ISBN: 972-8924-02-X © 2005 IADIS

MULTI-CLIENT CAR INSURANCE

N. Serbedzija, M. Fabiunke, F. Schön, G. Beyer
Fraunhofer FIRST
Kekulestr. 7, D-12489 Berlin

ABSTRACT

This paper is a part of a wider research that examines software technologies suitable for solving problems that appear in Internet programming. The focus is in developing multi-client user interface and Internet-based wrappers for the existing software that originally cannot run in a distributed manner. That enables application transformation from a single user's system to multi users' internet and web-based one. The text that follows describes in more details how distributed programming technology may be applied to enhance the use of a car insurance comparison application. To avoid the use of brand names from the practice and their corresponding business models, all system names used here are fictive.

KEYWORDS

Automotive Software, Financial Applications, Electronic Business

1. INTRODUCTION

The Internet technology is having major impact on the development and use of software systems. Most of the larger Web sites offer some kind of Web-based processing and new Internet-based systems are being developed on a daily basis. Examples are numerous: on-line businesses that deploy Web-based programming, large data-base systems and new environments with active and mobile objects that can perform useful tasks for the users [Schramm-Klein 2002, Degel 2003]. Internet-enabled systems fulfill most of the key software engineering promises: they are widely available, cross-platform compatible, parallel and distributed, accessible to all kind of end devices, from mobile computers to powerful workstations on intranets, as well as on the Internet.

The software techniques used for program transformation are illustrated on a concrete example, an application for comparison of car insurances. The car insurance in Germany is a lucrative job. It is estimated that more than 100 millions contracts (20 billions Euros) are made on car insurance yearly [BV 2004, GDV 2004]. More than 80 insurance companies are competing on the market, offering similar service with the price difference of up to 500 Euros, in single cases. Such a situation makes insurance comparison and mediating in finding an optimal solution a prospective business [Tietz 1993, Friedrich 1996]. Many insurance agencies recognize the opportunity and offer some kind of insurance comparison.

There are a number of applications available, either on the Web or single user programs, where customers can obtain a list of possible insurance polices. However, all present systems are too complex, expensive and time consuming to use. Furthermore, in the case of Web based systems, users have to enter personal data (many Web users would rather stay anonymous) and once the results are obtained, systems stop short of offering the contracts for the insurance.

The core of a comparison system is the insurance calculator that applies complex insurance determination algorithms to mach users' profiles with concrete insurance polices. There are just a few companies specialized in that field and their calculators are single user programs meant for the use at professional insurance agencies. There also exist Web-server interfaces for such programs but they all suffer from above mentioned drawbacks.

The major goal of the work described here is to transform such an insurance calculator into a multi-user server, capable of offering services to a range of client's devices. Furthermore, the new client application

should be user-friendly, easy to use and capable of preparing an optimal insurance polices followed by efficient contracting procedure.

2. INSURANCE COMPARISON APPLICATION

The software architecture for solving the stated problem is called Kavka and is a client-server system that uses modern software technology to connect different devices and assist in the decision and contract making process [SAP 2004]. The main server runs powerful car insurance calculator software and uses Internet and GPRS network to offer its services to remote clients. Clients are Kiosk systems equipped with touch screens, mobile phones, PDAs, Web browsers or the call centre. Each client is controlled by light-weighted software that organizes user-friendly processing. Connection to the server can be wireless or fixed. That gives clients free mobility and possibility to be present all around the country. The role of Kavka client is to attract customers and offer simple, anonymous and efficient comparison of the insurance providers. Should a customer decide to proceed, each Kavka offer includes the kTAN (transaction number) which is used for identifying the chosen optimal policy in a final insurance contract.

The finalizing of the insurance contract is done with a EuroAssistant system. Based on the data already given, marked by the kTAN, the personalized data have to be added in order to prepare a formal insurance contract. It is extremely effective mobile system that connects Kavka clients who want to make the contract with the agent office. The system uses electronic pen and special form to write down all personalized details with the customer signature. Then, the handwritten document together with previous Kavka offer, with the car profile, is transferred to the central office using GPRS, i.e. mobile communication. The handwriting is automatically converted to the computer readable form, archiving the signed form for legal purpose.

