7

Systematic Design of an E-marketplace for Interior Products

Hugo ter Doest, Henk Jonkers, Jaap van der Spek, Diederik van Leeuwen, Jaap Reitsma, Joseph Owusu[1]
Telematica Instituut, P.O. Box 589, 7500 AN Enschede, the Netherlands
Telephone: +31 53 4850485, Fax: +31 53 4850400
E-mail: Hugo.terDoest@telin.nl
[1]*Mieloo & Alexander, P.O. Box 85590, 2508 CG The Hague, the Netherlands*

Abstract: The Rapid Service Development (RSD) metholology defines an integrated, model-based approach for quickly and efficiently developing e-business services, with a focus on business-to-business transactions. This paper shows how RSD is applied in a pilot project to develop a web-based application for selling interior products. In the project Tapestria, an e-marketplace that sells interior fabrics to the U.S. market, is extended to support the sale of rugs. We show how RSD is applied to identify and develop business roles, processes, functions and transactions. Also, the next step of the software design for the (web-based) front office and back office is covered. The paper is concluded with an evaluation of the development process.

Keywords: e-marketplace, design method, enterprise modelling

1. INTRODUCTION

Technically, developing an electronic marketplace is an easy thing to do: deploy a database server and an application server, develop the required business logic, and the marketplace is up and running. The difficulty however is in making the marketplace work, i.e. setting up the business processes needed for logistic and financial operations such that they are in line with the transactions offered by the marketplace. To achieve this, it is important to apply a systematic development approach that results in carefully designed business processes on the basis of which business needs can be identified and technological solutions can be selected or designed.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35617-4_48

J. L. Monteiro et al. (eds.), *Towards the Knowledge Society*

Hunter Douglas, market leader in window covering products, is the owner of Tapestria (http://www.tapestria.com), a marketplace that enables the trade of interior fabrics between European fabrics mills and interior designers in the U.S. Although Hunter Douglas has good insight in the interior products value chain, at the time of the launch of Tapestria it was a newcomer in the market segment of interior fabrics. Hunter Douglas selected this market on the basis of opportunities present for an electronic marketplace. The main opportunity is to offer fabrics from a wide range of mills to a large network of interior designers. Suppliers benefit from a large number of customers while interior designers can order from a catalogue of products otherwise unavailable to them.

WYSIWYG is an initiative of Hunter Douglas funded by a BTS grant of the Dutch government in which multiple parties cooperate in order to create methods and processes for the mass customisation of interior products. Ultimately, the goal of the project was to develop a web site for the on-line design and digital printing of interior fabrics. This paper reports on the first phase of the project which explores on-line design of configurable products by the introduction of rugs. Telematica Instituut is responsible for the business design and part of the software design.

The project was an opportunity to validate the Rapid Service Development (RSD) methodology (Janssen, Steen, 2000). The RSD metholology, developed at the Telematica Instituut, defines an integrated approach for the development of electronic business services, with a particular focus on business-to-business transactions. The RSD approach covers all aspects of service development, from defining requirements, specifying (modelling) a business model, analysing the business processes to implementing a service possibly using (existing) software components. In Tapestria, the focus was on the validation and further improvement of a development approach for electronic marketplaces.

The rest of this paper is organised as follows. In Section 2 we give a more detailed introduction to the Tapestria business model and we give an overview of the business requirements involved in the introduction of rugs. Section 3 describes the RSD methodology and the approach of the Tapestria pilot. In Section 4 and Section 5 we elaborate on the business design and the technical design, respectively. Finally, in Section 6 we evaluate the RSD methodology and draw some conclusions.

2. TAPESTRIA

Tapestria is an electronic marketplace that enables the trade of interior fabrics between suppliers primarily located in Western Europe and interior designers in the U.S. Traditionally, the fabric industry has suffered from highly fragmented and inefficient supply structure. Tapestria connects them to a pervasive network of interior designers and retailers. Also, Tapestria offers high quality fabrics that are not unavailable through other channels in the U.S. Following Kaplan and Sawhney's classification of e-marketplaces (Kaplan, Sawhney, 2000), Tapestria most closely resembles a 'catalogue hub'.

Tapestria shares characteristics of both an e-marketplace and an e-wholesaler. Suppliers are responsible for pricing and stocking their goods whereas quality and catalogue are managed by Tapestria. See *Table 1* for an overview of the differences and similarities. For rugs, the picture is a bit more complex. Some products will be priced according to the marketplace model (suppliers set the price), other products according to the wholesale model (Tapestria sets the price).

