A Distributed Logistic Support Communication System

V. Gruhn, M. Hülder, R. Ijioui

Chair of Applied Telematics / e-Business[1], Computer Science Faculty, University of Leipzig
E-Mail:{gruhn, huelder, ijioui}@ebus.informatik.uni-leipzig.de

L. Schöpe

Informatik Centrum Dortmund e.V., Abt. Softwaretechnik
E-Mail: lothar.schoepe@icd.de

Abstract. The project „Mobile Spedition im Web (SpiW)[2]" presented here, develops such a mobile communication system, which focuses on the driver/dispatcher interaction. The main goals are integration with legacy logistics software and the possible use of new telematics and communication techniques. To achieve these goals, a component based architecture allows the later change and extension of components, making it possible to add new features to the system as they become available. A mobile workflow server supports the adjustment of business processes to individual needs.

1. Introduction

Truckage companies take advantage in business, when they can perform transports fast, securely, economically, and in time. "Time more and more becomes a critical component in freight transportation" [ErWa97]. This advantage is even more important, as due to globalisation and the extension of the European Union the number of truckage companies rises. Truckage companies that can achieve the named goals more efficiently due to the use of mobile communication can gain more trust as well as a better customer relationship.

1.1   Problems

According to [ErKo01] the following problems can appear during communication and co-operation between the different roles within a truckage company (drivers, dispatchers, customers) and thus stand in the way of reaching the required goals.

Problems for the dispatcher:

- Discontinuous, oral information interchange between dispatcher and drivers leads to delays and mistakes.
- From the company's point of view drivers are a main source for information, but the information pooled at a driver cannot be transmitted into the logistics software without further manual work.
- Because of missing knowledge abut the transport progress the dispatcher can hardly re-schedule transports.
- Calculations of transport costs can only be performed with a great delay.

Problems for drivers:

- Drivers can communicate problems only by mobile phone most of the time.
- Drivers have little influence on the scheduled tours and possible rescheduling.
- Data transmitted by the papers is often incomplete or even wrong.
- Dispatchers may not be available for questions.

Problems for customers:

- Transport progress is not known to the customer.
- Delays are not to be calculated.

By the development from location based acquisition towards a mobile acquisition and transmission of information and data, important information can be made available in time. This information may provide solutions to the named problems, and these solutions may not only help for the dispatching processes but also for the strategic fleet management.

Ideally such a communication system ought to provide a generic interface to logistics software systems and thus provide the advantage of being integrated into different logistics software systems. That way a clear cut between logistics software and communication system is made, proving the opportunity for truckage companies to extend their logistics software system rather than investing into a new monolithic system.

1.2   Usage Scenario

In order to plan the transports the dispatcher makes use of a logistics software system. To communicate with the drivers, the dispatcher usually uses paper forms or telephones. Using these means of communication exclusively can lead to loss of information or delays which prevent the dispatcher from reacting appropriately to events that happen to the drivers. The communication system introduced in this paper helps to overcome these problems. The communication system described in the following chapters is based on the following scenario: The dispatcher asks the driver to perform a transport. The driver then loads the freight and delivers it to the customer. This scenario differs from other parcel or express services in so far as there are very few private people but rather business customers involved who typically get freight with several tons of weight delivered.

To support the communication between drivers and dispatchers, the drivers are provided with mobile devices (e.g. PDAs). On one hand these devices can inform the drivers about the scheduled transports and on the other hand they enable the drivers to report back about each transport's status and problems that might appear during the delivery.

2. Architecture of the Communication System

The communication system's architecture consists of three components: mobile devices (e.g. PDAs), stationary devices (e.g. PCs), and an application server. The mobile devices connect to the application server via a wireless telecommunication system (GSM, GPRS, HSCSD, UMTS), whereas the stationary devices use a wired connection (Ethernet, FastEthernet) to connect to the server.

