THE IMPLEMENTATION OF A WEB BASED SUPPLY CHAIN INFORMATION SYSTEM - EXPERIENCES WITH A REGIONAL QUALITY GRAIN PROGRAM

Thomas Hannus, th.hannus@uni-bonn.de
Oliver Poignée, o.poignee@uni-bonn.de
Gerhard Schiefer, schiefer@uni-bonn.de
University of Bonn
Chair for Business Management Organization and Information Management

Supply chain management and communication of product-related information across the chain is gaining importance in agribusiness. The use of information and communication technology (ICT) and especially internet-based technologies to communicate information is one approach to supporting the information needs of supply chain management. In this paper, experiences made during the development and implementation of a prototype for a chain information system in the regional quality grain program “Eifelähre” are discussed.

Strict and clear terms of business within the supply chain – as provided by “Eifelähre” - help to establish well-defined rules for transaction and thereby provide an integrated and straightforward data model. The implementation in the quality program also limits the number of players involved and defines standards for the data collected and the quality of data.

The adopted development approach is a combination of established principals of software engineering and prototyping techniques. Experiences discussed cover essential areas of the implementation process: System analysis, semantic modeling, data modeling, the requirements of information architecture and considerations on the ergonomics of use. Methods of testing are reviewed in the context of the implementation process.

Keywords: Supply chain management, information system, data modeling, development process

1. Introduction

Due to various developments on markets for agricultural products, demands on effectiveness and ability to cooperate are increasing. This process generates the need to enhance the organization of information and communication structures across the chain (SCHIEFER (2002) p. 329). Information and communication technology (ICT) and especially internet-based technology can be suitably employed to support these efforts. This paper focuses on experiences made during the development and introduction of a prototype for a chain information system in the regional quality grain program "Eifelähre".

"Eifelähre" provides ideal settings for the implementation of a prototype, as product- and market-related complexities are reduced: Well-defined rules for transaction are decisive for the design process of an integrated and stable data model. “Eifelähre” provides strict and clear terms of business within the supply chain which help to establish these rules. The heterogeneity of participants in the global market for grain and grain products causes various problems concerning data-gathering and data-integration. Implementation in the quality program limits the number of players involved and defines standards for data collected and quality of data.
2. Development Process

Literature provides a multitude of models for the overall process of application development. A general partition can be made into phase-oriented methods with a mostly linear and stepwise approach and prototyping methodologies.

Phase models follow a sequential route from problem- and system-analysis to a tested and implemented application. Classic phase methodologies include a minimum of five steps from problem analysis to full usage and maintenance. Results of one phase are usually the starting point for the next (PAGEL / SIX (1994)).

The use of phase-oriented methods encourages a systematic and structured proceeding. The whole process is broken down into defined stages of development which always end with a defined result. This separation facilitates efficient project management as competence and liabilities are clearly defined. Another benefit is the fact that an individual phase is designated to the gathering of user requirements at the beginning of the production process (OTT (1991) p. 9 f.).

Criticism of these process models is multiple, oftentimes focused on the fact that stepwise methodologies tend to be inflexible and lack user integration (VELDER (1996)). The process of software development however is in many respects non-linear and dynamic: Creativity and flexibility are needed in order to solve the problem of the product to be developed (OTT (1991) p. 11). Strong interaction between future users and developers is needed to assure that the final product meets the users demands. Future users will, for example, be able to describe their expectations more precisely once they get their first hands-on experience.

Prototyping-related techniques pose alternatives to static approaches. Three types of prototyping methodologies are described in literature (e.g. BUDDE et al. (1992) p. 38):

- exploratory prototyping to specify unclear projects and problems – various designs are implemented, no premature restriction to one approach should be made.
- experimental prototyping focus on the technical implementation – users may further specify their ideas about the way the problem is dealt with – breadboard constructions are employed.
- evolutionary prototyping is a development process which continuously accompanies the application.