CHARACTERISTIC	E-MARKETPLACE	E-WHOLESALE
pricing policy	***by supplier***	by wholesaler
ownership stock	***supplier***	wholesaler
customer relations	supplier	***wholesaler***
web site	supplier-specific	***wholesaler site***
catalogue	one catalogue per supplier	***one catalogue***
order process	no combined orders	***combined orders***
payments	to suppliers	***to wholesaler***
returns and claims	no responsibility	***responsibility***
earnings	fees	***margins***

Table 1. Managed marketplace

2.1 Problem definition

In the WYSIWYG pilot the new product category of rugs has to be added to the Tapestria site for fabrics. The complete supply chain from on-line ordering to fulfilment and claim handling is developed and integrated with the processes already in place for fabrics.

Tapestria offers the possibility to configure rugs on-line. Parameters that can be changed in designing are the size of the rug, colours used in the design, and, in the case of custom border rugs, the field and border materials to be used. In addition, there is a possibility to order sample materials based on the design made.

From an order fulfilment perspective, a distinction is made between in-stock items and made-to-order items. Tapestria has contracted several rug suppliers to keep certain items in stock. If an in-stock item is ordered, the member pays the full product price, and Tapestria places a purchase order at the supplier who can deliver the product within weeks. The supplier updates a stock level indicator on the Tapestria web site on a daily basis. If a made-to-order rug is ordered, Tapestria also places a purchase order at the supplier. The difference is that the member pays 50% upfront and the remaining 50% upon delivery of the rug. Another difference is that delivery times are longer (approx. three months).

As is the case for interior fabrics, the sampling service is crucial for members to get a feel for the product. In general, a sample consists of a high quality colour print, a quality sample (representative piece of a rug), and a set of colour pompons.

3. DEVELOPMENT APPROACH

The development of electronic business services can be highly complex; it generally involves multiple organisations, and a wide range of aspects, from high-level strategic business concerns to low-level protocol definitions, need to be addressed. In order to deal with this complexity, the Rapid Service Development (RSD) methodology (Janssen, Steen, 2000) has been developed at the Telematica Instituut. It has been inspired on development methods such as Rapid Application Development (RAD) (Martin, 1991), the Unified Modelling Methodology (UMM) (Clark, 2000) and the Rational Unified Process (RUP) (Kruchten, 2000). In the remainder of this section, we describe the RSD methodology in general, as well as the specific approach that we followed in the Tapestria pilot.

3.1 The RSD methodology

The RSD methodology applies the 'separation of concerns' principle. The RSD framework (illustrated in Figure 1) distinguishes seven different aspect areas, called *cornerstones*:

1. *Ambition and scope* which describes the goal of the transaction services to be developed, both in terms of business goals and technology. Here we also specify starting points and constraints.
2. *Networked enterprise models*, which give a high-level view of the cooperation in terms of the actors involved, the roles they fulfil and their relationships, but also in terms of the business functions they perform and the flows between these.
3. *Transaction scenarios*, which describe the inter-organisational processes and transactions that achieve the co-operation.

4. Specifications of the *procedures*, which describe the interactions and message formats for conducting transactions between organisations in detail.
5. *System descriptions*, which describe the architecture and the composition of transaction systems.
6. *Component* specifications, which describe the functionality of well-encapsulated, reusable pieces of software.
7. *Protocol and code specifications*, which describe in detail the protocols use for communication between components, as well as the inner workings of these components where necessary.

The cornerstones provide a framework in which the different model types and design decisions used in a development trajectory can be placed. However, it does not specify the order in which the different models are developed. A number of develoment methods, or 'pathways', can be used to traverse the framework. Many different pathways are possible, depending on the purpose of the trajectory (e.g. process standardisation, legacy integration, development of a completely new application). They do not always address all the cornerstones. A method for application development, however, always has to provide ways to cross the boundary between the business design (left-hand side cornerstones) and the technical design (right-hand side cornerstones).

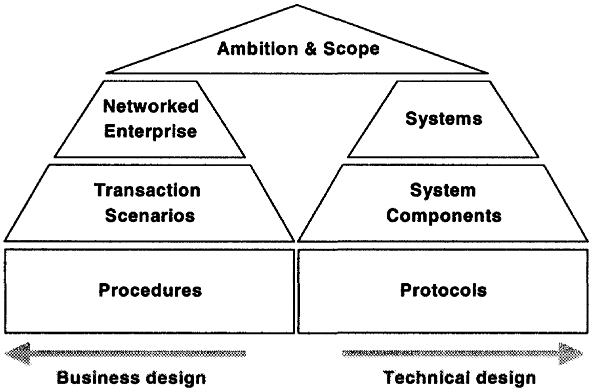

Figure 1. RSD framework

The RSD methodology provides modeling techniques for the business-oriented cornerstones. These techniques are derived from the Testbed business process modeling approach (Franken, Janssen, 1998), extended with specific concepts and model types needed to model "networked businesses" (with inter-organisational processes).

