

This Technology Guide is one in a series of topic-focused Guides that provide a comprehensive examination of important and emerging technologies.

This series of Guides offers objective information and practical guidance on technologies related to Communications & Networking, the Internet, Computer Telephony, Document Management, Data Warehousing, and Enterprise Computing.

Built upon the extensive experience and ongoing research of our writers and editorial team, these Technology Guides assist IT professionals in making informed decisions about all aspects of technology development and strategic deployment.

techguide.com is supported by a consortium of leading technology providers. IBM has lent its support to produce this Guide.

Visit our Web Site at
www.techguide.com
to view and print this Guide, as well as all of our other Technology Guides.

This is a free service.



The Technology Guide Series™
techguide.com™

Building a Customer-Centric Contact Center



This Guide has been sponsored by



Table of Contents

Introduction	5
The Customer Contact Environment	7
Key Integration Requirements	11
Enterprise Application Integration Technology	15
Right Price Insurance – An Example	21
Business Benefits of an Integrated Contact Center Strategy	28
Conclusion	32
Glossary of Terms.	34

About the Editor...

Jerry Ryan is a principal at ATG and the Editor-in-Chief of techguide.com. He is the author of numerous technology papers on various aspects of networking. Mr. Ryan has developed and taught many courses in network analysis and design for carriers, government agencies and private industry. He has provided consulting support in the area of WAN and LAN network design, negotiation with carriers for contract pricing and services, technology acquisition, customized software development for network administration, billing and auditing of telecommunication expenses, project management, and RFP generation. Mr. Ryan has been a member of the Network+Interop Program Committee and the ComNet steering Committee. He holds a B.S. degree in electrical engineering.

The Guide format and main text of this Guide are the property of The Applied Technologies Group, Inc. and is made available upon these terms and conditions. The Applied Technologies Group reserves all rights herein. Reproduction in whole or in part of the main text is only permitted with the written consent of The Applied Technologies Group. The main text shall be treated at all times as a proprietary document for internal use only. The main text may not be duplicated in any way, except in the form of brief excerpts or quotations for the purpose of review. In addition, the information contained herein may not be duplicated in other books, databases or any other medium. Making copies of this Guide, or any portion for any purpose other than your own, is a violation of United States Copyright Laws. The information contained in this Guide is believed to be reliable but cannot be guaranteed to be complete or correct. Any case studies or glossaries contained in this Guide or any Guide are excluded from this copyright.

Copyright © 1999 by The Applied Technologies Group, Inc. One Apple Hill, Suite 216, Natick, MA 01760, Tel: (508) 651-1155, Fax: (508) 651-1171
E-mail: info@techguide.com Web Site: <http://www.techguide.com>

Visit ATG's Web Site
to read, download, and print
all the Technology Guides
in this series.

<http://www.techguide.com>

“The significant problems we face cannot be solved
by the same level of thinking that created them.”

Albert Einstein

The way a company interacts with its customers is rapidly changing, with one of the goals being to take full advantage of every customer contact. Competitive advantage, in today's networked world, is achieved by knowing your customer better than your competitors and using that knowledge to the best advantage. Even though a company may have adopted a more customer-focused organization, the database and information systems it depends on are usually still built around either the product or the business process, not the customer. This has become an important constraint, especially with the emergence of integrated contact centers, because easy access to all required customer information is the only way to ensure "contact optimization." Given that re-development of existing business systems cannot and will not happen overnight, there is a gap between the tools that are needed and those that are currently available. The next generation of customer contact systems will be customer-centric and integrated at all levels. They will also provide complete access to older product-centric applications, offering for the first time a complete customer contact environment. This Technology Guide examines the requirements for a successful customer-centric contact center and identifies the real-world issues associated with enterprise-wide management of the customer relationship. The Guide discusses how you can build an advanced contact center while also avoiding wholesale replacement of existing IT systems.

Introduction

Today, most companies have to work hard to find, develop, and then keep their customers. Any interaction that turns out to be frustrating or unproductive could easily become the customer's last contact with the company. In fact, any interaction that does not impress the customer and expand the company's knowledge base has to be viewed as a wasted opportunity. Many companies now recognize the need for a stronger focus on the customer at any and every "point of contact." Companies are beginning to develop and implement strategies for collecting, organizing, managing, and using customer-related information. *The bottom line is that competitive advantage, in today's fast-paced, networked world, can only be achieved by exploiting customer knowledge better than your competitors.*

What does any customer look for when choosing among competitive suppliers? The product is no longer the only, or even the key, differentiator among companies. The primary distinguishing factors are most often the sales channel, the price, and the quality of customer service. Many company executives are asking themselves an important question: If we cannot count on our products to provide an inherent competitive advantage, then what do we use as the basis for generating sales and for winning customer loyalty (other than lowering the price)? One of the answers is to improve the effectiveness of the customer interaction: contact must be more individualized for the customer, the right tools must be available to the company's representatives, and all possible customer information has to be collected.