The most important Kavka client is the Kiosk with touch screen terminals. It is placed near the car registration authorities targeting customers who already need new polices or may decide to change insurance provider, when registering the car. However, due to wireless communication, Kiosk clients can also be mobile, equipped with touch screen notebooks. They may be present at special events (e.g. car auctions, fairs etc.) or any other place where it is feasible to find potential customers. The application controlling touch screen is a user-friendly system that guides non trained customer through a short interview. Once a customer gives a minimal amount of information about vehicle and previous insurance polices the resulting list of possible insurance providers is given, sorted by the price. Should customer decide to take one of the offers, the EuroAssistant (a fully mobile agent) is near by for finalizing the contract. Other Kavka clients that use Web, SMS or call centre, need to visit the office equipped with EuroAsistant, to make the final contract.

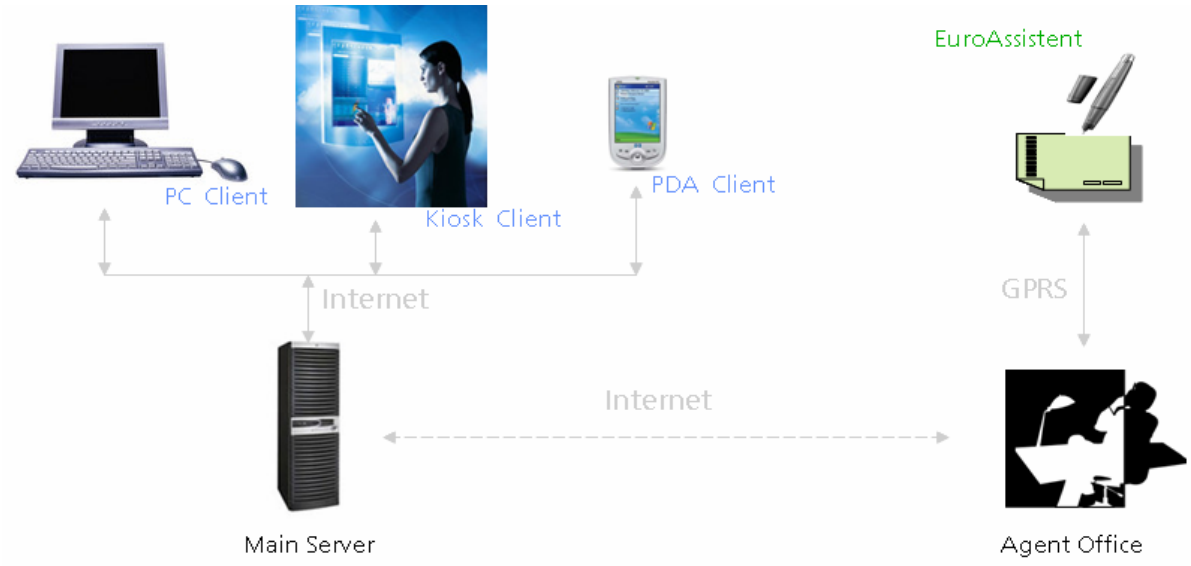


Figure 1. A sample configuration of Kavka system with the kiosk client

The figure 1 illustrates major Kavka components with the different clients and Euroassistent subsystem. The real power of the Kavkai system comes from its versatile mobile clients used for attracting customers and its immediate follow-up contracting EuroAssistant subsystem that performs (finalizes) insurance contract.

3. INSURANCE CALCULATION PROGRAM

The initial software that performs car insurance calculation is a monolith single-user program that has been designed for the office use in insurance agencies. It is programmed in Visual basic and C programming languages, uses a lot of local data structures and has an interactive part that allows for dialog driven insurance calculation.