VELDER (1996) describes the advantage of prototyping as providing permanent feedback as adequate pre-versions are supplied throughout the development process. Through intense communication between IT supplier and IT user, the achievement of development objectives is improved, especially when conceptual formulations are unclear and user expectations are evolving. As mentioned below, one of the major advantages is that changes in user needs can be taken into account easily. Changes in user expectations tend to occur during the process, as options and chances emanate when user groups intensely review their expectations and needs. Disadvantages include the problem that trial and error approaches can lead to imprecise specifications and generally unstructured proceeding (VELDER (1996) p.18).
The structure of the problem at hand therefore implies the method or method-mix to be used in the development process. Determining factors for the “Eifelähre” tool can be described in this context as follows:

- Data and data definition is provided mostly by the quality program.
- User groups and user community are predefined by the closed structure of the program.
- Few to no standards exist for this type of software as a whole.
- Application structure and graphic user interfaces (GUI) requirements are in some aspects unclear and developing.
- The tool is projected as a field ready prototype for the chain as well as for scientific needs.

Due to these specifications, the proceedings of developing and implementing the web tool for “Eifelähre” generally follow a stepwise approach to ensure strong structuring and documentation of results. Prototypes are widely employed whenever communication with users on the issue of application specifications is necessary.

The development process (see fig. 1) was initiated with a system analysis, a further specification on the basis of HTML prototyping and a detailed requirements definition. These results of these three phases set the framework for a prototype database developed in MS ACCESS. A form-based GUI was then implemented in compliance with the final HTML prototype to verify the databases correctness and completeness. An overall web design was then established and implemented in HTML. The so developed templates were used for the linking of the web front-end to the database using ColdFusion as the connecting technology. As a last step the database was migrated to a more powerful database system to comply with everyday needs of the chain.

![Figure 1: Development Process](image)

In phases marked by dark gray boxes, prototypes were amply used to achieve the development goals of the step. During phases marked by light gray boxes prototypes were as well applied but only in a limited extent. No prototypes were used in phases indicated by white boxes.
During the development process the tools outlines were increasingly specified to future users. Relevant steps on this path are marked on the x-axis of figure 1. X1 indicates the point from which on interactivity was established by HTML prototypes. X2 points to the first database prototype which enabled data entry and retrieval, X3 indicates the time from which on users where able to enter and review data interactively over the web.

3. Applied Methods

Based on the established overall proceeding, methods had to be found to support individual steps. The following sections describe the key methods with respect to their significance in the development process.

3.1. System Analysis

System analysis was used to examine chain processes and to extract relevant parameters which are communicated along the chain. System analysis was applied in a two-step process of recording and analyzing the status quo.

Sources for recording were players in the chain, organizational documentation and standardized forms used to communication within the chain. Content of documentation was information flow, document flow, flow of goods and communication interfaces. Sets of data to be gathered had already been developed within the quality program “Eifelähre”. However, add-ons to these parameters were established in expert reviews, through a chain wide HACCP study and an FMEA (POIGNÉE / HANNUS (2003)). Means of documentation were data flow plans and classic flowcharting. Methods applied were interviews and document evaluation.

The analysis of the actual state was done by developing a data matrix for in- and output, strengths – weaknesses - analysis and definition of requirements. Based on this preliminary definition of requirements exploratory prototypes were developed in HTML using highly integrated editors / generators in order to keep development cycles short.

3.2. Data modeling

Data modeling was done in a two-step approach using the Entity Relationship Model (ERM) as established by CHEN (1976). In the first step, a semantic model (business data model) was developed which gave no consideration to the logical and physical implications of the database system to be used. In the second step, this model was translated into one suited to work in a specific database system.

During the developed of the business data model (figure 2), the focal point was to systemize information to be handled. Types of datasets (entities) were identified in terms of attributes, values, dependencies of data (relationships) and cardinalities. Cardinalities quantify the relationships between datasets. Basic cardinalities are “one to many”, “one to one” and “many to many” relationships. Gathering this information and verifying the developed design was done by interviews, visualizations in simplified ER Charts and through questionnaires.