In a highly competitive environment, the emphasis must be on retaining existing customers and growing only the most profitable new ones. It has been estimated that it can be up to ten times more expensive to get new customers than it is to retain existing ones.

Needless to say, only those companies that are customer-centric can differentiate their customers based on their value to the company. This type of business analysis cannot be left as an ad hoc, informal technique; rather, it needs to be a decision-making process that is based on extensive knowledge and understanding.

The acronym “CRM” (customer relationship management) refers to this new customer-centric approach. It includes the establishment, cultivation, and maintenance of long-lasting and mutually beneficial relationships with all of the company’s customers. Large organizations such as banks, insurance companies, telecommunications providers, and utilities must learn to treat the customer as an individual, regardless of the method of contact. Marketing experts would agree that the one-to-one approach to the customer is generally the most successful.

The “contact center” can be viewed as the next generation of call center. Contact centers are multi-access and multi-service. They cover a wide range of channels including telephone, Internet, letters, e-mail, etc., thereby going considerably beyond the traditional single function, telephone-based call center. In addition, the contact center serves as a single point of contact for various types of customer interaction: sales, marketing, product support, information distribution, and any other service the customer may need.

The traditional retail point of sale is now just one of many important “points of contact” for a company. Mobile sales forces, call centers, the Web, direct marketing, and the telephone (i.e., any employee) all contribute opportunities for gathering information about the customer and what they want. The key to success is the organization and consolidation of all the data that is generated through these various contact channels.

This Technology Guide discusses the role that a contact center plays, defines requirements for contact

center systems and describes how you can build a customer-centric contact center using the latest generation of enterprise-wide CRM systems. It examines the importance of integrated information access, a consistent agent interface and real-time systems operations. The proposed solution is to add customer-centric applications at the front-end while also recognizing the need to avoid wholesale replacement of back-end business applications. The benefits to the company, as it will be shown, are increased revenues, lower costs and higher quality contacts with the customer.

The Customer Contact Environment

Customer contact is clearly critical to the success of any business – if you don’t interact with your customer, you won’t make any sales. Simply being able to complete a basic business interaction (such as taking an order) is no longer adequate, however. The quality and completeness of the customer’s contact experience has become a primary differentiator for any company that exists in a competitive marketplace. The customer contact infrastructure encompasses all of the systems, information, and processes that are needed to assure the complete success of the customer interaction.

The contact experience can be significantly improved when the company’s sales, marketing, and service personnel — in fact, anyone who is “customer facing” — can all access the same information about customer preferences, contact occurrences, and overall relationship history. This information must be immediately available whenever and wherever it is needed, and it must be accurate and complete at all times.

The following key requirements need to be met for the customer contact environment to be a success:

- **Focus on the Needs of the Customer**
The environment must provide a single point of contact for the customer for all products and services, whether for customer service or sales. The environment must help enforce business processes so any issue can be dealt with correctly the first time. It must use all the data that is known about the customer so that the customer does not have to continually provide the same information. Most importantly, it should be the customer who decides how to interact with the organization and who chooses both the time of day and the mechanism for that interaction. Lastly, those crucial, highly valued customers must be treated with the special degree of service they deserve.
- **Increase the Organization's Knowledge of the Customer**
Every customer interaction must be remembered. More than that, the environment should facilitate dialogue so that the organization is able to build a wealth of valuable information about the customer.
- **Exploit Customer Knowledge**
The organization's knowledge of the customer must be used intelligently at every interaction. (See Figure 1). Every contact should be seen as a possible sales opportunity, not as a random event, offering the most appropriate product at the most appropriate price. Knowledge should also be used to enable every function within the organization — for example, to allow sales to have an understanding of any outstanding customer care issues before engaging with a customer.
- **Provide an Efficient Environment**
In today's highly competitive world, cost is always a major driver. As a result, this new customer-

centric environment needs to be highly efficient. It should maximize the use of personnel and supplement this with appropriate use of new technology.

- **Improve Flexibility of the Environment**
Moving forward, the environment's flexibility is going to be key. The environment should support the rapid introduction of new product systems to reduce the time to market and to allow the organization to exploit new channels, such as the Internet.

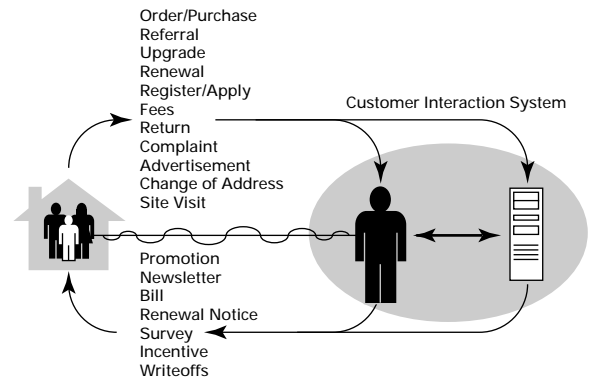


Figure 1 – Customer Interaction Examples

The latest generation of CRM applications, such as Siebel or Oracle CRM, seems to offer the solution. However, companies trying to implement these systems face a huge IT challenge. Their existing systems were designed for product-related processes, not customer-related ones. Their customer data is scattered across a range of stove-piped product systems with no ability to link that data together. It's seldom practical to scrap an entire, existing information infrastructure and adopt a completely new one. So, businesses must find ways to adapt their current infrastructures and integrate new, customer-related system components.