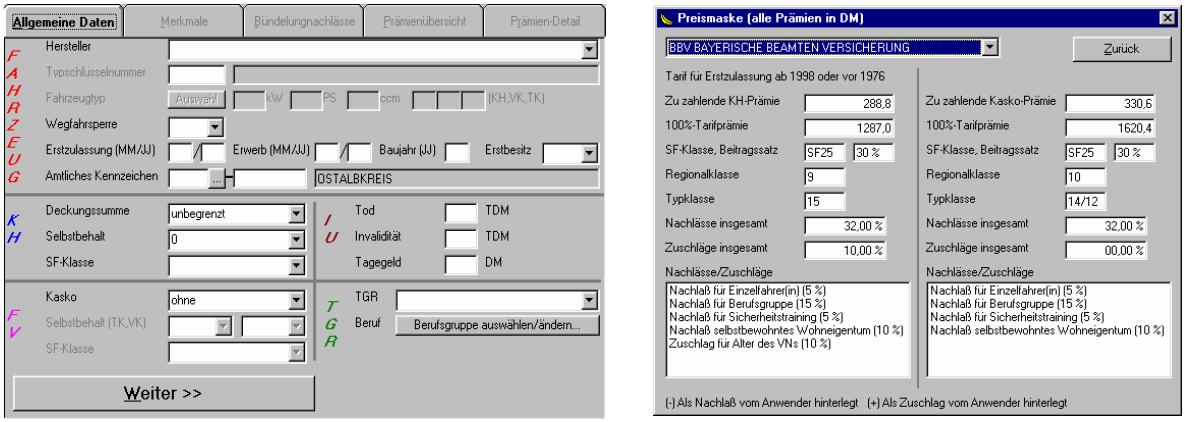


Figure 2. Screen shots of the original insurance calculation program

Figure 2 illustrates two screen shots of the dialogue driven program. It clearly indicates that a complex and tedious interaction takes place before the user may get any results. Furthermore, a monolith software organization is not suitable for a wider use of a program, especially in a multi-client setting. However, this application contains a valuable insurance calculation algorithms and data bases of all relevant insurance and vehicle data. Complete re-programming of the system would be impossible, without thorough knowledge of insurance algorithms, and even then, a weekly maintenance would be needed to keep insurance and vehicle database up to date.

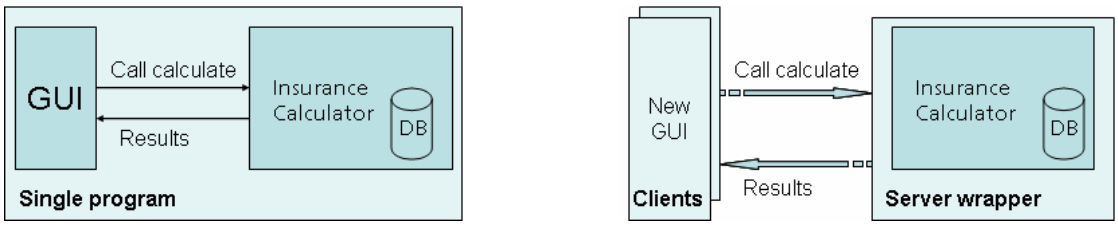
4. SOFTWARE TRANSFORMATION

To provide a new multi-client access to the insurance calculator, an original program has been transformed into a client-server system. The transition has been done in a transparent manner such that functionality of the old code remained unchanged, while opening possibilities for new additions. The user interface (client-side) has been de-coupled from the calculator (server-side) and communication between the two parts has been realized by (remote) procedure calls. The new client-server architecture brings numerous benefits:

- Client program is completely independent from the main server and may be coded in different programming languages and may run on different devices
- Server program retains the existing functionality and can be centrally maintained
- New functionalities can be added both at client and at a server side, providing full scalability of the new system.

With a new architecture further system extensions and coupling with other applications are made possible. For example, connecting to custom database servers for further processing or connecting to the insurance and/or governmental servers is now fully transparent and can be done without affecting the

functioning of the running system. That makes the new system fully open for new business models and new types of application.



a) Initial insurance calculation program

b) New client-server architecture

Figure 3. Program transformation

Figure 3 illustrates the transformation being done to the initial code. At the GUI side, completely new software is developed. At the server side, a program wrapper is developed simulating "old GUI" calls such that the rest of the code may remain unchanged (not being "aware" of the transition being made).

The software architecture for solving the problem of transformation from single to multi user software is called Kavka and is a client-server system that uses Web services and SOAP protocol for client-server communication. The overall system architecture follows the rationale of multi-tier distributed systems: the client is placed at the up most tier and is completely (both, physically and logically) separated from the rest of the system; the server is further divided into a number of layers that facilitates network processing and main program functionality.

The core of the problem in software transformation is to design and develop server wrapper to the initial application that would allow for distributed communication with the new client programs, leaving the old program in an "illusion" that it communicates with its own original user interface.