In addition to the framework, development methods and modelling techniques, the RSD methodology also offers a collection of reusable model components, patterns and reference models. Reference models have been completed for, among others, e-markets (Iacob, Smit, 2001) and e-procurement (Bruins, Steen, 2000).

3.2 Approach in the Tapestria pilot

Figure 2 illustrates the development approach that we followed in the Tapestria pilot. The types of models are shown with their dependencies and their position in the RSD framework. Note that both in the business design (cornerstones on the left) and the technical design (cornerstones on the right) the approach was mainly top-down. The dependencies in *Figure 2* show which models use information from other models as input. They do not specify a strict ordering of modeling activities: especially in the business design phase, several iterations over the three business cornerstones were necessary before the role definitions and business processes models converged to stable results. The technical design required fewer iterations because many of the technical design decisions had already been made during the development of the original Tapestria fabrics site. Also, the detailed technical design and implementation has been assigned to a Hunter Douglas software development team, outside the scope of the pilot project.

All activities were performed in intensive interaction with business managers of Hunter Douglas, serving as a sounding board for their ideas which were captured in role diagrams and process diagrams. In a period of four months several iterations over these diagrams resulted in a stable blueprint of the rugs module. Business design was completed with a man-machine assignment that specifies for each process activity if and how it will be supported by software. This is the link to the technical design (steps 8, 9, 10) that starts with input about the current Tapestria systems (step 8) and technical architecture (step 9). In step 10 technical requirements are specified and a data model for rugs is designed.

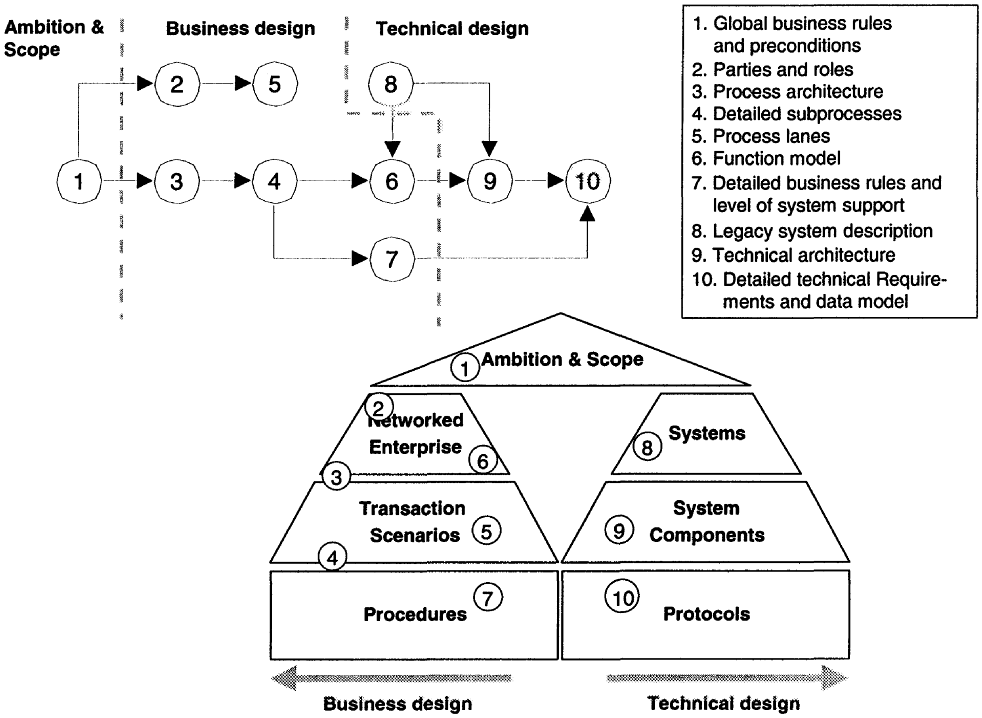

Figure 2: Overview of models, their dependencies and placement in the RSD framework

The 'ambition & scope' cornerstone (step 1) provides the starting point and constraints for both the business design and the technical design. Here, the business model to be adopted (managed marketplace) is determined, preconditions and constraints for the processes are identified, and high-level business rules are described (e.g., ownership of materials, price setting policy, product categories to be offered, etc.).

In the next two sections we describe the business design phase and the technical design phase of the project in more detail.

4. BUSINESS DESIGN

The business design covers the cornerstones on the left-hand side of the RSD framework. Three types of models are used in this phase, *role* models, *business process* models and *function* models. In the following subsections we consider each of these model types in some more detail. We do not show all the models that have been constructed in the project, but only some illustrative examples.

