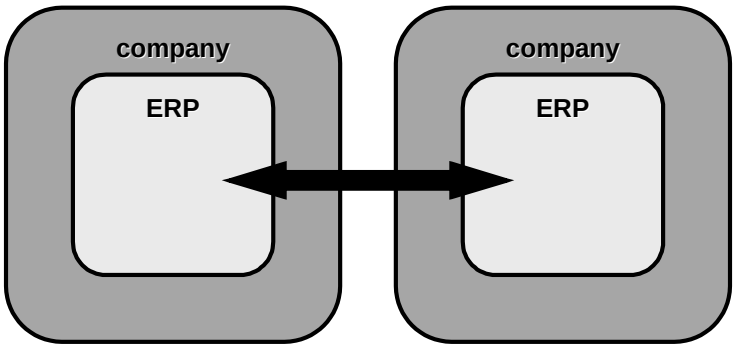

Figure 8 — From non-coupled to coupled ERP.

One solution to this issue would be to use the component-based paradigm across company borders as well. This however has some major obstacles. First, it would require all companies involved to use the same or at least interoperable component frameworks, as well as a shared component-based architecture. It is highly questionable whether this would be a realistic aim in the heterogeneous world of business and IT. Only in tabula rasa situations, or in highly controlled contexts, such an approach may apply across companies as well.

Therefore, there is a considerable market for middleware technology that imposes fewer constraints on the applications coupled, such as message brokers [Bijl98]. Message brokers provide cross-applications functionality such as messaging, data transformation, message logging and warehousing, and workflow. Using message brokers to couple applications requires no change to the applications themselves, because the interface between the message broker and the application is extremely flexible.

Another advantage of the message broker concept is that it also enables applications to be coupled on their process aspects, rather than just a data coupling, such as in the case of classical EDI. With respect to data coupling, message brokers handle the variety of data format by means of a metadictionary. Currently evolving XML technology will offer the same type of features.

Of course, message brokers provide application coupling on an ad hoc basis, rather than by means of a well-designed component-based application architecture. Still, since they are flexible all-purpose application integrators, they enable application coupling on an evolutionary, rather than a revolutionary basis. This is particularly suitable for environments with many legacy applications, in which a one-step revolution towards a component-based architecture would be too costly or too risky. In some cases, a restricted integration may have to be accepted. In fact, the ad hoc approach can be conveniently combined with the architectural approach by first using the ad hoc approach and then migrating individual applications to a component-based architecture on a one-by-one basis.

From single-company to multi-company ERP

Classical ERP systems apply to the business processes of individual companies. In a networked economy, not only do companies conduct commerce on an electronic basis, they also co-operate in design, planning and other activities in order to effectively or efficiently conduct business. For instance, there is a growing need for tracking and tracing of products through an entire chain or network of companies. Also, logistics planning, forecasting, and replenishment can be executed in co-operation, rather than by individual companies [CPFR98]. And, there are many examples of data collections which are of inter-

est to an entire business networks or chains rather than to individual companies, such as sales or product information. Hence, there is a need for chain-wide, network-wide, or community-wide ERP (Figure 9).

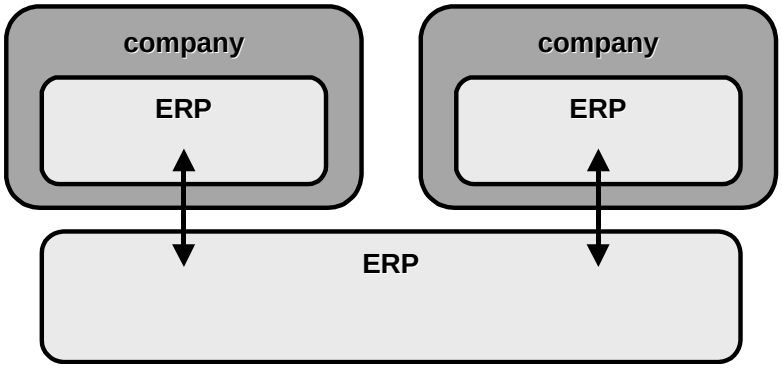

Figure 9 — From single-company to multi-company ERP.

Multi-company ERP services constitute a chain's or network's ICT service infrastructure, and lift this infrastructure from the network to the business level.