A related Technology Guide “Building a Successful CRM Environment”, describes a blueprint

for solving this problem. It consists of a Data Warehouse that consolidates data from all these multiple systems into a single customer-centric data store, where it can be analyzed by a range of sophisticated tools to give organizations deep insight into customer behavior. It also includes a platform for the latest generation of customer contact applications, enabling the applications to access this knowledge to support customer-specific scripting and seamlessly integrating the applications with the underlying product and functional systems in real-time. (See figure 2).

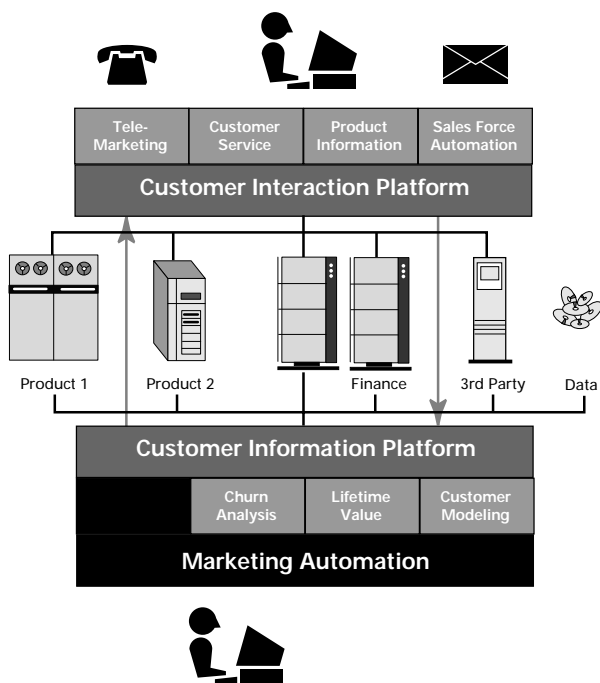


Figure 2 – An Integrated, Customer-Centric CRM Architecture

However, the complexity of integrating a CRM application into this framework should not be underestimated. The rest of this guide describes this technical integration challenge in more detail.

Key Integration Requirements

Specific capabilities are needed for the successful integration of a CRM application with the organization's existing IT infrastructure and Data Warehouse:

a) Single User Interface

For customer-facing staff to be productive, a single user interface is required that provides access to all the systems needed for effectively servicing the customer. A solution that gives the user access to multiple systems is slow, error prone, and very expensive to operate. Customer-facing staff should only need to access the CRM application suite to service the customer.

Requirement 1: The integration solution should be transparent to the end-user. Any data or functionality accessed external to the CRM application should still be delivered through the user interface.

b) Sophisticated Customer Management Support

Sophisticated customer interaction management functionality is required both initially and in the future as the customer interaction strategy evolves. This functionality may include scripting, workflow, integration with telephony equipment, scheduling of callbacks, literature distribution, integration with the Internet or other new channels, etc. It is not possible to determine the definitive requirements for customer interaction management in advance.

Therefore, all features of both current and future versions of the CRM application need to be supported.

Requirement 2: The integration solution should be transparent to the CRM application. Any functionality provided by the CRM application should work as intended even if it involves access to data or functionality that is external to the CRM environment.

c) Performance

The customer interaction environment needs to run at conversational speeds, and yet typically the existing systems are not designed for this type of use. For example, a call center agent may need to go through multiple screens to get to the appropriate piece of information, or a transaction may span multiple systems. Even if individually the systems are responsive, the overall response time is unacceptable if processed sequentially in this manner.

Requirement 3: The integration solution must enable the CRM application to run at conversational speed even if external access to data or functionality that it requires is substantially slower.

d) Availability

The customer interaction environment needs to be available 24 hours a day, 7 days a week in order to meet the access demands of an increasingly demanding public. However, the existing systems are frequently not designed to support this. They require significant downtime for backups, etc. or have large batch windows when the data is available read only.

Requirement 4: The integration solution needs to enable continuous operation (i.e., application availability on a 24 hour/day, 7 day/week basis) of

the CRM applications, even if some or all of the external systems it accesses are unavailable.

e) Exploit Back-End Systems in Real-Time

The customer interaction environment may need to access both functionality and data from the existing systems in real time. Such access operations may cause an external event to occur such as provisioning of a mobile phone feature or enabling an increased credit limit on a credit card.

Requirement 5: The integration solution needs to access the functionality and data of the back-office systems in real-time.

f) Exploit Back-Office Systems through Supported Interface

The customer interaction environment needs to access both functionality and data from the existing systems. These systems may only be designed for use through the provided user interface.

