Customer Relationship Management for SMEs

Hubert BAUMEISTER*
Institut für Informatik, LMU, Oettingenstr. 67, D-80538 München, Germany
Tel: +49 89 2180 9375; Fax: +49 89 2180 9175
Email: baumeist@informatik.uni-muenchen.de

Abstract. Customer Relationship Management (CRM) is getting more and more a key strategy for companies big and small. Customer care strategy and CRM software go hand in hand. In particular SME's need a CRM software that easily adapts to their customer care needs while still being low cost. In this paper I discuss the benefits of CRM for SME's and their special requirements wrt. CRM software. Further, I introduce the IST-project CARUSO whose objective is to provide SME's with a software framework to implement low cost, customized CRM applications. This paper presents the rational behind the CARUSO project, the architecture of the framework, and the software development process used to build the framework.

1 Introduction

To survive in the global market, focusing on the customer is becoming a key factor for companies big and small. It is known that it takes up to five times more money to aquire a new customer than to get an existing customer to make a new purchase [12]. Therefore, customer retention is in particular important to SME's because of their limited resources. A second aspect of CRM is that knowing the customer and his/her problems allows to acquire new customers more easily and facilitates targeted cross-selling.

But how does one maintain a good customer relationship? Only for very small companies, i.e. less than ten employees, it is possible to know the customer personally and know the products he bought, his/her preferences and problems. For larger companies, to maintain the same impression, a "common memory", i.e. software support, is needed.

It is important to note that establishing and managing a good customer relationship is a strategic endeavor. Having a CRM software installed by itself does not ensure a successful customer relationship. For this to happen business processes and company culture have to be redesigned to focus on the customer. CRM software can be only a tool to implement a customer strategy.

CRM software can be broadly divided into two categories (cf. [10]). The first category manages customer interaction and contains applications for customer service and support, sales force and marketing automation. These applications allow the company to establish a personalized relationship with the customer. The second software category analyzes the behavior of the customer and contains reporting and data-mining tools. This allows to measure customer satisfaction and retention and enhances the understanding of the customer's problems and preferences. This, in turn, allows a more targeted marketing and sales strategy.

SME's need a low cost CRM solution that adapts to their business model and IT structure instead of having to adapt their business model and IT structure to the CRM software. In addition, it should be possible to introduce the software in little steps to not interrupt the

* appeared in proceedings E2002, Prague, 16–18.10.2002

current business. Very likely the business processes will have to react to changes in the future; thus the software should be easily adaptable to these future changes once installed.

Currently available CRM software is either targeted to large enterprises and provide big, inflexible, and difficult to implement all-in-one solutions. Or CRM software is targeted to SME's supporting only part of the necessary functionality. For example, they may have Unified Messaging and Interactive Voice Response, but fail to provide intelligent call and message routing.

The objective of the IST-project CARUSO is to provide SME's with a framework for creating low cost, customized, and integrated CRM applications. The CARUSO framework provides a set of components that can be configured and plugged together by tools with only little programming effort. This framework can be easily instantiated to the needs of an SME by following the provided guidelines and best practices for building CRM applications. Instantiations of the framework for different market segments, like utility, banking, and service, exists and can be easily adapted further to other companies in these segments.

The next section describes requirements software supporting CRM strategies should satisfy. Section 3 gives background information on the CARUSO project and describes the CARUSO framework, its components, tools, and methodology. Section 4 gives an overview over the software development process used in the CARUSO project and finally Sec. 5 presents a conclusion.

2 Requirements on CRM Software for SME's

Introducing IT support for CRM, in particular for SME's, requires solutions that adapt to the business model of the company. Most likely, it is not feasible for a company to change their whole IT infrastructure for their CRM business, therefore the CRM software has to easily integrate in the existing IT environment of the company. To reduce costs, the software should be easily configurable by the company itself and not require expensive consultancy to adapt the software to the needs of the company. Business processes are not stable and need to be adapted when the business of the company changes. A CRM software should be adaptable to such changes. To reduce costs and the risk of introducing customer relationship management at a company, a CRM solution should be established not in one big step but in several small steps in a company. This, of course, requires software that has to integrate into the current IT infrastructure and that can be extended in a modular fashion.