4.1 Roles

The first step in the business design is to identify the parties involved in the processes, and the roles that they fulfil. For the main process categories, the relation between the roles, in terms of the main (physical, information and money) flows between them, is described. As an example, *Figure 3* shows the roles and flows involved in the ordering and order fulfilment of in-stock products. The 'member' is an interior designer who may place orders with Tapestria, while the 'delivery address' may be the home of the member's customer. 'Operations / F&A' is the party responsible for the actual operation of the Tapestria marketplace, taking care of order intake and co-ordinating the order fulfilment. The products are shipped from a European supplier, through a hub in the Netherlands, to the USA. Optionally, a service provider in the USA takes care of additional services such as rolling out the rug.

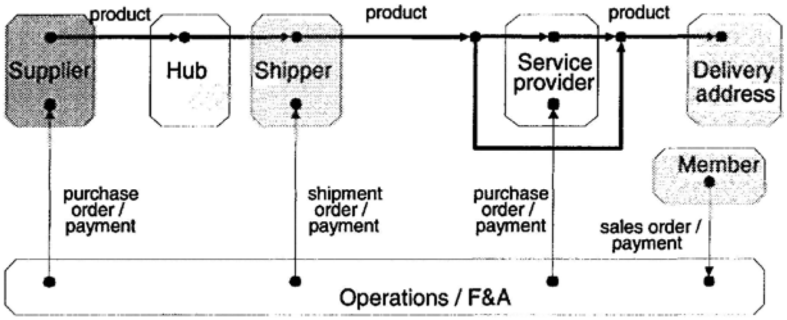

Figure 3. Role diagram (step 2)

RSD role diagrams are similar to e3-value models proposed by Gordijn (Gordijn, 2002). However, e3-value models have the purpose of modelling value exchanges of objects of economic value between actors, whereas RSD role diagrams have a broader application. e3-value offers, besides actors and value exchanges between them, concepts like value offering, value interface, market segment (a collection of interfaces) to precisely specify how and under what conditions value exchanges take place. Furthermore, e3-value has the possibility to define scenario's using Use Case Maps (Buhr, Casselman, 1996).

4.2 Business processes

The inter-organisational business processes are the core of the business design. The processes are documented in a number of steps.

Process architecture (step 3): First, we show the global structure of processes in terms of subprocesses and their relationships, resulting in a "process architecture". At the highest level, we classify the processes in three categories (see *Figure 4*): processes related to the *product supply*, processes related to the *buying of products*, and *supporting processes*. For each of these categories we specify the subprocesses and their main causal dependencies. *Figure 4* shows the structure of the buying process as an example.

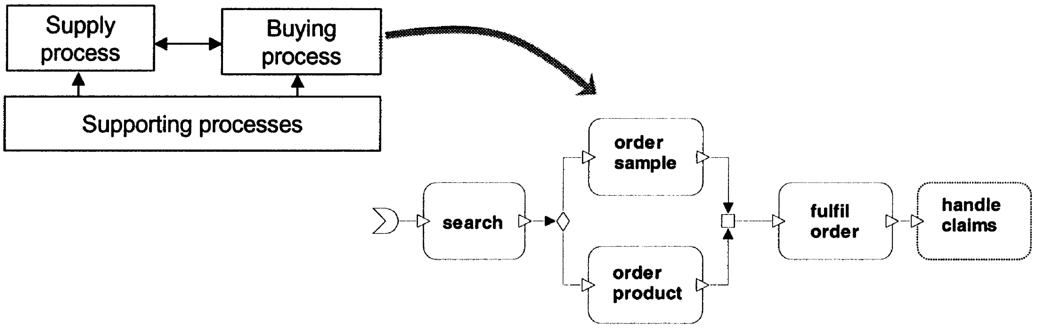

Figure 4. Process architecture (step 3)

Detailed subprocesses (step 4): The subprocesses in the architecture can be considered as placeholders for "process components" each of which has to be described in more detail. A process component may have multiple instances, e.g. one for each of the product categories that are offered. Detailed business rules may be specified as additional information attached to actions (in text, e.g. cancellation policies, allowed modifications for orders). As an example, *Figure 5* shows the detailed process for the order intake of products in stock. The model describes the activities and their dependencies, which may be, e.g., simple causal order, choice (denoted by an open diamond), or parallelism (denoted by a black diamond). The detailed process models also show the operations that the activities perform on data collections (*items*, denoted by parallelograms).

