SISU

PUBLIKATION 98:06

DOKUMENT – MAJ 1998

Initial Usability Evaluation & System Architecture

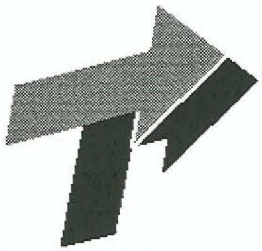
The RITE Project

Matts Ahlsén
Mathias Axling
Per Fossum

SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING

SISU

Interim Report on the Demonstrator:
System Architecture



Version 1

February 1998

SISU

The Swedish Institute for Systems Development

RITE är ett EU-projekt som syftar till att integrera och utvärdera teknik för datorstött samarbete och content management. Projektets användarparter representerar mindre företag inom europeisk musikindustri, vilka bla provar projektets resultat inom marknadsförings- och distributionsaktiviteter.

SISUs uppgifter i projektet har innefattat utveckling av en prototyptillämpning såväl som genomförande av användbarhetsprover.

Denna rapport beskriver den demonstrator som utvecklats. Rapporten utgör en obearbetad version av en leverabel inom projektet.

ISSN 1403-414X

Interim Report on the Demonstrator:

System Architecture

RITE P22078

Abstract

This report describes the revised system architecture for the RITE system demonstrator. The system has been redesigned to take advantage of changes in technology affecting the earlier design, enabling a flexible system based on open standards. The server part of the system has also been given more weight. We first state the objectives of the RITE project and the design principles used. This is followed by a description of the primary functions that the system will support. Then we describe the changes made to the earlier design and why, followed by a description of the revised system architecture. This is followed by more detailed sections on the technical design of the client and server parts of the system and an explanation of the new technologies employed.

Identifier	D67a-1.0- SISU-MAX-MA-PF
Class	**Deliverable (D)**
Version	**1.0**
Date	9802
Status	**Final**

DESTINATION

- CEC
- RITE Consortium

HISTORY

Version List of changes, Author(s)/Revisers		Date
Version 0.1, document created	Mathias Axling	9801
Version 0.5	Mathias Axling	9802
Version 1	Mathias Axling, Matts Ahlsén	9802

TABLE OF CONTENTS

1 INTRODUCTION 1

1.1 PROJECT OBJECTIVES OVERVIEW 1
1.2 THIS REPORT 1
1.3 DESIGN RATIONALE 1

2 FUNCTIONS 3

3 SYSTEM ARCHITECTURE 4

3.1 CHANGES FROM PREVIOUS DESIGN 4
3.2 THE REVISED ARCHITECTURE 5

4 SERVER SIDE SERVICES 7

4.1 PUBLISHING AND ACCESS CONTROL 7
4.2 DATABASE ACCESS 8
4.3 WORKSPACE FUNCTIONALITY 8

5 CLIENT SIDE SERVICES 9

5.1 COMMUNICATION 9
5.2 DATA ACCESS 10

6 SYSTEM DESIGN 13

6.1 SERVER SYSTEM DESIGN 13
6.2 CLIENT SYSTEM DESIGN 13
6.3 TECHNOLOGIES 15

7 SUMMARY 17

1 INTRODUCTION

1.1 Project Objectives Overview

The objective of RITE is stated in its very name – *Radical Improvements to Efficiency*. It aims to solve the business problems of inter and intra company communications, especially where these are related to the needs of the music industry and similar multimedia intensive areas.

One of the goals of RITE is to ensure that no esoteric technologies are employed and another is that all solutions must be standards compliant. The product so far has met both of these objectives.

1.2 This Report

This is primarily a technical report describing the architecture of a prototype implementation of the RITE system. The report's main purpose is to serve as a documentation of the systems design for the projects continued development. It will also be used to obtain feedback on the proposed design from parties external to the project.

Since the last review, both changes in technology and the project organisation has resulted in some redesign of the system. The changes are totally compliant with the design rationale described in earlier documents and in the next section. The design changes are thoroughly described later in this document. The Rite system will, no doubt, be subject to further redesign in the course of user testing and validation. This document is intended to be refined and complemented in the course of that work.