In order to be of real use, multi-company ERP should of course be coupled to the ERP systems of the companies involved (Figure 9). This requires flexible interfaces to many different ERP systems. As to the flexibility of the multi-company ERP system itself, this should be substantial, since it should support the changes of each of the companies involved. Hence, they should enable far-reaching — preferably on-the-fly — configuration features.

Finally, the deployment of multi-company ERP should be accompanied by an appropriate set of tools and techniques, including reference models for networked business.

From restrictive to flexible ERP

There is an ongoing discussion about the rigidity of ERP systems, that is, the extent to which they superimpose data and process structures on a company. From an ERP point of view, the similarities between business processes and their fit to reference models is stressed. Others stress the differences between companies and advocate a custom-made approach. In any case, ERP systems will have to be configurable to some extent at least, both with respect to the data structure as well as with respect to the process aspect, in order to adhere to company-specific processes or requirements.

Configurability
Configurability of ERP is one way of offering flexibility. In e-commerce context, this will be even more important, because the flexibility of cross-company functionality will have to answer to requirement changes of multiple companies, especially when the relations between these companies are volatile.

In the past, electronic interaction among companies (including EDI) has been often restricted to companies with long-term and proven relationships, partly due to the rigidity of the data formats used. The economic trend however is towards shorter-term relations, sometimes even for the duration of a single order. ERP systems should be able to follow this volatility by means of flexible, preferably on-the-fly reconfiguration.

Multimedia
In e-commerce, there is also a growing need for multimedia data, rather than just text- or record-oriented data. This typically holds for graphical product information directed at end-customers.

CSCW
A so far unexplored area however is the integration of structured-data structured-process applications such as ERP and many e-commerce applications with ICT applications that apply to looser forms of communication or co-operation, such as CSCW (Computer-Supported Co-operative Work). CSCW applications typically support communication and collaboration where these cannot be structured beforehand, with respect to neither data nor process. In e-commerce context, this type of communication can be of use in for instance concurrent design, where designers of multiple companies both need flexible support for their creative group process, as well as for exchanging structured information on the products designed. As another example, consider a company suddenly looking with a huge order, which can only be accepted when current production planning is carefully reconsidered and longer delivery times are negotiated with the customer or even with other customers. This requires a complex interaction between the customer, sales, and production planning, which cannot be structured beforehand. Still, all parties in this interaction should have ERP data at their disposal (Figure 10).

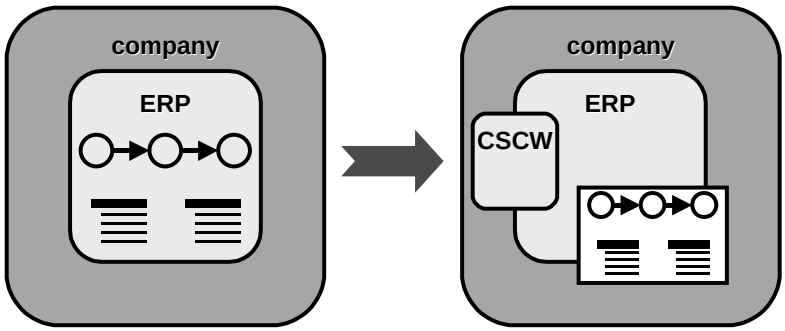

Figure 10 — From restrictive to flexible ERP.

ERP as a product or as a service

In the above discussion, we did not make assumptions about whether ERP functions should be installed and used as a software product or provided as a service. Even for classical ERP, ERP hosting is a solution for companies seeking to cut down on maintenance costs. When ERP is offered on a component basis, ERP components may even be outsourced one by one, or ERP modules from different vendors may be used.

Third-party ERP is however particularly important in the case of extended ERP functions, coupling of ERP systems across companies, and multi-company ERP. The wider the group of users or the more volatile the relations among the users, the more reason there will be to have it provided by a third party, which may be a neutral organisation or a commercial service provider. A third party will also be better equipped to handle the high degree of flexibility required. This party typically provides these services on top of lower-level services such as message brokering or VAN services.

Of course, it is also possible to have one (typically large or powerful) member of the chain or network provide these services.

In short

In Table 1, we summarise the discussion by relating the characteristics of classical ERP, the problems these yield in e-commerce context, and the solutions we discussed.