Application logic may be resident in the users' screens. Application vendors may not publish the data model for the application, and if they do, frequently the applications will not allow for updates from foreign applications. Furthermore, it may not be possible to automatically detect changes to data on these back-office systems.

Requirement 6: The integration solution needs to access the functionality and data of the back-office systems through a supported interface, even though the only supported interface may be the user interface.

g) Minimal Changes to Back-End Systems

It is frequently not possible to change the existing systems to support the needs of the customer interaction environment. It may simply be prohibitive from a cost perspective or just too far down the

priority list (behind ensuring Y2K readiness, for example) to be practical. The systems may even be external to the organization.

Requirement 7: The integration solution needs to access the required functionality and data on the external systems without requiring changes to these systems.

h) Transaction Integrity

Data integrity needs to be maintained internally within the CRM applications and the external systems and in all data transfers between the CRM applications and the external systems. The external systems may not support the sophisticated two-phase commit protocols that this normally requires. Indeed, they may not even be available when the transaction needs to be committed.

Requirement 8: The integration solution needs to maintain transaction integrity without relying on two-phase commit or other features of the external applications and taking into account the performance and availability constraint.

i) Referential Integrity

Potentially, data can be accessed through two views, either through the CRM environment or by accessing the external systems directly. The view must be consistent regardless of which method is used.

Requirement 9: The integration solution needs to maintain referential integrity between the CRM applications and the external systems it accesses.

j) Access to Customer Knowledge

The customer interaction environment needs to be able to access all the customer knowledge that has been established within the organization. This could include lifetime value, propensity to churn,

propensity to buy information and so on. The CRM applications may require this information at any point during the customer interaction.

Requirement 10: The integration solution needs to provide access to customer knowledge in a format that is readily useable by the CRM applications.

Enterprise Application Integration Technology

A plethora of middleware or Enterprise Application Integration products exist that purport to address this integration problem. Indeed any of these products could be used to build a solution but the effort and associated cost would be radically different depending on the tool or tools selected. Before going on to look at a specific solution, let's look at the range of products available today.

Enterprise Application Integration products can be categorized as follows, although frequently individual product fall into multiple categories:

1. Communication Middleware

Communication middleware products provide program-to-program communication facilities. They typically provide an Application Program Interface (API) that simplifies communication for the programmer.

Communication middleware supports a range of communications patterns including:
(See Figure 3)

- Conversational

The two programs have an extended interactive dialogue.

- Request/Reply
One program issues a 'request' and then waits for a 'reply' from the other program.
- Message passing
Program sends a message and then continues. There is no response from the receiving program.
- Message Queuing (Store and Forward)
Program passes a message to an intermediate message service. Messaging service passes it on to the receiving program when requested. Original ordering of messages is maintained.
- Publish and Subscribe
Program sends information to intermediate service. Intermediate service passes information on to any program that has requested this type of information.

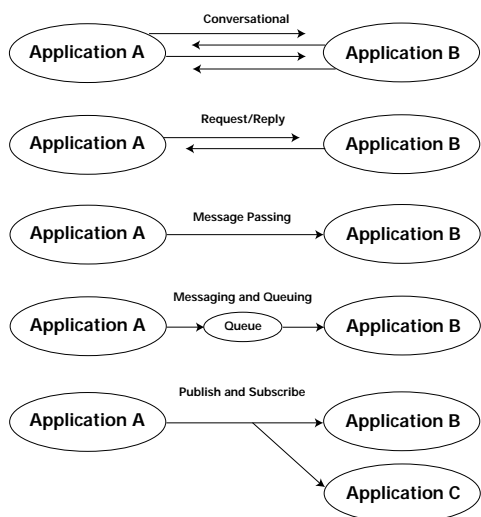


Figure 3 – Example of Application-to-Application Communication Facilities

Communication middleware products include Message Orientated Middleware (MOM) products such as IBM's MQSeries and Remote Procedure Call (RPC) products such as implementations of Open Network Computing (ONC) RPC.

Communication middleware is the underlying technology that supports a range of other Enterprise Application Integration products. It is not an ideal choice on its own to support the integration requirements previously outlined as the amount of coding required would be too great.

2. Platform Middleware

Platform middleware products extend the functionality of communication middleware by providing a variety of run-time services to support the environment and further isolate the programmer from the underlying infrastructure. Services could include name resolution, transaction control, security, load-balancing, etc.

Platform middleware products include Transaction Processing Monitors (TPM) such as BEA's Tuxedo or IBM's CICS and Object Request Brokers (ORB) such as Inprise's Visibroker.

Platform middleware could be used to meet our integration requirement but as we will see there are other products that are built on top of platform middleware that make the task easier.

3. Database Gateways

These products provide synchronous integration at the data level. Database gateways provide transparent access to heterogeneous data sources. This allows an application to access an entity that is integral to another application.

Oracle's Transparent Gateway to DB2 would be an example of a database gateway.

Database gateways are probably not a suitable technology for meeting our integration requirement. Gateways may not even be available for all our back-office systems. Even if they are available as they offer synchronous integration, it is unlikely that we would be able to meet the performance and availability requirements.