The software architecture of the communication system follows the client/server paradigm [Lewa98]. The business logic for working on business objects is provided by an application server [BaGe98], which itself takes advantage of other server components: a workflow server, a communication server, and a database server (see Figure 1: Architecture). The services provided by the server according to the business processes are used by the clients to deliver data to the targeted user.
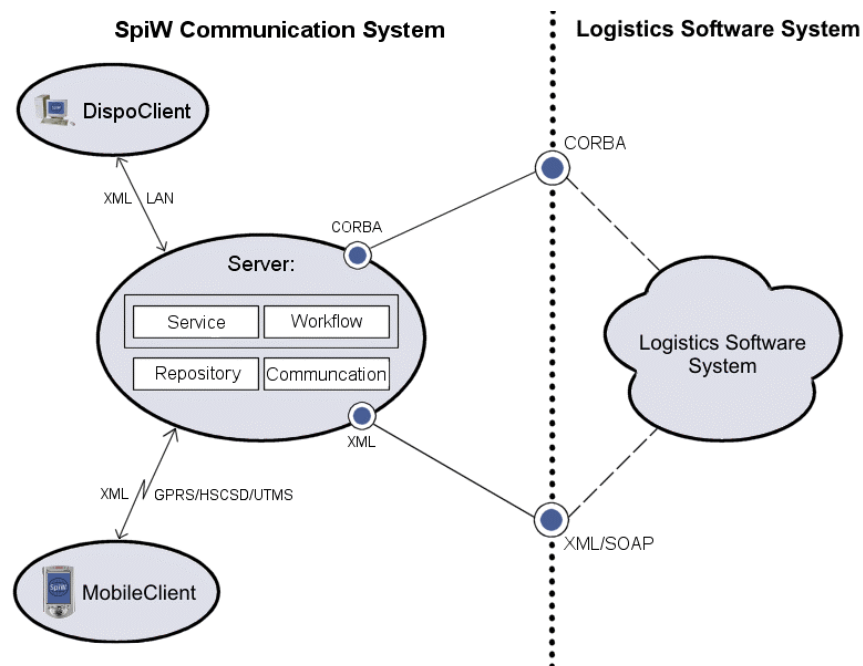
Figure 1: Architecture

The kind of data that is supplied by different clients may differ to a great extend, according to the users' roles and needs. While a driver mainly needs transport data, the dispatcher needs to have not only transport data but also the appropriate managing data as well. Depending on the kind of device used and its particular technical possibilities (e.g. display size) the different clients do not only render the data differently but also the amount of data displayed is ad-

justed. Similarly the usage of the user interface is realised in a different way (e.g. preconfig-ured hardware keys on PDA or mouse control on a PC).

2.1   Thin vs. Thick Clients

If business logic is not only supplied by an application server, but parts of the business logic are realised on a client, clients are called "thick clients" [LeWa98]. On the other hand so-called "thin clients" (e.g. web browsers) do not have business logic on their own, but ex-clusively use services provided by an application server. Although thin clients generally need fewer resources, and therefore appear to be especially suited for devices with limited memory and processing power, thick clients are used on the mobile devices. This is due to the fact that a wireless connection cannot be guaranteed to be available or even stable. But to take care of the requirements mentioned in the introduction, parts of the business logic need to be exe-cuted even when the communication link is temporarily unavailable. After re-establishing the connection, data synchronization has to take place.

It is rather unlikely that the communication link over the wired medium in a local area network (LAN) breaks down for a long time, so that this line of argument is not valid for sta-tionary devices. Still, a thick client is used for the stationary devices, for this client being the logistics management software which contains the complete business logic for rendering and working with the data. For more information on the advantages and disadvantages of thin and thick clients [LeWa98] and [OrHe96] can be named.

2.2   Data Exchange with legacy Software

The communication system defines interfaces for exchanging data with logistics manage-ment software.