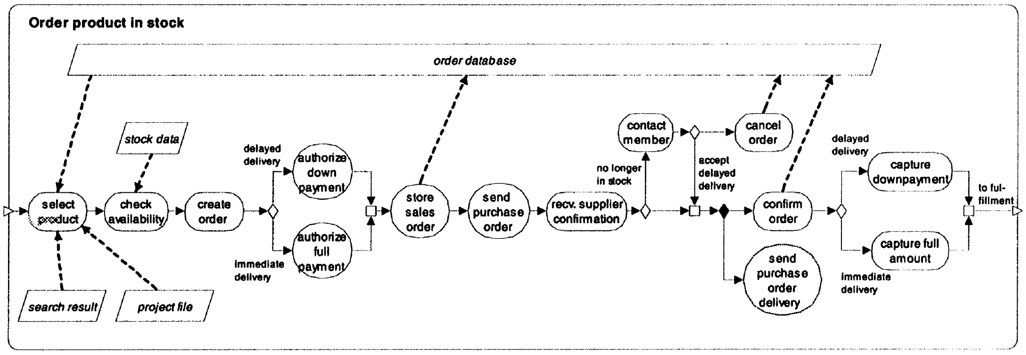

Figure 5. Details of "order product" subprocess for in-stock products (step 4)

Process lanes (step 5): In the detailed processes, each activity can be assigned to one or more of the roles that have been identified before. This assignment, which represents the link between the role models and the process models, can be shown in a comprehensible way by means of "process lanes". An activity in which two roles are involved becomes a *transaction*, depicted by two connected semicircles. *Figure 6* shows the process lanes for the order intake subprocess for in-stock products as an example. Process lane models are not directly used as input for the application development, but mainly serve for *communication*, and are used to check the completeness and consistency of the process models. For example, there should be an explicit interaction for every information transfer between different roles, which may require system support.

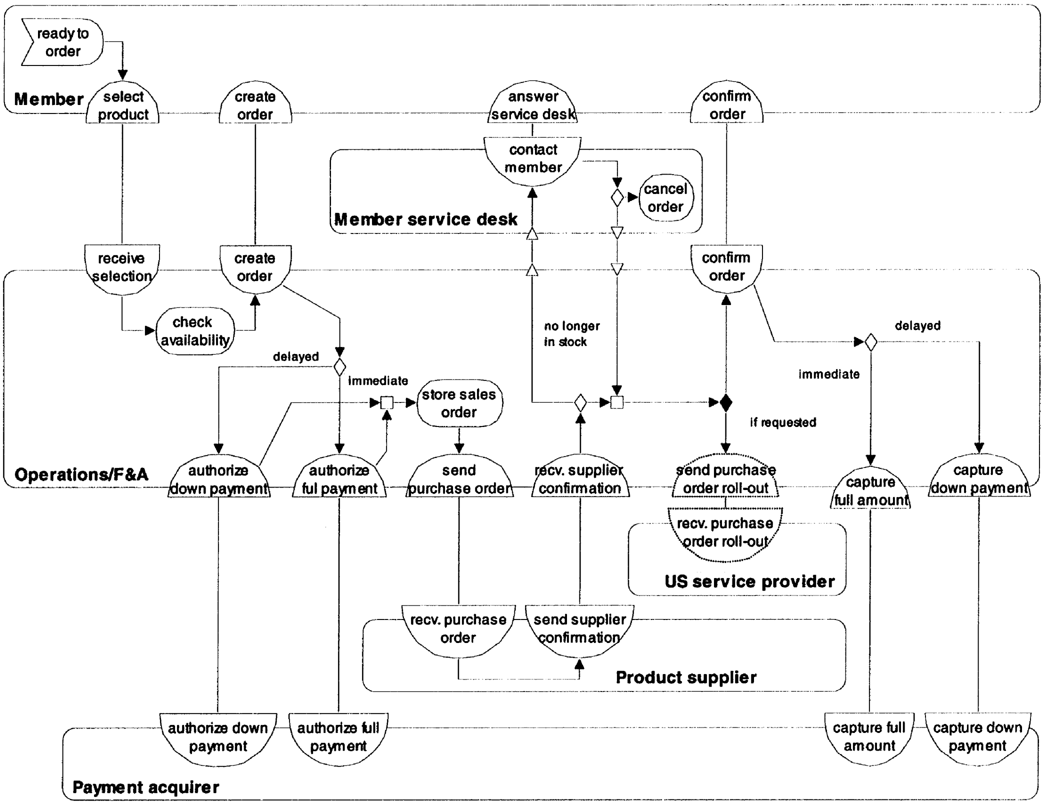

Figure 6. Process lanes for the order intake subprocess for in-stock products (step 5)

System support for actions (step 7): For each action, the level of system support is determined: it is decided whether it is performed manually, completely automated, or partly automated. *Figure 7* shows an examle. A brief description of the type of required system support is also provided, but in this stage we do not yet specify the full requirements. This model provides important input for the technical design.