4. Extract, Transform, Move, and Load (ETML)

These products provide asynchronous integration at the data level. Their focus is on bulk movement of data.

Examples include ETI Extract and Constellar Hub.

ETML tools may have a role to play in meeting our requirement but they are typically not performant enough to support real-time use.

5. Application or Integration Servers

These products are specifically designed for providing a single, transaction orientated, interface to one or more existing systems.

These products usually have a 'hub and spoke' architecture and have risen in popularity as a mechanism for providing a Web front-end to existing applications. They are all evolving to support Extensible Markup Language (XML), as this becomes a standard.

Examples include Allaire Cold Fusion and Object Design's eXcelon.

Another class of application server is the so-called, "Screen Scraper", which provides programmatic interfaces to legacy applications by including software that automates the key strokes the user types. Computer Network Technology's Enterprise Access is an example of a fully featured screen scraper application server.

Application servers are a possible candidate for solving our integration requirement, although our focus is not to put a new user interface onto existing systems but a complete application. Screen scrapers will definitely have a role to play. Whilst they are notorious as being difficult to maintain, it is often the only supported interface to an applications logic.

6. Integration Frameworks

These are typically product suites that combine a platform middleware product with the ability to transform data and route it according to a set of rules. Many also include pre-built links to the leading ERP packages. These frameworks are specifically designed to offer peer to peer real-time integration between applications and as such are the obvious choice for meeting our integration requirements. They are 'hub and spoke' architecture and rely on the ability of the source application to propagate any changes or 'events' to the hub, which in turn transforms and routes these events to any other applications that require them.

This so called 'push' approach offers a solution to our performance and availability requirements as the data required will be 'pushed' to the CRM application and hence readily available to the user. There are, however, two issues that need to be addressed. Firstly, we cannot rely on the ability of an existing system to detect changes in data and pass them on. This functionality is simply not supported in most legacy applications. It is tempting to think that it is possible to perhaps trap these changes within the transaction monitor or database. These are typically not business transactions but low-level data changes and it is not possible to re-combine these. Secondly, the CRM application is not a 'peer' of the other applications.

The CRM application needs access to all data on the system to support customer enquiries. Consider for example, a telecommunications company. A customer service representative would require access to telephone call records. It is inconceivable that you would implement a solution where every call record was 'pushed' to the CRM application.

Integration Frameworks vary greatly in the depth of functionality provided and the range and quality of any pre-built links to applications. Key vendors in this space include Active Software with ActiveWorks, Crossworlds, IBM with MQSeries Integrator and Workflow, STC with DataGate and Vitria Technology with BusinessWare.

Right Price Insurance – An Example

The best way to demonstrate what is required to build an integrated contact center and to illustrate the complexities and benefits of doing so is through an example. The following is a description of Right Price Insurance (RPI), a fictitious company that is a large, well-established composite insurance provider.

RPI offers a variety of life insurance, home insurance, travel insurance and automotive insurance policies for individuals and groups. These offerings were either developed in-house over a period of years or acquired through the purchase of other companies. RPI has traditionally marketed its products primarily through company sales agents or independent insurance brokers. Although in recent years more and more business is being done directly over the telephone via a centralized call center. Recently RPI has established an Internet site, although because of system limitations it currently only handles provisional quotations for auto insurance. RPI also handles fax requests, emails, and postal inquiries.

The IT infrastructure that supports these offerings reflects the way that RPI was structured with separate environments for each product offering. (See Figure 4). The result is a systems environment that is a mix of application vintages and platform technologies, few of which were designed to the same standards.

In the face of increasing competitive pressure RPI has embarked on an ambitious CRM program with the objective of reducing churn and increasing sales to its existing customer base. RPI has already implemented a Data Warehouse to provide consolidated customer information and support for database marketing activities. RPI has now selected a full featured CRM application to provide a customer

centric environment for sales, marketing, and customer care for all customer contact channels.

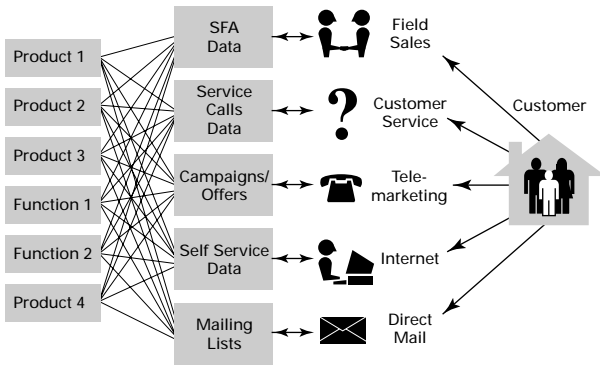


Figure 4 – Example of Product-Centric Systems

The technical challenge RPI needs to address is how to integrate this CRM application with its existing systems and its Data Warehouse in a manner that addresses the “Key Integration Requirements” in the previous section.