The data structure for the data exchange between clients and application server is described by a Document Type Definition (DTD) [Tolk99, BöUl99] and transmitted according to the XML-format. For this purpose attributes of objects need to be transformed into XML data and then send over the communication link. During the transmission additional compression and cryptographic techniques may be used. The receiver then has to search the transmitted XML-document (parsing) and reproduce copies of the original data from the structure and the con-tents of the document.

This process is necessary because the development of application server and clients is based on a component model [GrTh98, Szyp98] used in conjunction with an object-oriented programming language. Due to this, a later extension of the system does not require the data transmission part to be developed again, because only the required classes and the extended DTD have to be deployed on the system. According to the object-oriented paradigm the parser itself can use the objects' methods to produce an XML-code representation of the objects and vice versa reconstruct an object from the XML representation.

3.   Support for mobile Workflows

Business processes that are to be supported by the communication system are described by so-called workflows. A workflow consists of a number of single actions, which can be simple or

complex in themselves again. The actions of a workflow can be carried out in either a sequential, or parallel way, or as alternatives to each other. Workflows can be connected, i.e. initiate or depend on one another.

For any component in the architecture (see Figure 2: Components of the communication system) there are one or more workflows which deal with the creation, manipulation, and visualisation of business objects that are connected with it.
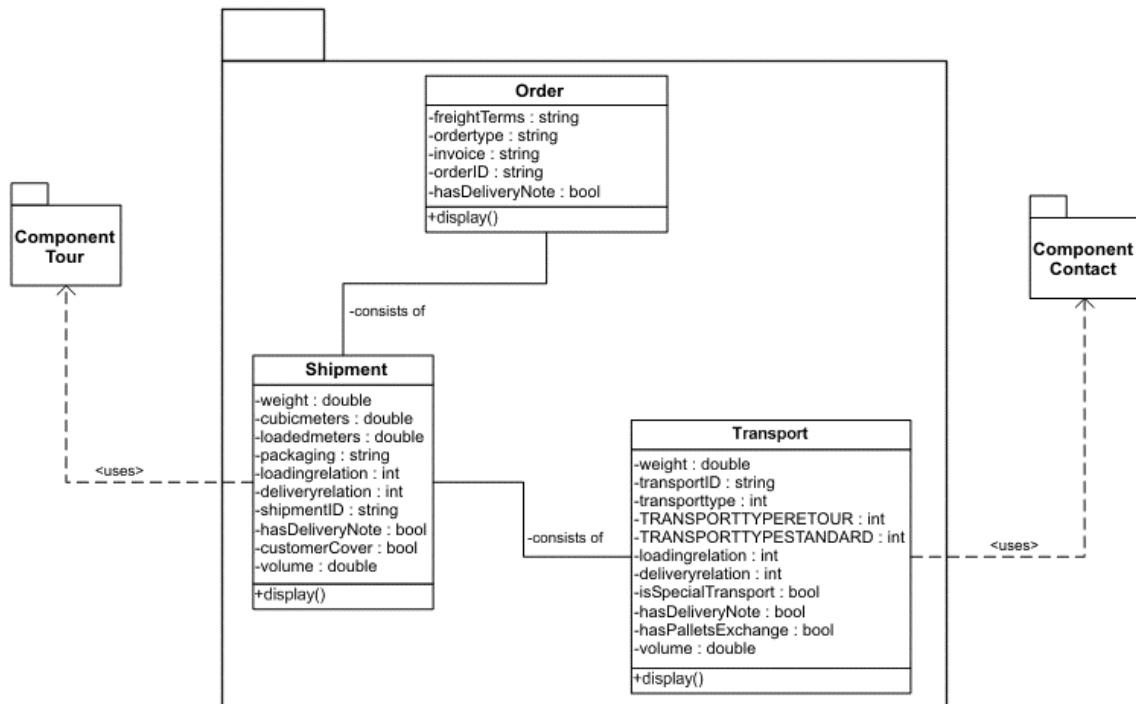

Figure 2: Components of the communication system

According to dependencies between components, workflows of different components may also be dependent on each other. Such dependencies are not always fixed but can evolve after creation or modification of data. Any class of a component contains a so-called display-method which is used to visualize data of that class according to a style-guide and the user interface. This way it is made sure that rendering of data is consistently done in the same way and the user is directed to his goal in the same manner.