Recently, the Gartner Group reports that about 55% of all CRM projects fail, and Bain & company note that of these failures 20% lead to damage to long-standing relationships (cf. [11]). Six reasons for these failures were identified. In addition to missing top-management support and trying to disguise efficiency improvements as CRM, but without really caring about the customer, the most important reasons for failures are unrealistic expectations, missing customer care strategy, and neglecting metrics. Unrealistic expectations are created by aggressive marketing of CRM software, and quite often customers fail to define their needs and therefore buy software whose features are defined by the vendor and not by their own needs. Another reason for failure is that the company has no customer strategy. For example, is the goal of CRM to reduce the service-costs, or to aquire new customers? But even if a customer care strategy is defined, it is important to install a feedback loop to improve the customer support. Another common mistake is not collecting metrics necessary to install this feedback loop, or to use the wrong metrics not aligned with the customer strategy.

Therefore, in addition to providing a software tool, a methodology for selecting the right CRM components and implementing them is needed. Ideally this methodology is provided

by the vendor of the CRM software itself as he is the most experienced implementing CRM solutions.

3 The CARUSO Project

The CARUSO (**C**ustomer C**a**re and **R**elation**s**hip S**u**pport **O**ffice) project [5, 2] is a research and technological development (RTD) project funded by the European Union within the Information Society Technologies (IST) program of the 5th framework program [7]. Partners are REMU, a Dutch utility provider in Utrecht [13], DataCall, a German software house in Munich [6], and the Institute of Computer Science of the Ludwig-Maximilians-University in Munich [9]. The project started January 2000 and has a duration of 30 month.

To achieve the requirements outlined in Section 2, the CARUSO framework to build customized CRM front-office solutions was designed. Instead of providing one monolithic solution, the framework provides a set of basic components. These components can be plugged together and configured by a set of tools to build front-office applications that are indented to act as shell around the company (cf. Fig. 1). Ideally, each communication with the customer,

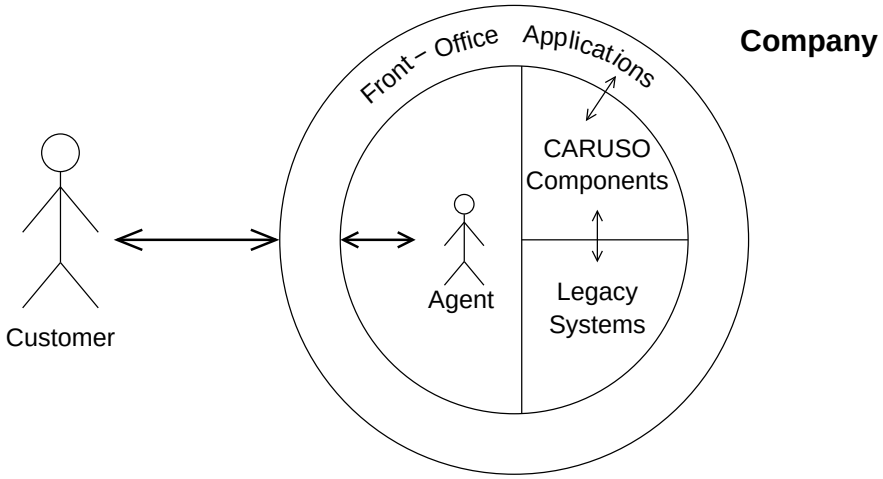

Fig. 1. The CARUSO Shell

either initiated by the customer or the company, is passed through this shell and handled by CARUSO components. In addition, there is a set of tools to monitor the current status of the CRM software and to produce customized reports. Finally, the CARUSO framework consists of a methodology defining steps need to be done to implement a CRM solution at a customer.