For simplicity, let’s focus on two key systems that need to be integrated with the CRM application.

- Auto Insurance System

This is a policy administration system that supports the auto insurance product range. It provides quotation, claims, and policy management functionality. It is a modern application with an API that allows us to access the functionality of the application and to intercept any changes that are made on the system.

- Home Insurance System

This is a policy administration system that supports the auto insurance product range. It provides quotation, claims and policy management functionality. It is an old mainframe application and the only way to access the functionality and data on this system is either via IBM 3270 screens or through some standard batch reports.

- Credit Check System

This is a credit check system that is actually run by an independent credit scoring agency. It is accessed using a simple request/reply message protocol.

In order to implement the solution, RPI took a decision to buy in as much of the integration solution as possible. This has a number of major benefits. It allows RPI to implement a solution quickly, it is more cost effective to maintain and it offers greater flexibility in the future as integration with more systems need to be added. The integration solution is therefore, based on an Integration Framework product suite. (See Figure 5).

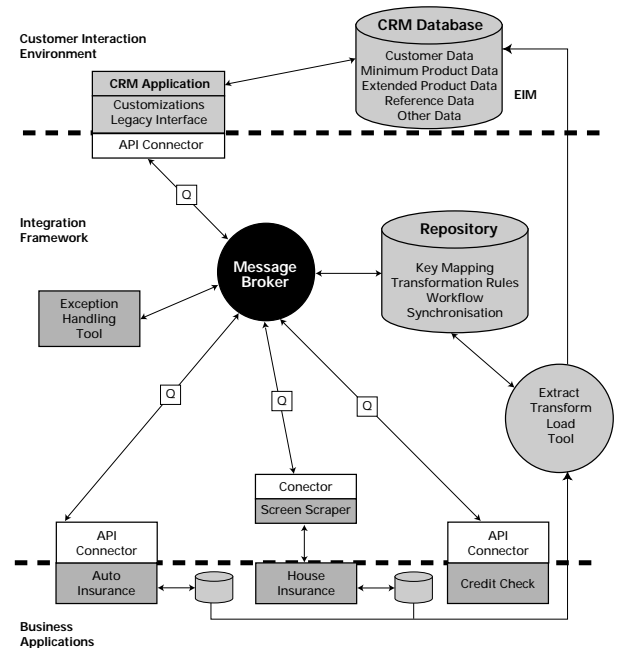


Figure 5 – Functional Components of an Integration Framework Product Suite

There are three layers to the solution. At the top is the CRM application with its own relational database. At the bottom are the Auto and Home Insurance systems and the Credit Check. Between these two layers is the integration solution built using an Integration Framework product suite.

At the center of the Integration Framework is a Message Broker, the purpose of which is to route and exchange information between the CRM application and the insurance systems. The message broker uses a database repository to provide data transformation, business process workflow, routing, and synchronization rules. The message broker uses IBM's MQSeries as a transport mechanism, which is readily available on most systems.

The message broker communicates with the CRM application and insurance systems using a component called a Connector. In the case of the CRM application and the Auto Insurance applications, the connector translates messages from the broker into direct calls to the application, using whatever APIs are provided. There is no API to the House Insurance system, however, the connector uses a Screen Scraper tool to access the application using a terminal data stream.

Particularly when using screen scraping, there is a possibility of error situations being detected by the message broker. To allow these to be resolved manually, an Exception Handling Tool is used.

To perform periodic batch data transfers between the CRM application and the insurance systems, an ETML tool is used. This tool uses the same data transformations defined in the message broker repository.

The integration solution relies on the following approach to data management where data is classified and treated according to its classification.

- Customer Data

This is customer profile information and any attributes of customer (e.g. name, address, date of birth, etc.). This customer data is defined in the generic or customized database schema of the CRM application. The integration solution assumes that this data is mastered by the CRM application and is only changed by the CRM application. That is, the CRM application becomes the 'System of Record' for customer data. On initial deployment of the solution, the customer data is bulk loaded from the insurance systems (probably via the Data Warehouse). From then on, any changes made to the customer data by the CRM application will be propagated through to the relevant insurance systems by the integration solution. Changes to customer data will not be permitted directly on the insurance systems.

- Customer Data Tags

These are some specific customer attributes that are reserved to hold information about the customer based on the analysis performed in the Data Warehouse. These would include attributes like customer value, propensity to churn, product to attempt to cross-sell, etc. These values will be loaded on a periodic basis from the Data Warehouse.

- Minimum Product Data

This is the minimal set of policy or 'product' data that the CRM application requires for interactions with the customer, even when the underlying insurance systems are not available and performance is inadequate. This is stored in a customized area of the CRM application schema. The integration solution assumes that this data is

mastered in the appropriate underlying insurance system and can be changed by the insurance system or by the CRM application. On initial deployment of the solution and on a periodic basis (e.g. weekly) the minimal product data is bulk loaded from the insurance systems. From then on, any changes made to minimal product data by the CRM application will be propagated through to the relevant insurance system by the integration solution and vice-versa.