In addition to the global workflow which describes business processes, there is local workflow which describes method collaboration within a class. This local workflow can be realised by implementing the display-method of a class.

Any component of the architecture consists of several classes. The components themselves possess a specified interface. The components shield data and store it permanently in a database. Realisation of components is done using an object-oriented programming language (such as Java or C#).
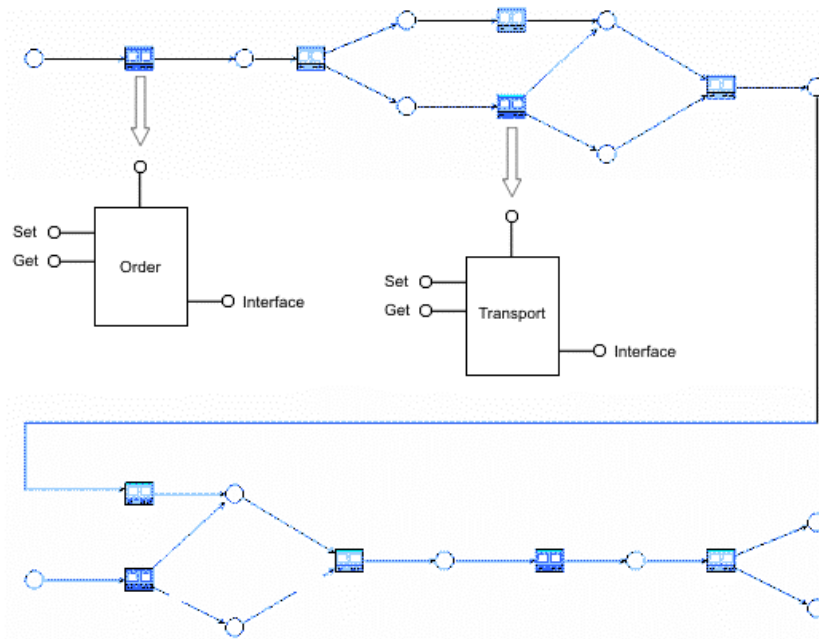
Figure 3: Realised and associated components

Components are associated with actions of a workflow (Figure 3: Realised and associated components). In general there should only be one component associated with each action, but in exceptional cases there can be an association with several components. Such a complex association may only be allowed if changing the workflow is not possible or does not solve the problem. In this case workflow between associated components has to be described explicitly.

If there is a decision to be made depending on the data during the course of an action which defines the following course of the action, there has to be a decision table (Figure 4: Decision table for actions).

Transport delivered	Continue with tour and notify scheduler	Decision depending on workflow
Transport partly delivered		
Delivery rejected	Immediate notification of the scheduler and wait for the reaction	
Transport postponed		
other		

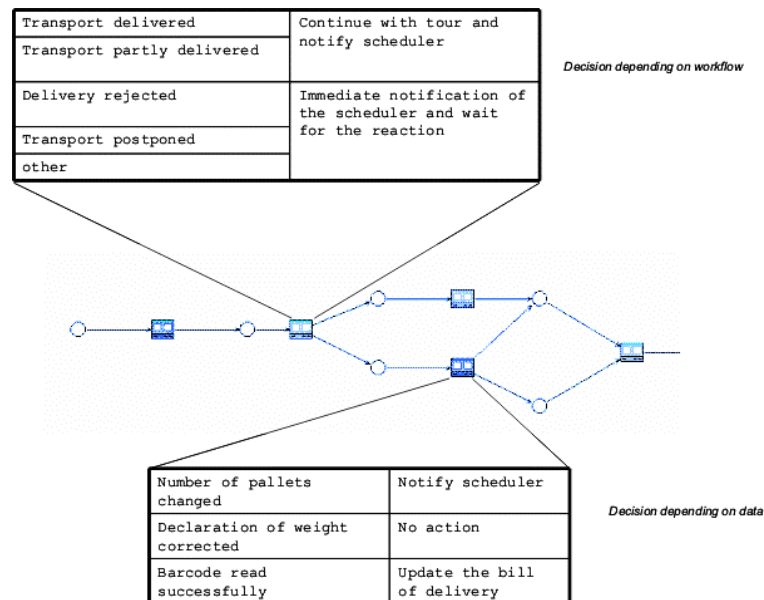| Number of pallets changed	Notify scheduler	Decision depending on data
Declaration of weight corrected	No action	
Barcode read successfully	Update the bill of delivery	