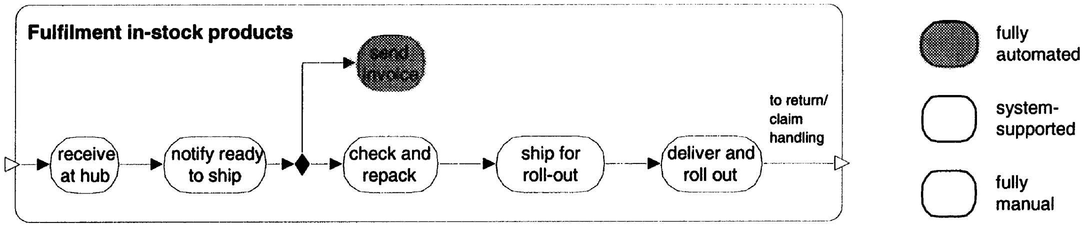

Figure 7. System support for the(simplified) fulfilment process of in-stock products (step 7)

4.3 Functions

The last step of the business design is the construction of a function model (step 6). This model specifies the (business) functions that provide the functionality needed to realise the processes. These functions may or may not be realised or supported by ICT system functionality. A function can be considered a *grouping* of actions from the process models. This means that every activity is always associated with one function. A business designer has some freedom in *how* to group the activities. However, guidelines for this are provided by existing reference models (e.g. for electronic marketplaces (Iacob, Smit, 2001, Schmid, Lindemann, 1998), and by the way in which legacy systems are structured. *Figure 8* shows the top-level functions that we identified. A function model also depicts the most important flows between the functions, both physical (thick lines) and information flows (thin lines).

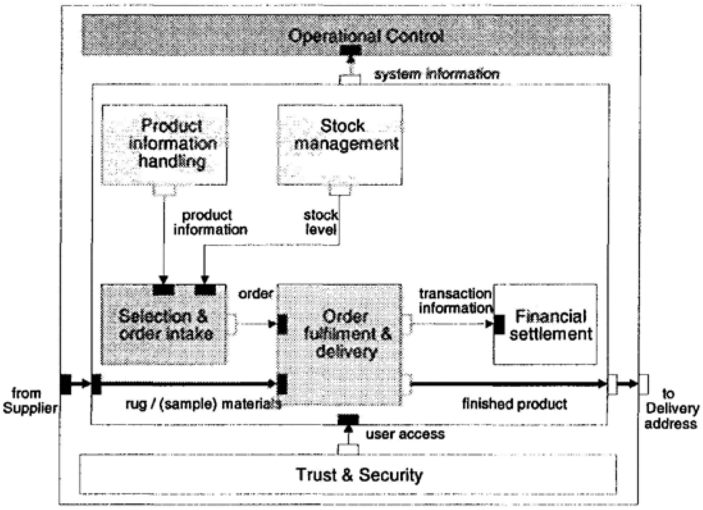

*Figure 8.*Top-level function model (step 6)

The function model is refined one step further by identifying subfunctions within each of the top-level functions. As an example, *Figure 9* shows the refinement of the "selection & order intake" function. In addition to subfunctions, a function may also include (information or physical) *stores*, denoted by triangles.

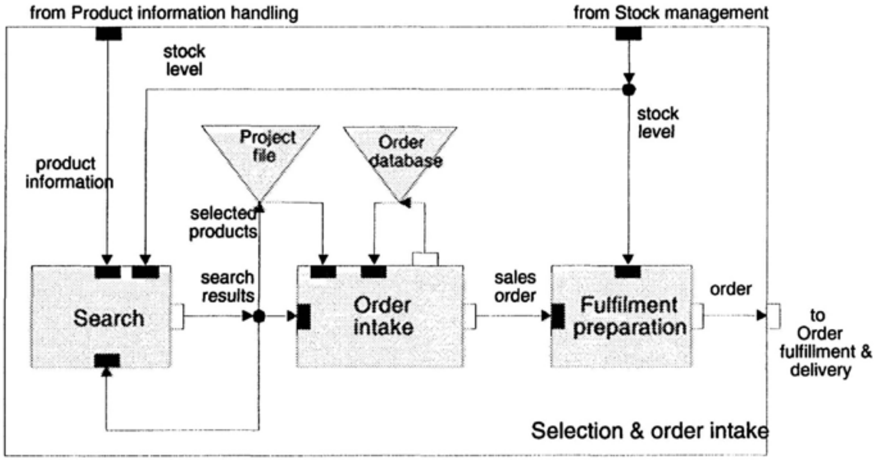

Figure 9. Refinement of the "selection & order intake" function

Function modelling serves as a useful step towards the technical design, because it offers guidelines for the structuring of the information systems. In contrast to this, the business processes usually cross several (business and system) functions.