- **Extended Product Data**

This is additional information about products, or other objects associated with a customer, from the underlying core business systems that is accessed by CRM application in some customer interactions, but is not required to be available when the underlying insurance applications are unavailable. This is essentially a compromise between the size of the CRM application database and the ability to function 24x7. The goal here is to ensure that the bulk of customer transactions can be handled by the minimum product data set. This data is not stored permanently within the CRM application but a customized area of the CRM application is used for staging purposes. The integration solution assumes that this data is mastered in the appropriate underlying insurance system and can be changed by the insurance system or by the CRM application. Any attempt to access this data by the CRM application will cause it to be fetched synchronously from the relevant insurance system and stored temporarily in the CRM application. Any changes made will be propagated back to the insurance system.

Let's see how this works by looking at a few simple transactions and see how the key integration requirements are met:

House Insurance customer phones call center to inform RPI that they have a new telephone number.

1. Call Center Agent changes address through CRM application.
2. Customer ID and associated old and new telephone number is passed via CRM application connector to message broker.
3. Broker performs lookup and queues change of customer details message to house insurance update queue.
4. Screen scraper connector reads message (will block if House Insurance system not available).
5. Screen scraper performs data integrity check (i.e. checks that telephone number on House Insurance systems matches old value).
6. Screen scraper updates telephone number on House Insurance system with the new number.

Auto Insurance premium updated following claim. Auto Insurance customer queries monthly premium via Internet. (Insurance premium is part of minimum product data set).

1. RPI employee updates auto insurance premium on Auto Insurance System.
2. Premium change is picked up by Auto Insurance connector and passed to broker.
3. Broker passes change to CRM application connector and value is updated in CRM application database.

4. Customer accesses CRM application via the Internet.
5. Monthly premium information displayed from CRM application database.

Business Benefits of an Integrated Contact Center Strategy

In a large, successful company, effective CRM is increasingly based on a corporate contact center strategy. A comprehensive contact center solution gives customer-facing constituencies, the customer and salespeople, instant access to a single, up-to-date “customer view,” allowing customer contact optimization. Benefits will be felt throughout the company if the contact center is able to collect and process customer-related information efficiently and effectively.

Many customers become impatient with companies that cannot respond immediately, at any time of day or night. For example, an order placed through the Internet must be immediately viewable by a call center agent if the customer calls to re-confirm the item or to change a feature. New events and information must be immediately available for use in the next interaction. Any delay in updating the customer’s file that leads to an unsuccessful contact should be considered as unacceptable.

For companies that have large numbers of customers, it is important to treat the customer as a unique individual. The organization’s ability to access information about the customer, such as their prior contacts with the company, the types of purchases they have made and any preferences they have expressed,

can provide the customer with a sense of their value to the company. This must be achieved regardless of the purpose of the contact or the contact channel being used. Every interaction with the customer must be effective for the customer, efficient for the company and successful in increasing both the knowledge base and the number of relationships.

An integrated customer contact system can provide a range of important point of contact benefits. These include:

- Easy access to all available customer information (including common formats and definitions for the data) allows the agent to gather useful data without repeating questions that have previously been asked. If the customer is asked for an address each time they make a contact, for example, it would be easy to become disenchanted with the company and feel anonymous.
- Minimization of keying overhead during the customer interaction process is also important. Being required to enter a name in multiple application windows, especially with customer self-serve channels, cannot be considered as acceptable. A company agent should be able to focus primarily on the customer, not on getting the system to work.
- Real-time updates and continuous system availability have become a necessity, especially when the customers may be located worldwide. Many sales can be lost if Internet access has to be suspended when, for example, a system back-up is being taken.

A customer contact system allows the company to move beyond simply serving every customer better. Customers can be differentiated on the basis of their value to the company (e.g., priority service can be given

to requests from high-value customers). Selling can be focused on the customer's preferences or can be based on prior history (an automotive policy was the initial query but a package home/auto policy would be a much better product, for example). A real competitive advantage can be derived by fully exploiting the knowledge gained both at the point of contact and as a result of correlating customer data across many contacts.

One of the major functions of a customer contact system is the integration of existing systems into a customer contact environment. (See Figure 6).

The ability to do this can be a major benefit to the company. Every large business is constrained by the need for a gradual evolution of computer systems – massive re-developments of business applications are seldom feasible or successful. Company mergers and acquisitions may leave a legacy of different systems that are not consistent or cohesive – none of which should be visible to the customer. Many companies choose not to build their own applications – they buy what they can and adapt it to their own needs. All of which means that presenting a common “look and feel” to the user, either directly or via an agent, is non-trivial. The application integration platform, as described earlier, helps to overcome the difficulties.