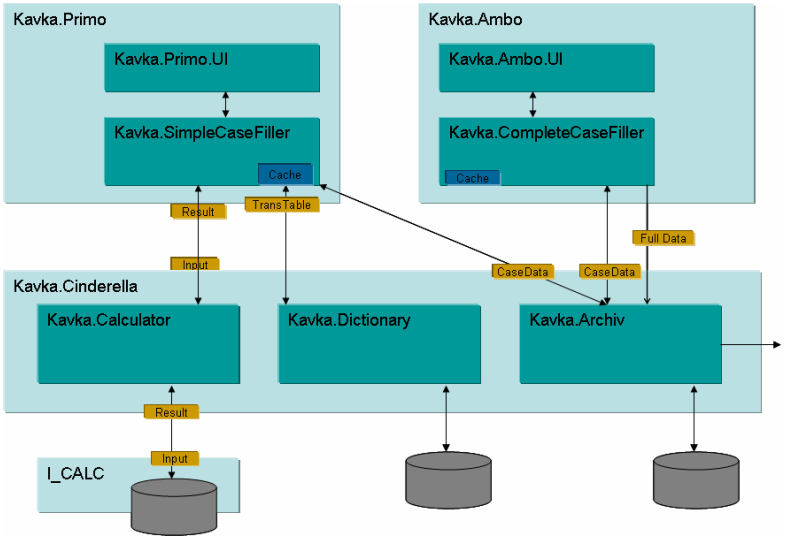


Figure 4. More Detailed Kavka Architecture

The Figure 5 illustrates detailed functioning of the new enhanced system. The lowest level I_CALC program is unchanged: it simply receives user/car profile, calculates the costs and gives the list of insurance prices. The next layer, Kavka.Cinderella, is the actual server wrapper. It receives three types of requests from the clients: (1) request for the I_CALC calculation, that is proceeded to the I_CALC calculator; (2) request for the car description that forward the call to the Kavka.Dictionary which, again uses the old code to check the type of the car in the database and either accept the car code, or warn the user that the car does not exist.; (3) request to allocate the kTAN and store the user profile and car insurance offer in a database for later

processing (contracting). The communication between I_CALC and Kavka.Cindarella is by local (native) procedure calls (I_CALC is coded in visual basic and C, Kavka Cindarella in C#). The lowest level arrows on the figure illustrate unchanged old code. Kavka.Primo is is a touch-screen client, which runs remotely at the insurance kiosk. It communicates with Kavka.Cindarella by issuing remote procedure calls, again having the same three request types: (1) for the I_CALC calculation, (2) for the car description and (3) to store the user profile and car insurance offer in a database. Kavka.Ambo is also a client side program. It has the same functionality as Kavka.Primo, but slightly different GUI. This client runs at the insurance agent office and offers a more detailed user description. At the level of Kavka.Primo and Kavka.Ambo, new clients may be included, offering the same functionality for new types of clients (PDAs, smart phones, call centre etc).

5. CONCLUSION

This paper discusses the rationale for upgrading existing software applying internet-based technology. A report from the practice describes the experience in the domain of automotive and financial software focusing on concrete steps in the process of software transformation. The advantages obtained are seen in several aspects: system usability and business model improvement, extension of the functionality of existing system while preserving original usability, inclusions of smart clients and system scalability.

A possibility to dynamically configure the system provides means to deploy different business strategies. Economical improvements are achieved by applying expertise of individual insurance agency that can configure the system in a way suitable to their business model. For example, instead of interviewing a user with 80 questions, an agent may set more than 60% of questions to default values and reduce the dialogue to just 20 questions, making the procedure user-friendly and efficient. Furthermore, individual discounts can be built-in allowing for a customizable insurance policy offers.

The future work within this application domain is to extend the system to allow insurance contracting of other motor vehicles like motorcycles, tracks and busses. To achieve that, both client and server side have to be extended with new functionality, though the present system will remain intact. In the further phases also other insurances will be included, from households up to a variety of other insurance sectors. Another possible extension is the data exchange among insurance companies and authorities for vehicle registration. This step, however, goes in direction e-government application domain and requires consensus on data structure and automatic information exchange among different institutions.

ACKNOWLEDGEMENT

The authors are grateful to their customer and partner who has supported this work and provided the resources and valuable information about car insurance market and insurance calculation details.

REFERENCES

Schramm-Klein, Hanna, 2002. *Multi Channel Retailing*, Gabler Edition, Berlin.

Degel, Joachim, 2003. *Erfolgsfaktoren im Multikanalvertrieb*,
http://www.z-funke.de/lunch/images/vortrag_degel.pdf

BV, 2004. Bund der Versicherten, *Kraftfahrzeugversicherungen,*
http://www.bundderversicherten.de/bdv/Merkblaetter/KFZ.pdf,

GDV, 2004. Gesamtverbands der Deutschen Versicherungswirtschaft, *Statistisches Taschenbuch*
http://www.gdv.de/download/

Tietz, Bruno, 1993. *Der Handelsbetrieb*, Vahlen, München.

Friedrich, Wolfgang, 1996. *Auto-Pro*., Heyne Verlag, München

SAP, 2004. Sap Design Guild, *Interaction Design Guide for Touch screen Applications*,
http://www.sapdesignguild.org/resources/TSDesignGL/TSDesignGL.pdf