5. TECHNICAL DESIGN

The technical design covers the cornerstones on the right-hand side of the RSD framework. The results of the business design are input for the technical design, in particular the system support for actions (Section 4.2) and the function model (Section 4.3) were essential input for the technical design.

Also, since Tapestria is an existing marketplace that is being extended, it is important to recognise that there is already a technical architecture in place and that important technical design decisions already have been made. These are part of the legacy system description (see *Figure 10*, step 8) and the technical architecture (step 9), and input for the technical design of the rugs extension.

The technical design aimed at specifying the new functionality needed (new in comparison with Tapestria for fabrics) for extending front office and back office for rugs:

1. A specification of *technical requirements* for both front office (web site) and back office.
2. A *data model* to support storage and retrieval of rug orders, rug configurations, rug components, configurable parameters, etc.

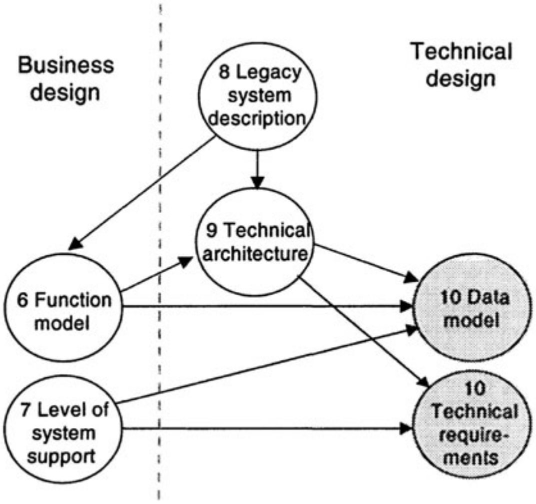

Figure 10. Overview of technical design approach

5.1 Technical requirements

Based on the functionality already available in Tapestria for fabrics, we focussed on the following categories of requirements to be realised by the Tapestria site (see also *Figure 11*):

- *Product configuration*: Product configuration defines how rugs can be configured by choosing materials and setting parameters like color and size. Related to function Selection & order intake/Search (*Figure 9*).

- *On-line price calculation*: Since rugs can be configured on-line, price calculation cannot be done off-line. An on-line price calculation module is specified that builds list prices from precalculated list prices per unit measure. Related to function Selection & order intake/Order intake (*Figure 9*).
- *Order configuration*: The configuration of a rug order including several types of additional services and products. Related to function Selection & order intake/Order Intake (*Figure 9*).
- *Supplier browser*: The supplier browser enables the supplier to price articles and materials, and maintain stock level indicators and delivery times. Related to function Stock management (*Figure 8*).

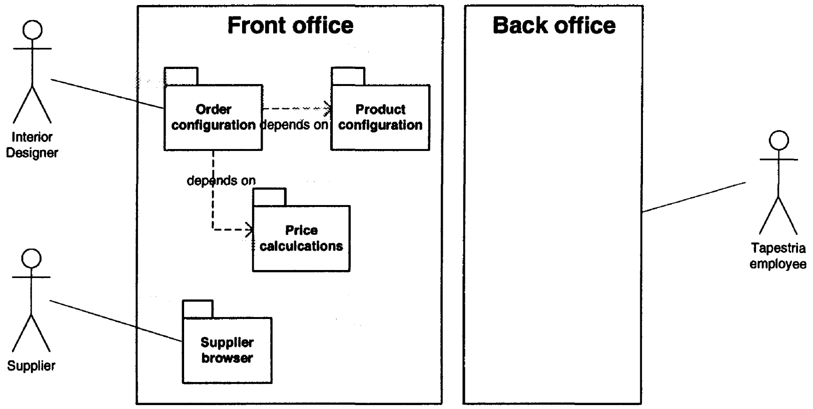

Figure 11. Overview categories of requirements (steps 8 and 9)

Besides the traceability to the function model, we also maintained traceability to the business processes. Each functional requirements has a trace to the processes and their activities. In order to establish a relation to the data model design, of each requirements the use of database tables was specified in terms of read and write.

The requirements document is the basis for the realisation of the web application and back office software.

5.2 Data model

The data model consists of a logical data model that defines concepts and their relations and a physical data model that specifies database tables. Here we focus on the logical data model. Logical data models are specified using UML (Booch, Jacobson, Rumbaugh, 1999).