1.3 Design Rationale

The design of any software artefact requires a careful balancing of a number of architectural qualities, not unlike the design "physical" objects to be used by humans, like buildings or machinery. The difficulties in providing "good" designs for software stem from its abstract nature and, the (still) poor understanding of "architectural styles" for software systems. The technical or engineering qualities are fairly well understood (although no consensus exist on design methods), whereas usability qualities can only to a certain extent be designed into a software product from start. Specific difficulties arise when designing with 3rd party components from different vendors, which is an experience the RITE project shares with many others.

In the RITE design process a major emphasis has been put on balancing simplicity with function, appreciating the above mentioned difficulties. The overall design objectives for RITE are to:

- provide a coherent set of communications and database retrieval functions for multimedia communications
- provide the system in an industry standard computing environment, based on adapted 3rd party software as far as possible.

Some desired properties of the resultant design include,

- synergy for the user, through the combination of functions
- extensibility, in accommodating new or improved software tools

- modularity, the system should be based on well-defined components (promotes extensibility)
- openness, i.e., employ or be prepared to employ de jure and de facto standards (may promote modularity)

2 FUNCTIONS

The result of the project system pilot will be a product that provides a comprehensive co-work support system based on integrating multimedia database technology with a number of options for networked communications services. The system is based on commercially available PC software and on state-of-the-art Internet technology. The system includes and integrates a set of functions for:

- Personal Information Management (PIM), including contacts management with address data and communication channels
- Unified Messaging Interface, including email, fax and videoconferencing)
- Generation of multimedia presentations of business data
- Shared workspaces for collaborative tasks
- Access to personal and shared databases
- Access to information sources and services on the Internet

These system functions can be divided into three main categories, presented and calibrated in the RITE Clients to suit the particular work processes of the different users. These categories are communication, data transfer and data access. They are implemented both in the clients (communication) and the server (data transfer, data access). They are supported in some parts by third party software components and in others by RITE system components. The purpose of the RITE system is to glue them all together transparently to the user. All these functions should be just a couple of mouse-clicks away.

2.1.1 Communication

The system shall make it easy to communicate i.e. by e-mail, fax, telephone, videoconference application sharing and whiteboarding. Most of these functions will be third party components. The system should also hide the intricacies of these systems from the user.

2.1.2 Data Transfer

The system shall perform much of the transportation of music, images, multimedia and documents, that so far is being done off-line (by courier).

2.1.3 Data Access

The RITE System shall make it easy to access databases, the RITE database and other interesting databases, as well as web pages and "active channels". Preferably, all data in the organisation should be accessible from the RITE system and never duplicated.

3 SYSTEM ARCHITECTURE

The RITE System consists of a set of general services available on the RITE server, and clients containing sets of Client Components with "thin" functionality that support business specific work processes. After having tested the prototype of the new client, the first demonstrator, at the user site, we decided that it was now time to concentrate our design work on the server parts. The general response to the client prototype we got during the tests was positive, and we were quite satisfied with the client side architecture. Of course the client needs to be refined and some different versions made to illustrate the possibilities, but the overall design is clear. The real functionality of the RITE system is in the server, though, and the full potential of the system cannot be illustrated without a server demonstrator.

3.1 Changes from Previous Design

The design changes made since the last review are due to technological changes and mainly concern the client part of the RITE system. Two changes have been made: the shell extension technology has been replaced, and ISDN is no longer a central issue in the project, although it is still an option. These changes are necessary to keep up with the rapidly changing technology in this area and take advantage of all new possibilities.

The RITE system clients were formerly based on a technique called *shell extensions*. A shell extension, sometimes referred to as a virtual folder, provides a way to seamlessly integrate an application with the Windows user interface. A shell extension defines a new object that the Windows Explorer can explore, i.e., it can be displayed and browsed as any other object in the Windows environment. You are able to define the icon images and text that the user sees while viewing your data, as well as the menus, toolbars and status information the user can use on your data objects. The standard Windows objects My Briefcase and the Recycle Bin are examples of virtual folders.