3.1 CARUSO Components and Tools

The basic architecture of CARUSO components is shown in Fig. 2. The communication server is the heart of CARUSO. Ideally, each communication between customer and company is handled by the communication server. Currently the communication server handles inbound telephone calls using Automatic Call Distribution (ACD), Interactive Voice Response (IVR), and VoiceMail; and supports outbound calls including marketing campaigns. In the near future, support for other channels of communication, like e-mail, fax, Web, etc. will be added using a unified messaging approach.

The main advantage of the communication server is that it is light-weight, i.e. can be used in a call-center starting with only four agents and with an ISDN connection, but still scales

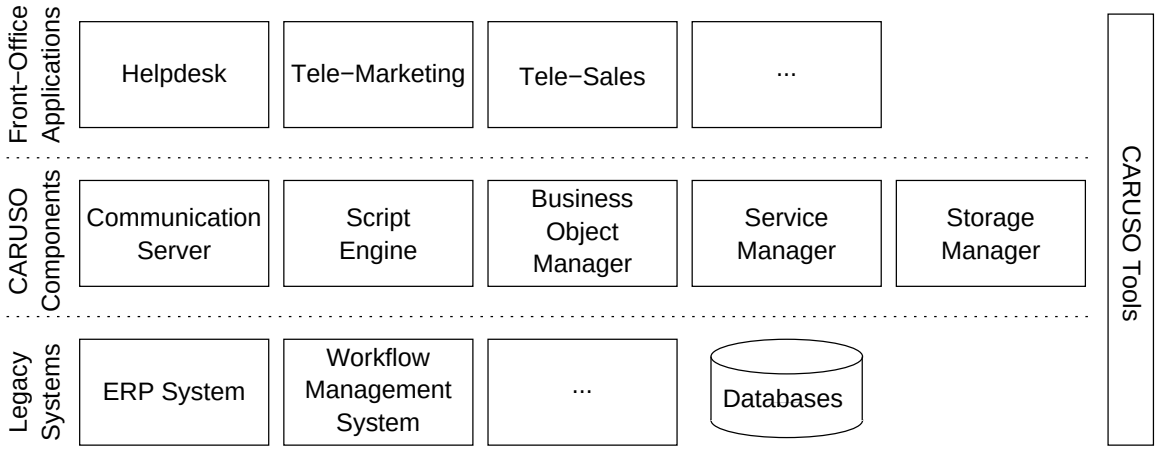

Fig. 2. The CARUSO Architecture

well with up to 180 agents and possible connection to existing PBX's. Another advantage is its intelligent call routing functionality. To have a call distributed to an agent, the agent need not be present at a particular physical location; the call can be routed to his home phone anywhere in the world. The only requirement is that the agent has two phone lines, one for the voice connection and the other for the Internet connection to run the front-office application.

To facilitate and standardize the communication with the customer for common problems, dialog scripts are used. Such scripts consists of predefined questions and information items that are shown to the call-center agent who presents them to the customer and makes notes of the customer's answers in case of a question. The sequence of questions and information item depends on the answers given previously by the customer and the information about the customer in the databases. In addition, arbitrary actions can be associated to transitions between subsequent questions and informations, like sending an e-mail, updating the ERP system, etc. The script engine is used to execute these dialog scripts and, in addition, allows to run these scripts on the Web for self-service, that is without the presence of an agent.

Dialog scripts can be defined by the company who is using CARUSO with the help of the script developer tool. In addition, dialog-scripts are stored in XML, and thus can be read and modified using standard text editor if needed. The DTD (document type definition) for dialog scripts and a Java implementation of the script engine will be made available in the public domain.

Business objects are an object-oriented view on the data occurring in CRM processes. They are designed with the business object modeler by mapping class names and attributes to tables and columns in relational databases. The way business objects are presented to the user is defined by the application builder tool.

The storage manager abstracts away from the details of particular implementations of relational databases. Using this abstraction layer, it is easy to change the underlying database system or to transfer and extend front-office applications from the same vertical market built for other companies. This is in particular interesting since, for example, applications in the utility market will have similar business object models.