An important feature of Tapestria rugs is the ability to design and order customised articles. In order to allow members to configure their own articles, we need a more general notion than *article*. The notion we use is *product*. An example will illustrate the difference: suppose we want to sell paint and want the customer to configure the amount and the color, e.g. a liter of yellow paint. Then 'paint' is the product and 'a liter of yellow paint' is an article. We call the amount and the color the *product configuration.*

An article is the physical result of a product and a product configuration. Only articles can be produced, ordered or stocked. In the design the article is the hinge upon which the other concepts depend. *Figure 12* illustrates this.

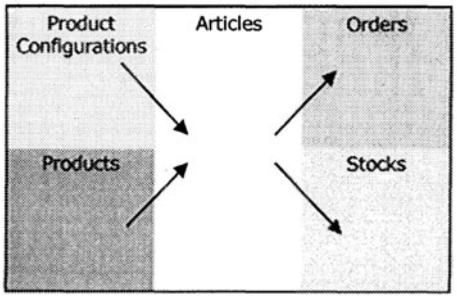

Figure 12. 'Article' as the central notion of the data model (step 10)

On the basis of this idea, a logical and a physical data model were developed.

6. CONCLUSIONS

In this paper we illustrated how RSD, a methodology to support the rapid development of network-based applications in an inter-organisational setting, can be applied in practice. We used the methodology in the development of an extension of Tapestria, a web-based marketplace for interior products.

RSD and its (tool-supported) modelling techniques proved very suitable for documenting business processes, and to support the communication between the different parties involved in the development process. The models also help to check whether the process descriptions are complete and consistent. The business process models provide a good starting point for the technical design, as a way to structure the technical requirements.

The Tapestria pilot project also identified a number of limitations of RSD, thus providing valuable feedback for the methodology.

- One thing that is still missing is a step-by-step approach ("pathway") for developing a web-based application. However, the approach that we followed in this trajectory is a good starting point for the description of such a pathway.
- Also, the support for bridging the gap between the business design and the technical design is still underexposed.

- exercise To maintain traceability between models proved to be a troublesome in some cases.

- RSD as a methodology, and its supporting modelling language in particular, are strongly focussed on realisation of networked enterprises while underexposing the initial requirements specification that captures and subsequently formalises the original idea. Also, RSD must facilitate feasibility studies from financial (profitability), operational and technical perspective.

7. ACKNOWLEDGEMENTS

The work on which this paper reports is supported by the a BTS grant of the Dutch Ministry of Economic Affairs.

We thank Erwin Fielt and Wil Janssen for their comments on an earlier draft of this paper, and Hunter Douglas for permission to publish this material.

8. REFERENCES

Booch, G., I. Jacobson and J. Rumbaugh, *The Unified Modeling Language User Guide*, Addison Wesley, 1999.

Bruins, A. and M.W.A. Steen, *Electronic Procurement: A Functional Reference Model*, Tech. Rep. GigaTS/D3.1.2, Telematica Instituut, Enschede, the Netherlands, 2000. *https://doc.telin.nl/dscgi/ds.py/Get/File-15270/*.

Buhr, R.J.A. and R.S. Casselman, *Use Case Maps for Object-Oriented Systems*, Prentice Hall, 1996.

Clark, C.J., *The UN/CEFACT Unified Modeling Methodology:* An Overview, UNCEFACT/ TMWG/N097, March 2000.

Franken, H.M. and W.M.P. Janssen, *"Get a grip on changing business processes"*, Knowledge & Process Management, Wiley, winter 1998.

Gordijn, J., *Value-based Requirements Engineering, Exploring Innovative E-commerce Ideas*, Ph.D. thesis, SIKS Dissertations Series No. 2002-8, 2002.

Iacob, M.-E. and A. Smit, *Electronic Markets: Classification and Reference Model,* Tech. Rep. GigaTS/D3.1.3. Telematica Insituut, Enschede, the Netherlands, 2001. *https://doc.telin.nl/dscgi/ds.py/Get/File-15266*.

Janssen, W.M.P. and M.W.A. Steen, *"Rapid Service Development: An integral approach to e-business engineering"*, in S. Murugesan and Y. Deshpande (eds.), Proc. 3rd Workshop on Web Engineering at the 9th International World Wide Web Conference, Amsterdam, the Netherlands, May 2000.

Kaplan, S. and M. Sawhney, *"E-hubs: The new B2B marketplaces"*, Harvard Business Review, May-June 2000, pp. 97-103.

Kruchten, P., *The Rational Unified Process – An Introduction*, Addison-Wesley-Longman, 2000.

Martin, J., *Rapid Application Development*, Macmillan Publishing Company, 1991.

Schmid, B.F. and M.A. Lindemann, *"Elements of a reference model for electronic markets"*, in Proc. 31st Annual Hawaii International Conference on System Sciences, vol. IV, IEEE Computer Society Press, 1998, pp. 193-201.