With the introduction of Internet Explorer 4.0 and the Windows 98 beta version in fall 1997, however, this technique was no longer necessary for the RITE system. The Internet browser is now also the browser for the operating system, enabling the user to browse folders and web pages in the same way. One could say that Internet Explorer is a shell extension that provides a virtual folder view for web pages and FTP directories. In short, what previously required a shell extension can now be done via a web page containing dynamic HTML, JavaScript, VB Script or ActiveX components. These techniques provide a more flexible and "plug-in" component-based way of building the RITE Client. It also greatly facilitates the administration of the system.

With Internet Explorer 4.0 and Windows 98 also comes another innovation: the Active Desktop. This allows us to place web pages directly on the desktop, making the RITE Client interface accessible at all times. No separate window for the RITE Client is needed, so it is even more integrated into the user environment than before, but now in a higher technical layer of abstraction than before. We can now easily change the interface radically without the need of low-level operating system rewrites.

Also, as the Internet has become more widespread and the organisations in question have started using it, a better system can be built using Internet technology rather than dial-up ISDN connections. Note that ISDN is still an option or a possibility, though it is not a central issue within the project. The further development of the RITE system will not go into optimisation of videoconferencing or the intricacies of ISDN communication. These things are now considered plug-in functionality. The quality and speed of the communication is an issue best solved by the manufacturers of videoconferencing software and protocols, and network bandwidth is a problem best solved by the telecommunication companies, who have much more resources for this than the RITE consortium.

3.2 The Revised Architecture

The RITE system architecture has changed on the client side, although the main principles still hold and the vital components are still there. The server side of the system has evolved more than changed. The new client side techniques and the near omnipresence of the Internet have enabled a new and more server-centred design. This new architecture, which still leaves ISDN as an option, utilises the Internet and web techniques to build a more flexible and powerful base for communication and data management services for the record (and multimedia) industry.

As we have mentioned before, the system has three main function categories, communication, data transfer and data access. The key behind the new system design is that data transfer and data access is much the same thing if you are connected to the same network, e.g. the Internet. Taking advantage of that fact is what the new design is all about. E.g., one of the key functions of the system is to make it possible to transfer data between companies and people in the companies. This is now usually done by courier or by phone. Rather than actually distributing copies of data, as the case is in the test organisations now, we have decided to send references to data available on the network. This is preferable if you are connected to the same network and have data that should not be manipulated directly by the recipients. Of course, if the data is supposed to be altered in some way by the recipient, a copy could be sent via file transfer or e-mail.

