Supporting Regional Sustainable Development by IT-Assisted Communication and Cooperation

Stefan Naumann

Abstract

In the paper we describe a software system that supports a community in ordering, purchasing and delivering organic products like food, wooden building material etc. The software supports especially the communication and cooperation among the members/consumers to improve local sustainable economy and to allow orders in organic wholesale (what requires summing up single orderings to wholesale bundles).

1. Introduction and Context

One possibility to procure regional organic food and products as contribution to sustainable development (Word Commission 1987) is to organise oneself in food-coop (Bundesarbeitsgemeinschaft 2000). If this community uses information technology (IT) to support procurement and transactions, we can talk about a virtual community (Rheingold 2000, Schubert 2000) with ecological interests. The advantage of these ecological consumer communities (Hummel/Lechner 2000) is the possibility of buying from wholesale and regional organic farms. However, on the other hand the members have to organise everything like purchasing, prepacking and distribution within the group.
An electronic supported food-coop is one possibility to bring to market organic food besides other sales markets, see e.g. (Nachtmann 2002). From an ecological point of view the following advantages of a food-coop result:

- Organic farming and its less environmental impact is supported
- Traffic is reduced because the products are produced and distributed within the region
- Direct communication between producers and consumers and among the consumers enforces indirect side effects like car sharing / transport sharing or like product improvements, in the sense of e-support, see e.g. (Fichter 2001, p. 114).

In rural regions the distances between community members are large, and so the input to cooperate and decide together. An obvious idea is to support ordering, communication and distribution of food-coop goods by a software system. Furthermore, this system can be extended to a software that assists sustainable everyday life requirements like product ordering, communication, social contacts etc.

2. **Problem Scope**

Common purchasing requires communicating corporately. Before using the described new software-system “eFood-Coop” (Naumann 2001) the catalogue with organic products was handed round through the members of the purchasing community (the food-coop). Hence, it took a lot of time (several weeks) to decide and then to purchase the products from wholesale.

The second problem is to optimise the bundles. As no private person normally wants to buy e.g. 25 kg of wheat, it is necessary to collect and fit all requests for parts of the bundle.

Therefore, the system should assist the food-coop members in communication and cooperation within the group to achieve completed wholesale sales units and should therefore be able to recognise completed bundles.

3. **Procurement Process**

Before using the software, a food-coop ordering transaction was handled as followed:

1. The organic wholesaler sent his (printed) catalogue to the ordering coordinators (the person or group which organises the complete process)
2. The catalogue was handed round serially and every coop member marked his requested articles and whether he / she wants the offered sales unit completely or only a part of it (e.g. 4 parts from a sales unit with 12 parts of 500 g noodles)
3. When all members had chosen from the catalogue, the requests were summed up by the ordering coordinator. Because the members could give additional information to part orderings like “at least” and “at most”, he was able to decide which sales units could be ordered from the wholesaler. “At least” means, that they want the requested quantity or more (up to the complete sales units), if not enough other members are interested to fulfil the bundles; “at most” means a maximum delivery of the requested parts. In case of not enough requests from other members, it is possible that the member won't get any part.

4. After the compilation of all requested sales units, the ordering was sent to the wholesaler and afterwards delivered to a central point (normally to a member of the food-coop). If the delivery differed to the order, another step similar to step three was necessary. Then the delivery had to be prepackaged and picked up by the members.

5. In the last step, invoices were sent to all members referring to the effective delivery.

This practice had some disadvantages: Termination took a long time, coordination of the fulfilment of bundles was elaborately, and procurement from more than one deliverer within the food-coop was hard to manage.

4. Methods

Last year a distributed Java application was developed to meet these requirements and solve the mentioned problems. As the idea is to purchase together, the users can download the catalogue with products and are able to order. One problem is here, that the considered suppliers do not use standardised catalogue formats like BMEcat (Bundesverband Materialwirtschaft, Einkauf und Logistik 2002) or xCBL (Commerce One 2002), or electronic classification systems like eCl@ss (Institut der deutschen Wirtschaft Köln 2002). Hence, a proprietary format had to be processed and read in via a csv-file. In addition, small farms and companies do not often have an electronic catalogue file, so it has to be created by the ordering coordinators.

We decided to set up a distributed application and not to use a web-interface, since the user community is limited and we had more programming possibilities. In addition, we could offer to order offline, as all data from the database server are replicated locally.

In the computer-assisted procurement we can differentiate between the following steps:

1. First the product data with additional features like country of origin etc. has to be sent from the wholesaler and the producers to the food-coop and then converted to fit the internal database schema.
2. For each supplier a deadline has to be fixed. Up to this date all members can make requests, write down comments and communicate about the common procurement.