Figure 4: Decision table for actions

The course of workflows is controlled by a workflow server which is part of the application server. Due to the fact that parts of the business processes need to happen in a mobile and

therefore distributed environment – on the clients as well as on the application server – the workflow server has to support the distributed execution and is responsible for data consistency and integrity. Distributed execution may be achieved by either a centralised or a decentralised approach. Considering the wireless connection between mobile clients and application server, and the problems originating from this (loss of connection, network availability, etc.), a decentralised solution appears to be suitable. That means that there is a central workflow server available as well as local workflow servers on each of the clients, although they may differ in the amount of functionality they provide.

Depending on the environment and equipment of the mobile devices the mobile workflows have to be adjustable. For example it is possible to attach a barcode scanner to the mobile device. In this case, the workflow has to handle scanner and read barcodes appropriately, e.g. when delivering goods. In another case, when there is no barcode scanner installed, the delivery process has to leave out the scanning and proceed accordingly, e.g. prompting the driver to confirm that he has delivered the appropriate goods. This mechanism of modelling different workflows for different user and device profiles, and then executing the workflows in the workflow server, allows for fine grained adjustment of the software without changing the source-code.

Whether components may be loaded dynamically at runtime from the server according to the workflow executed or whether all components have to be deployed statically on the clients, depends on the communication techniques (HSCSD, GPRS, EDGE, UMTS) available.

4. Related Work

The complexity of the transportation field is reflected by the richness of research areas, methods, and software. The future of truckage companies, however, lies in providing a more efficient and cost effective transport service. The objective is to use modern technology in order to exploit the full potential of saving costs. More and more goods have to be moved efficiently, quickly and cost effectively by service and transport companies. In order to survive, transport companies must respond quickly to their customers' needs and focus on cost control. Continuous and up-to-date understanding of business processes is essential and requires a reliable and user-friendly system for mobile data communication [Fred02].

Researchers have already developed a number of techniques to solve the communication and fleet management problems of truckage companies. For example, researchers at the University of Bremen proposed a concept for controlling truck fleets through cellphones and the Internet [ErKo01, SiEK03]. Economists at the University of Koblenz are also working on a prototype for supporting logistics processes [Jung01]. The main goal of these research efforts is to provide flexible fleet management and introduce new kinds of services that truckage companies could offer in the future (e.g. location-based services).

There is no universal solution that is suitable for any truckage company, and the systems available today still have a few weaknesses.

Conclusion

Electronic support of transport execution is barely integrated with conveyance systems. Communication during the transport process and afterwards is done either on paper documents or by telephone, both of which require additional work to receive an electronic representation later on. This additional work is not only time consuming but also error prone. The scheduler is not informed well enough about the progress of the transport, and needs to acquire additional information actively himself.

Therefore we see the need for bidirectional communication to exchange transport information in time on one hand, and on the other hand unidirectional communication between mobile devices and other backend software systems to exchange data over a wireless yet stable medium.