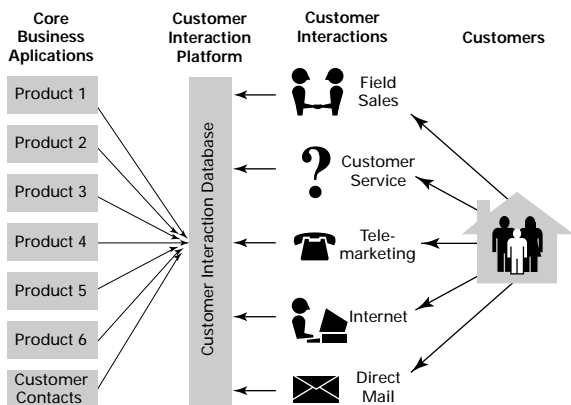


Figure 6 – Example of Customer-Centric Systems

Conclusion

The objective of any contact with a customer should be to make it count – through increased revenues (i.e., make a sale), reduced costs (i.e., perform efficiently) and improved customer relationships. CRM, a key goal of customer-focused business strategy, is difficult to achieve without re-developing the legacy, product-oriented systems that exist today.

Any new system can be designed and built from scratch or can be developed as enhancements to existing systems. Contact center systems depend both on collecting and analyzing a wide range of data about the customer and on being able to present it quickly and effectively to the customer themselves or to the company agent. The very high cost of application re-development, however, makes replacing existing systems unacceptable. The solution has to be adding new capabilities that integrate and re-package the legacy systems.

Customer information files can be integrated by adding a data warehousing facility as a back-end system. The ability to extract information about the customer allows modeling, churn analysis, campaign management, and other analyses. The opportunity to ask questions that add to or confirm the company knowledge base provides a clear advantage in dealing with the customer.

Customer interactions can be improved by introducing a customer-centric layer in front of the legacy systems and by adding easy-to-use presentation services that encapsulate the interaction process. This provides an opportunity to harmonize all channels and ensure the customer receives the same service regardless of how they contact the company.

An IT vendor that can help create this customer interaction environment must provide a combination of skills and tools, including:

- *People*, who are able to develop the software and interfaces that can both complement and supplement existing legacy systems.
- *Processes*, which can be implemented using consistent software elements that integrate with the underlying business applications both from a technical and a business perspective.
- *Partners*, which allows the use of “best of breed” products within the environment and ensures the availability of expertise.
- *Platform*, which provides the underlying technical support for availability and accessibility using popular operating systems such as Unix and Windows NT.
- *Proof of viability*, which can best be demonstrated by having a track record of implemented systems and satisfied customers.

Glossary

Application Program Interface (API) — Software library that allows a programmer to access well defined facilities of an application. An API isolates access to these facilities from how they are implemented.

Asynchronous — Form of communication where sender of message is not waiting for receiver. Email would be an example of an asynchronous form of communication.

Common Object Request Broker Architecture (CORBA) — Standard architecture for Object Request Brokers produced by the Object Management Group.

Connection-less — Form of communication where sender and receiver have no direct relationship with each other but talk through an intermediary. HTTP is an example of a connection-less transport protocol.

Connection-orientated — Form of communication where sender and receiver are directly linked. TCP/IP is an example of a connection-orientated transport protocol.

Conversational — Communication style where sender and receiver have an extend dialogue.

Customer Relationship Management (CRM) — Business strategy that focuses on maximizing profitability by understanding and exploiting customer behavior.

Enterprise Application Integration (EAI) — Class of software products that are used for linking applications together.

Extract, Transform, Move and Load (ETML) — Class of software products that provide asynchronous integration at the data level and that focus on bulk movement of data.

Gateway — Software that translates different protocols.

Message Orientated Middleware (MOM) — Type of middleware that is implemented by passing packets of data via queues. IBM MQSeries is the most widely implemented MOM product.

Message Passing — Communication style where sender sends a message to receiver but does not wait for any response or acknowledgement.

Message Queuing — Communication style where sender sends a message to an intermediate message handler which will hold it in a queue. The receiver can then read messages from the queue.

Middleware — Class of software product that enables communication between two or more applications in a distributed computing environment.

Object Management Group (OMG) — International industry consortium for defining standards for object-orientated distributed computing. The CORBA standard is a key deliverable of this group.

Object Request Broker (ORB) — Software component that transmits object requests to distributed objects and returns the result to the requestor.

Publish-and-subscribe — Communication style where sender sends information to an intermediate service. Intermediate service will forward that information on to any receivers who has registered for that type of information.

Pull — An integration style where the user of a data item is responsible for getting the latest value from the system of record, usually via an intermediary.

Push — An integration style where the system of record for a data item is responsible for forwarding any changes to that item other users, usually via an intermediary.

Remote Procedure Call (RPC) — Style of API where call actually invokes a function on a remote application. Often used to describe the 'sockets' API which was the first de facto standard for implementing distributed applications on the Unix operating system.

Request/Reply — Communication style where sender sends message to receiver and waits for a response.

Sockets — De facto standard RPC for UNIX and widely ported to other platforms.

Store-and-forward — Same as Message Queuing

Synchronous — Form of communication where sender of message waits for receiver to reply.

System of record — The system that is deemed to have the correct value for any data item.

Wrapper — An API that provides a simpler form of access to the functionality of a complex application.