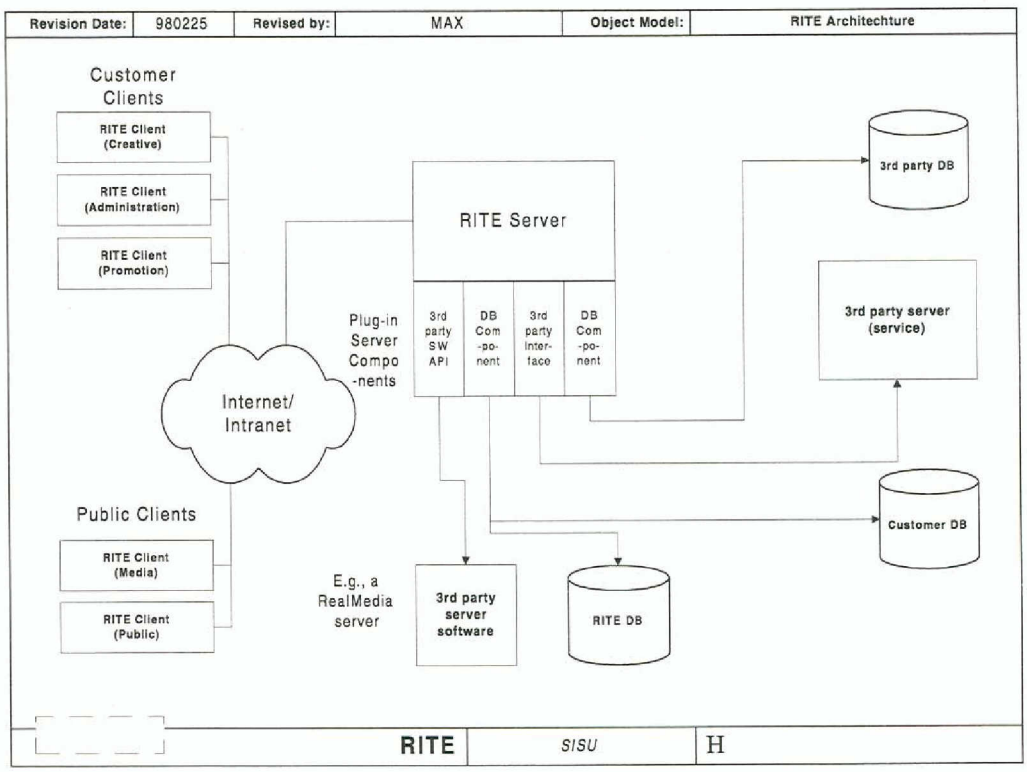


Fig. 1:The RITE System Architecture

This new way of distributing data is similar to the way the Internet works today. Instead of communicating by sending pieces of information to each other, one publishes the information and gives the recipients a link, URL, to the data.

The RITE client is a DHTML (Dynamic Hypertext Mark-up Language) document that provides a framework for presenting and selecting between several customisable Client Components. The interface can easily be tailored for any function within the company, e.g. marketing, design or management. Some components are interfaces to RITE system functions, while others are interfaces

to standard third party software. The communications component integrates different third party communication software, such as address books, videoconference and fax. The DHTML document and the components reside on the RITE server and are downloaded to the client when the user first uses the system. This greatly facilitates installation and upgrading of the system.

The RITE Server consists of a central Server Component, which is aware of the business objects on a higher level and handles access control and automatic functions. It also calls other Server Components that handle general RITE System Services, like those mentioned below. The central Server Component also has APIs for controlling and getting events from specific third party server software. These APIs are the parts that have to change when third party software is changed. The system is administered from the RITE server and this is where the RITE-specific services are located.

4 SERVER SIDE SERVICES

The server components will together implement the general server side services that are accessible from the components via various interfaces. These services will be used to build the more complex services of the RITE clients. The services that users showed particular interest in during the usability tests will be the first to be implemented. They are:

- Publishing of data
- Access control
- Database access
- Shared Workspace functionality

4.1 Publishing and Access Control

Publishing data and access control is really part of the same thing. The system focuses on referencing views onto data in the system. These views could be seen as "filters" for the information that both handle access control and presentation. Data only exist in the company database, and it is then referenced with different views applied to the data, with the basic view being only the data itself. The view applied could be dependent on the user, so access control will be implemented this way