3. After the deadline, the requests have to be summed up by an ordering coordinator. Since there is a flexibility within the requests by “at least” and “at most”, it is easier to decide whether a requested article will be ordered or not. The coordinator is assisted by the software with appropriate algorithms to check which bundles are uncompleted and therefore which requests have to be modified.

4. After ordering and delivering, the articles have to be prepacked. In order to support this process, the software can printout a list of all requests, sorted by articles and consumers.

4.1 Communication and Cooperation

To reinforce communication between the participants, it is possible to write down comments. We can classify these comments into two groups: comments directly referring to the product (like additional information, recipes) and comments which belong to the actual order.

Fig. 1: Survey of the complete catalogue.
For instance if one person wants 6 of 12 parts from a rice sales unit, and another person wants 6 of 12 part from another sort of rice and both requests are "at most", an ordering by wholesale is not possible. If they communicate through the comment option in the application, they may decide to order the same article (so the same sort of rice) and the bundle may be fulfilled.

The system supports several suppliers. Figure 1 shows the electronic catalogue with the different suppliers and their products. The food-coop members can see all suppliers, all product groups and their own orderings. Additionally the total requested amount of a product is visualised, and they can see if there are any comments to a concrete article.

After having ordered, the consumers can have a view of their requests with a corresponding mask "My orderings". Here they can see all their requests and the resulting orders, as well as the bundles that are completed at that moment and fitting the requests, or were a modification is expected.

Fig. 2: Mask that supports the ordering coordinator. On articles right he can see whether the sales unit is (in-)complete (column “Bestellung”=Order)
4.2 Optimisation of the Bundles

After the upload of all user requests to the server has been finished and the deadline is over, the ordering coordinator is able to look whether the sales units are completed. He is supported by the mask shown in figure 2.

We distinguish between “requested” (that means the user requests for (parts of) products) and “in fact” (that means how many parts will be ordered from wholesale after optimisation). The ordering coordinator is authorised to change the “in-fact”-ordering, but not the “requested”-ordering. At any time, he can start an implemented algorithm to compute which bundles are completed and which are not.

Fig. 3: Distributed architecture between suppliers, consumers and provider.
4.3 Technical View

We implemented the software with the programming language Java to be secure in internet-transactions and to be platform-independent. The data is stored locally in files (serialised objects), on the server (using Java classes to wrap objects to relational tables) we run a relational database. Figure 3 shows the distributed architecture.

As a classic three-layer architecture, we have a client, a server application and a database. Main tasks are:

- **Client:** Displaying data and product information, collecting user data (requests, orders and comments) and interaction with the server. Additionally, checking bundles whether they are complete, because every member could be the ordering coordinator
- **Server:** Acting as the middleware between clients and database, wrapping objects from the clients into relational database tables
- **Database:** Storing the persistent data and ensure data safety and security

5. Conclusions and Outlook

With the new application, a parallel ordering is possible; so more people can participate at the food-coop system. After using it the first time in December 2001, more (new) consumers wanted to order organic food through the efood-coop software. Ordering and completing bundles is also easier, since the algorithm computes which sales units are fulfilled. A main problem is the electronic data exchange between software and suppliers. The considered small and medium-sized businesses and farms do not often use ERP-Systems or utilise standard electronic catalogue formats and classification systems. Here the interaction can be improved, for example with electronically processable delivery notes.

The next step will be to extend the offered products by non-food articles and services which can be lent or exchanged in a local exchange trading system (LETS, Hoffmann 1998). Then we have a virtual regional shop for organic products and environmentally friendly services that supports a virtual community with a sustainable living culture called “decentral eco-village” (Naumann 2001). This community subsists in a bounded area and is able to reduce environmental impacts through traffic and to avoid rebound effects because of the decentralized structure.

Actually we focus on two research topics: Using the software to support a green box scheme (Kreuzer 1996), and the question how algorithms and data structures can help the community members to transport people and goods from A to B within the decentralized community. Here concepts like links to public transports, agencies for arranged lifts and car pools have to be considered.
Bibliography

Bundesarbeitsgemeinschaft der Lebensmittelkooperativen (Hrsg.) (2000): Das Food-Coop Handbuch, Eigenverlag, Bochum


Kreuzer, Kai (1996): Bio‐Vermarktung. Vermarktung für Lebensmittel aus ökologis‐cher Er‐zeugung, Pala‐Verlag, Darmstadt


Rheingold, Howard (2000): The Virtual Community, MIT‐Press, Massachusetts