The concept of identifying keys is also fundamental for efficient data handling throughout the chain. TRIENEKENS and BEULENS (2001) describe the process as information decoupling: Only an aggregated piece of information which is unique within the system is handed on to the following chain member. This unique identifier establishes the link between aggregated and detailed information on the product traded. Within the “Eifelähre” model, these identifiers are exchanged on every stage in the chain. In order to provide tracking and tracing, identifiers therefore need to be related to their predecessor and successor.
The model that represents the information flow within the chain was then translated into the logic data model suited to perform within a relational database system. Even though the ER model is in no way limited to the implementation of the model in a relational database, it provides ideal settings to do so. The two structural elements of the ERM (entities and relationships) are reproduced within the structure of the relational data model. Relationships are transferred using identifying keys as links (SPILKE (2002) p 134 f.).

As a last step towards the working data model, data definitions were made. This issue had already been approached during earlier phase of the development process and information was mostly provided by the HTML prototype. However standard values, minima and maxima had to be verified at this point.

3.3. Information Architecture and ergonomics of Use

The two main considerations in this process were the structure within the site to enable easy accessibility and the individual screen design.

Menu structures of the application were deduced from the business model’s function structure. A first implementation was made within HTML prototypes and gradually refined.
During the later implementation of the ACCESS based prototype workflow in the chain was also considered. Elementary processes were prototyped and tested in multiple scenarios. The so developed site structure was then transferred into sitemaps as a basis for development in the ColdFusion prototype.

A well-structured layout supports fast and easy intake of the information delivered (ZEIDLER / ZELLNER (1994) p.58). One widely accepted layout is based on a three row vertical segmentation of the screen (STRAY (1994) p.86 f.): An orientation area at the top of the screen, the workspace in the middle of the screen, its layout according to the applications requirements and the activity area that allows user interaction.

Web design commonly uses a navigation element that is placed on the left hand side of the screen. The use of web-based technology implies the use of such navigation, as it is familiar to the user in this context.

3.4. Tests

Numerous authors mention that apart from analysis, design and programming, testing as well as validation are of paramount importance for the projects success (e.g. GUMM (2002) p.677). LIGGESMEYER (1990) proposes that every phase of construction is to be followed by a phase of analytic quality insurance. He later establishes a classification which comprises three main categories: verifying methods give prove of the programs correctness, analyzing methods quantify specific attributes and testing methods serve the detection of errors.

All methods applied in the development of the “Eifelähre” web tool were either verifying or testing. HTML prototypes were tested by future users against conformity to their expectation. Before testing the database with future users, internal tests were performed in the development team. A laboratory test was then conducted to check the databases within a multi user environment. Users were therefore asked to execute standardized tasks and evaluate the system in terms of correctness, conformity to expectations and the ability to assist in performing the assigned tasks.

In a last step, a field test was conducted to ensure proper working of the ColdFusion based prototype. To do so, a limited number of chain members was introduced to the tool and asked to do the hot testing of the web tool. User tracking and error transmission ensured automated documentation of these tests.

4. Conclusions

The introduced methodology aims at providing a complete and integrated process model and toolkit to develop and implement a web based supply chain information system. Using a stepwise approach with prototyping cycles within certain stages of the process, the method provides intermediate results and a strong organizational structure. Still it gives feedback and feed forward to future users and developers thus overcoming the problem of diverging expectations. The use of well-established methods within the process makes little demands on the method knowledge of the developing team. Visualizations can in some cases also be used for communication with future users. Integration of results from phase to phase prevents structural interrupts in the overall concept. Due to this integration, a stepwise refinement can be utilized in most aspects of the development process. System analysis as the initial step in the process and the continuous verification of results ensures optimized adaptation of chain requirements.
LITERATURE