Most of the systems available today do not focus on the need of small conveyance companies. Those companies often cannot invest into a completely new software system. Employees would have to get used to the new software and the market situation is so unclear that no one can say which systems are going to last for enough time to secure the investments. Systems that are based on application service providing concepts put up another barrier, because they require hosting vital company data about customers and lorries on the service provider's side. Thus the company becomes dependent on a third party's accessibility and reliability.

The project "Mobile Spedition im Web (SpiW)" which is supported by the German Ministry of Education and Research, aims to reach the two main goals of integration with legacy logistics software and the possible use of new telematics and communication techniques. The component based architecture of the communication system allows the later change and extension of components, making it possible to add new features to the system, as they become available (such as transmission of video data or data gained from board sensors).

Within the project consortium it is not possible to reach these goals completely, especially for the integration with legacy software the interface defined has to be supported by the legacy software developers.

Although the benefit of such a communication system is obvious, it also depends on the costs of acquisition and operation which is even more important. The industrial partners in the project consortium are to ensure that the benefit exceeds those costs.

References

[BaGe98] Baker, S. and R. Geraghty (1998). Java For Business Objects. In: *Developing Business Objects* (Carmichel, Andy, ed.), SIGS, Cambridge University Press, pp. 225-237

[BöUl99] Böhnlein, U. and A. Ulbrich vom Ende (1999). XML – Extensible Markup Language. In: *Wirtschaftsinformatik, Band 41, Heft 3*, Vieweg Verlag Wiesbaden, pp. 275-277

[DaCh03] DaimlerChrysler (2003). *Fleetboard - fleet management.* [available online from: http://www.fleetboard.info/show_page_t1.php4?advantages,e, Last accessed 07 Mar 2003]

[Data03] Datafactory AG (2003). *Webfleet - Fahrzeug und Personenortung im internet.* [available online from: http://www.webfleet.de/en/doc/datafactory_Webfleet-Brochure.pdf, last accessed 07 Mar 2003]

[ErKo01] Erkens, E. and H. Kopfer (2001). WAP-LOG: Ein System zur mobilen Fahrzeugeinsatzsteuerung und Auftragsfortschrittkontrolle. In: *Logistik Management - Supply Chain Management und e-Business* (Grünert, S.), Teubner Verlag, Stuttgart, pp. 293-303

[ErWa97] Ernst, M. and D. Walpukis (1997). Telekommunikation und Verkehr. Verlag Franz Vahlen, München

[Fred03] Frediani, J. (2003). *fleet management via the internet.* [available online from: http://www.daimlerchrysler.de/environ/report2001/pdf/fleet_e.pdf, last accessed 07 Mar 2003]

[GrTh98] Gruhn, V. and A. Thiel (1998). *Komponentenmodelle*. Addison Wesley, München

[Jung01] Jung, J. S. (2001). *Flottenmanagement im Handwerk durch integrierte Telematikdienste*. [available online from: http://www.uni-koblenz.de/~flotthit/Project/overview.html, last accessed: 07 Mar 2003]

[Lewa98] Lewandowski, S. (1998). Frameworks for Computer-Based Client/Server Computing. In: *ACM Computing Surveys, Vol. 30, No. 1*, ACM Press, pp. 3-27

[OrHE96] Orfali, R., D. Harkey, J. Edwards (1996). *The Essential Client/Server Survival Guide*. Wiley Publ.

[SiEK03] Siek, K., E. Erkens, H. Kopfer (2003). Marktübersicht über Systeme zur Fahrzeugkommunikation im Straßengüterverkehr. Submitted to *Logistik Management, 5. Jahrgang, Ausgabe 1*, Germa Press Verlag GmbH, Hamburg

[Szyp98] Szyperski, C. (1998). *Component Software – Beyond Object-Oriented Programming*, Addison-Wesley, Reading, MA

[Tolk99] Tolksdorf, R. (1999) XML und darauf basierende Standards. In *Informatik Spektrum, Band 22, Heft 6,* Springer Verlag, Heidelberg, pp. 407-421