A very important component of the CARUSO framework, however, not visible to the outside, is the service manger. The service manager manages CARUSO components and components that connect to other CRM systems or legacy components. Each of the CARUSO component represents a service that registers itself with the service manager. Each front-office application that needs a certain component asks the service manager for that service. Therefore the front-office application does not need to know where components are to be

found; the only component whose location the front-office application has to now is the service manager. In addition, it is easy to add new components and to replace unused components by new versions of that component at run-time.

In addition to CARUSO components, the service manager can also manage other components that implement the service interface. This is the core of the extensibility of CARUSO. If access to a legacy system or to any other IT system in the company is needed, the only task is to implement a service that acts as a wrapper around the other component.

3.2 Monitoring- and Reporting Tools

The CARUSO framework includes tools to monitor the current status of several customer care parameters. These are updated permanently. The status monitor shows actual information on the status of agents, i.e. the number of registered agents and their status (ready to receive calls or busy); the list of waiting calls; the campaign status, i.e. the number of (active) campaigns; and other relevant information.

In contrast, the reporting tools are used to generate statistic reports from past information. These reports can be configured; typical examples include reports on agent availability, incoming calls, and voice mail. For example, the reports on incoming calls includes statistics of waiting, transferred, abandoned calls per defined period, average time connected, and others.

3.3 CARUSO Methodology

The CARUSO methodology is intended to guide the process of building a particular set of front-office applications from the components and the tools provided by the framework.

1. The customer strategy, the business processes, and the requirements of the company implementing the the CRM solution are defined. This addresses several of the mistakes mentioned in [11] (cf. Sec. 2).
2. The business rules and the participating business objects are identified and modeled with the help of the business object modeler.
3. Identification of required interfaces to legacy system. Simple interfaces, like accessing relational databases, are already provided by the storage manager component and are configured with the business object modeler. Complex interfaces may need additional programming effort.
4. Creation of appropriate dialog-scripts using the script developer.
5. The user-interfaces for screens and views are defined using the application builder. This includes the reporting and monitoring views displaying metrics according to the customer strategy as defined in step one.
6. The system is implemented step-wise with intermediate tests to check if the requirements are satisfied.
7. Training of the agents with the software.
8. Introduction of administrator into the change procedures.

4 The Software Development Process

We have used practices from Extreme Programming (XP) [4] in the development of the CARUSO framework. The major problem with designing such a framework is finding the right components and their functionality because the requirements on CRM software are quite complex as they involve all the business processes of a company, like marketing, sales,

service, etc., and all its IT systems. A classical software development process would have required us to analyze a good deal of these processes before starting the design of the system [14]. Because of the complexities involved in CRM, this proved to be impossible.

In contrast, XP is a lightweight process which incorporates methods to react to change (coming either from changes in the business or from unclear requirements) while not sacrificing the quality of the resulting software. XP is most suited to small and middle-sized projects where the software has to adapt to changes in the requirements and the environment, and where the software needs to produce business value even if not all functionality is implemented.

XP consists of a set of practices, each of them a practice also common to other software processes, but taken to the extreme. For example, testing is well-known also with other processes; however, taken to the extreme, testing means in XP that tests are automatic and written even before the code that they test is implemented. Another XP practice is pair programming which puts code-review to the extreme. In XP each line of production code is produced by a pair of programmers sitting in front of the same computer; while one programmer is writing the code, the other immediately reviews that code. Note that each of these practices may have negative effects when done in isolation; however, taken altogether the positive effects of the practices cancel the negative effects of other practices. For example, the XP practice simple design requires that the software always represents the simplest design necessary to achieve its functionality. When functionality is added, it may happen that to keep the simplest design the software needs to be restructured. This, however, may break already existing functionality. To ensure that this does not happen, automatic tests, to test the existing functionality, are needed (cf. [4, 8]).

In the CARUSO project we started with a rough idea of the CARUSO architecture and defined user-stories based on the CRM needs of REMU. During the implementation of these user-stories, the components of the framework and their functionality were discovered and implemented. More information on this topic can be found in [3, 1].

5 Conclusion

The CARUSO framework provides a powerful tool to create low cost and easy to use customer care solutions that adapt to the companies business model and are easy to integrate in existing IT infrastructures. Using XP as the software development process helped a lot to clarify unclear requirements and to keep the project on track.