CHAR.	E-COMMERCE PROBLEM	SOLUTION OPTION
incomplete	– no e-commerce functions included – no coupling of front- and back-office	– extended ERP – coupling of extended ERP with classical ERP
monolithic	– inflexibility	– component-based architecture and framework
non-coupled	– no electronic interaction – only data coupling, no process coupling	– component-based architecture and framework and/or – message brokering
single-company	– no network-wide services or products	– chain-, network- or community-wide ERP
restrictive	– inflexibility – rigidity	– (on-the-fly) configuration, – multimedia data, and/or – CSCW

Table 1 — Summary.

Research issues at the Telematics Institute

In selecting its research topics in the area of e-commerce and networked business, the Telematics Institute has taken the above discussions as a starting point, albeit not from a pure ERP perspective. On this basis, it has recognised two major areas of research, which are currently pursued. The research is accompanied by different kinds of pilot projects in different areas of business, such as finance, trade, and logistics. The two major research lines are:

- developing a Rapid Service Development (RSD) environment for transaction services, and
- developing generic services for networked companies.

Rapid Service Development of transaction services

The RSD environment is directed at enabling quick and business-driven — rather than technology-driven — design and implementation of transaction services between companies [BJO+99]. Because RSD is business-driven, the business models used and the business transactions needed are also matter of explicit design. They precede the design and implementation of the transaction service itself. The RSD environment consists of a coherent set of tools and methodologies. The toolkit contains both modelling and analysis features needed for appropriate design decisions.

RSD encompasses four levels of design (see Figure 11): business models, transaction scenario's, transaction services, and transaction systems.

Business models are high-level descriptions of the actors and roles involved in (electronic) transactions, the assignment of roles to actors and the main physical, information and financial flows among these actors and roles. As such, their role can be described as reference models for cross-company interaction. However, they are not necessarily standard, but rather a matter of explicit design.

Transaction scenarios are descriptions of the actual step-by-step proceedings of transactions, taking into account the procedural as well as the data aspect. Transaction scenarios implement business models. Neither business models nor transaction scenarios necessarily describe electronic transactions only. They are descriptions at the business-level. By means of electronic transaction services, it is decided which (part of) the transactions is carried out by electronic transaction systems and components. Trans-

action services are identified as those services that should be supplied by electronic transaction systems. They hence serve as the system specifications for these.

RSD is component-based with respect to business models as well as transaction scenario's and transaction systems. That is, during the design of each of these, there is access to a repository of components, which can be selected, configured, and assembled in the design. Hence, this enables the use of standard business models (reference models) and standard transaction scenario's (such as the SET scenario) as well (Figure 11).

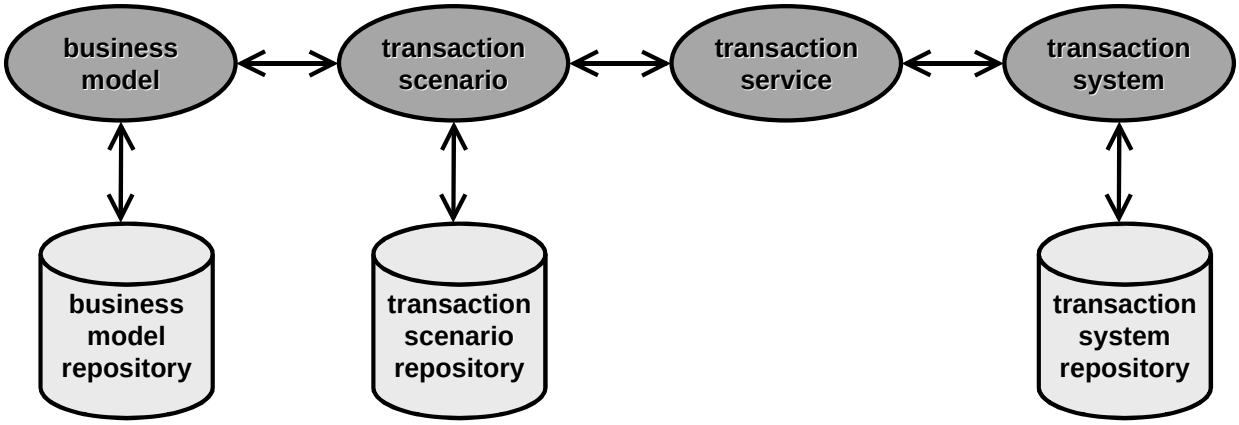

Figure 11 — Rapid Service Development.