The main goals of CRM are improved customer retention and easier acquisition of new customers. CARUSO improves customer retention by providing a framework intended to act like a shell around the company so that each interaction with the customer can be recorded. This allows to maintain a consistent view on his/her problems and preferences regardless of the channel he/she is using.

The CARUSO components are easy to configure using the CARUSO tools. This allows to adapt CARUSO to the company and not the other way round. Further, this reduces costs for consultancy, as most of these configurations can be done by the company itself. In addition, this allows to adapt to changes in the CRM strategy forced by changes in the business and from experiences gained with using the CRM software. Experience with CARUSO has shown that a 2.5 Cent investment on each call results in a 23 Cent increase in productivity.

CARUSO is intended to be used starting from as low a four agents. Therefore, it is suitable for SME's and Micro-SME's. Further, the communication server does not restrict the physical location of agents. Agents can be at any place in the company and even be working

at home and still being reachable through a single access point, e.g. the support number of the company. This is in particular useful for Micro-SME's, as usually their employees assume different roles as needed. For example, the person that does the marketing may be also responsible for sales and support. On the other hand, the CARUSO communication server scales well up to 180 agents to cope with the increasing needs of SME's.

The CARUSO methodology ensures that the introduction of the CRM framework is in accordance with the business strategy of the company and that the benefits and costs for introducing CRM into a company are accounted for. The introduction of CRM requires changes in the culture and the business processes of a company regardless which CRM software is used. Only with business processes and culture adapted to the CRM strategy, a company will benefit from using CRM software. The CARUSO philosophy is to have the changes in business processes and culture happen in small steps. The CARUSO monitoring and reporting tools are used to quantify the effect of these steps. This helps to establish and gain experiences with new practices while not risking the current business.

Future work on CARUSO is going to apply the framework to different vertical markets to gain more experience with the CARUSO methodology and to improve the cross-fertilization from different instantiations of the framework at different customers. Already several extensions of and applications for the CARUSO framework are planned:

- A care center application for general customer service providers
- A care center application for banks
- A technical care center application for industry and distribution
- A citizen and tourist care center application for regions and islands
- An investor relation care center application for share-based companies

Acknowledgments

This work has been partially sponsored by the European Union within the IST project CARUSO (IST-1999-12192).

References

[1] Hubert Baumeister. Using XP to develop a CRM framework. In *XP2002. 3rd International Conference on extreme Programming and Flexible Processes in Software Engineering*, 2002. www.xp2002.org.

[2] Hubert Baumeister and Piotr Kosiuczenko. CARUSO: Customer Care and Relationship Support Office. In Ricardo Gonçalves and Adolfo Steiger-Garção, editors, *ECPPM, Lisbon, Portugal, 25-27 September 2000*, pages 115–120. A. A. Balkema, Rotterdam, Brookfield, 2000.

[3] Hubert Baumeister and Martin Wirsing. Applying test-first programming and iterative development in building an E-business application. In *International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet, SSGRR 2002, L'Aquila, Italy*, 2002.

[4] Kent Beck. *Extreme Programming Explained*. Addison Wesley Longman, 1999.

[5] CARUSO Consortium. CARUSO Web site. www.caruso24.com, 2002.

[6] DataCall Systeme. www.datacall.biz, 2002.

[7] European Commission. CORDIS: IST: home page. www.cordis.lu/ist/, 2002.

[8] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison Wesley, 1999.

[9] Institute for Computer Science, LMU. www.pst.informatik.uni-muenchen.de, 2002.

[10] Ronni T. Marshak. Defining CRM. www.crm2day.com, 2002.

[11] Adrian Mello. Six mistakes that will sink your CRM. www.zdnet.com, March 18, 2002.

[12] Adrian Payne. Customer Relationship Management. www.crm2day.com, 2002.

[13] REMU. www.remu.nl, 2002.

[14] Ian Sommerville. *Software Engineering*. Addison Wesley Longman, 2000.