Although Figure 11 may suggest a top-down waterfall approach, RSD is iterative. This not only allows for bottom-up elements in the design, but also enables reverse engineering, required for incorporating legacy systems in the design. Legacy is an inevitable topic when dealing with ERP systems in an e-commerce context.

The transaction services and systems delivered by RSD cover any functionality needed for coupling existing company-internal systems, with respect to data and process aspects. Hence, this may include extended ERP functionality. Furthermore, RSD will be largely applicable to in-company application integration and coupling as well, although it is directed at cross-company transaction service at first. In conclusion, Table 2 highlights those issues in Table 1 that are addressed by RSD.

CHAR.	E-COMMERCE PROBLEM	SOLUTION OPTION
incomplete	– no e-commerce functions included – no coupling of front- and back-office	– extended ERP – coupling of extended ERP with classical ERP
monolithic	– inflexibility	– component-based architecture and framework
non-coupled	– no electronic interaction – only data coupling, no process coupling	– component-based architecture and framework and/or – message brokering
single-company	– no network-wide services or products	– chain-, network- or community-wide ERP
restrictive	– inflexibility – rigidity	– (on-the-fly) configuration, – multimedia data, and/or – CSCW

Table 2 — Issues addressed by RSD.

Generic services for networked companies

As argued, many of the current solutions for electronic transactions are entirely tuned to individual companies. The second research line is involved in designing chain- or network-wide ICT services.

These services should be coupled to individual ERP systems of the company's involved. Also, the service should be considerably flexible so as to be able to quickly incorporate new participants or changes in the network. Preferably, these services should incorporate a level of flexibility that allows for on-the-fly reconfiguration.

Consider for instance tracking and tracing. In some form, tracking and tracing functionality plays a central role in many ICT applications in physical product chains. However, there are many differences between the precise ways in which tracking and tracing is performed. There are no chain- or network-wide tracking-and-tracing services; these are restricted to those parts of chains or networks that are under the control of a single company. Different companies use different product identifications or different collection units (pallets vs. containers, for instance). One company requires complete insight in all tracking data; the other just wants to be informed in case actual proceedings deviate from planning. This, along with many other potential differences, makes it hard to develop a truly chain-wide tracking and tracing service, whereas the need for such a service is more and more recognised.

Other candidates for generic chain-or network-wide services are multi-company management information services, collaborative planning, community-wide commercial services, sharing (multimedia) product or sales data, et cetera. Table 3 highlights those issues of Table 1 that are addressed by this line of research.

CHAR.	E-COMMERCE PROBLEM	SOLUTION OPTION
incomplete	– no e-commerce functions included – no coupling of front- and back-office	– extended ERP – coupling of extended ERP with classical ERP
monolithic	– inflexibility	– component-based architecture and framework
non-coupled	– no electronic interaction – only data coupling, no process coupling	– component-based architecture and framework and/or – message brokering
single-company	– no network-wide services or products	– chain-, network- or community-wide ERP
restrictive	– inflexibility – rigidity	– (on-the-fly) configuration, – multimedia data, and/or – CSCW

Table 3 — Issues addressed by generic services for networked companies.

References

[Bijl98] Dik Bijl, *Networked (Inter)Enterprise Computing*. Series in Open Computing **7**, 1997 (p. 38), **3**, 1998 (pp. 40–44), **4**, 1998 (pp. 44–48), **5**, 1998 (pp. 46–51), **6**, 1998 (pp. 44–47), **7**, 1998 (pp. 8–12,17). In Dutch.

[BJO+99] Frank Biemans, Wil Janssen, Paul Oude Luttighuis, Hans Schaffers, and Petra van der Stappen, 'Business-Driven Design of Transaction Services', In: *Proceedings of the 5th International Conference on Concurrent Enterprising (The Hague, The Netherlands, 15–17 March 1999)*, pp. 337–344

[CPFR98] Voluntary Interindustry Commerce Standards, *Collaborative Planning, Forecasting, and replenishment* — Voluntary Guidelines. 1998.

[Gre99] Emily Green, *Lipstick on a Pig*. Internet in Business **2**, 1999.

[SJO99] Petra van der Stappen, Wil Janssen, and Paul Oude Luttighuis, *NETS State of the Art Networks and Services*. First release, Telematics Institute, Enschede, The Netherlands, March 1999.