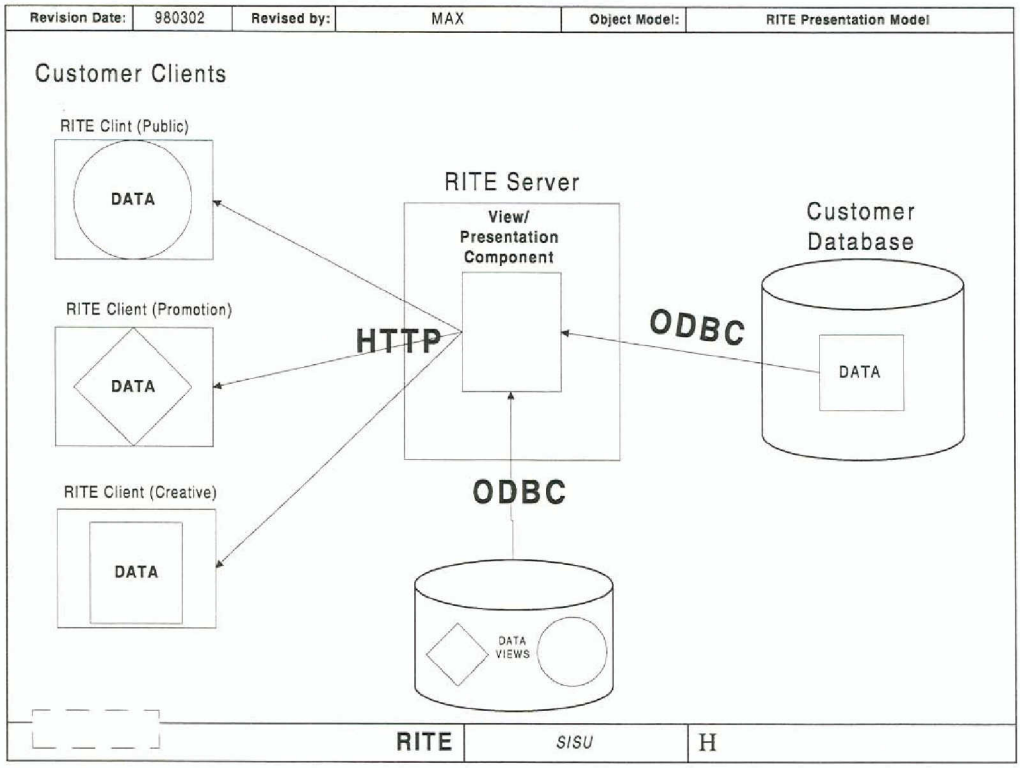


Fig. 2: The RITE Presentation Model

If the user is not authorised to access certain data, the view applied for that user could be a message containing e.g. a textual description of the data combined with an email form to fill in where the owner of the data can be requested to personally distribute it.

Presentations and newsletters are views on data that are predefined for certain types of data. These views can be selected as templates together with the data that will be filled in the templates. E.g., a "release" template is designed once, and then applied to different releases - which are a collection of related database entries - just by the click of a button. There will exist a "record release" presentation for every record release; it's just a matter of defining it once and then applying it to all possible releases.

4.2 Database Access

The customer business databases and external databases containing e.g. sound samples or pictures are accessed by the client via references. The references are translated by the RITE Server into database queries for the different databases. Those databases are then accessed from the server via ODBC, so the RITE server can control all database queries for validity. The communication between client and server is via HTTP. This also allows all RITE customers to access all kinds of public databases, even if they themselves only have a dial-up connection to the RITE system and no Internet connection or direct connection to the databases. The RITE system could offer access to other attractive databases as part of the system services. The RITE Server could also host the company databases for smaller companies, offering a complete set of outsourcing services accessible via TCP/IP and a web interface.

4.3 Workspace Functionality

The workspace functionality is intended for the creative, or pre-release stage of data, when the data is dynamic and subject to change, new data is frequently added and the data is only to be seen or altered by a small group of people. They can then store the data in a shared virtual file system, with added functionality for meetings, version handling and of course access control. This workspace can be shared over the local network or over the Internet. When the pre-release work is finished and the data is stable, it can be inserted into the company database.

The workspace code used is a modified version of a system developed at SISU, based on the BSCW (Basic Support for Co-operative Work) system from the German research institute GMD. It is accessed via an HTML interface that is part of the RITE Client. The HTML interface will be redesigned to fit into the RITE environment and special functionality will be added for tighter integration with the RITE System.

5 CLIENT SIDE SERVICES

The client side services consist mainly of the software supporting communication and data transfer: third party software for e-mail, videoconferencing, application sharing, fax and web browsing. It also has components for data access. The different components are used via simplified interfaces in the RITE Client.

5.1 Communication

The RITE system includes a set of standard functions for user to user communications. RITE is based on a series of expandable communications modules. Whilst video conferencing is one of these it has no higher priority than, say, file transfer or faxing. All the addresses and numbers for email, fax, telephone and videoconferencing are stored in the Personal Information Management component, which is the user interface for all communication. All the communication services are initiated from this component. This set of modules includes:

- Video conferencing
- Application sharing
- Whiteboarding
- File transfer
- E-mail
- Fax creation and sending

5.1.1 Video Conferencing, Application sharing, File Transfer and Whiteboarding

Video conferencing will enable users to have face-to-face interaction with other members of the independent music community. As the system is standards-based, inter-operability to third party will be possible.

There are numerous situations where it is useful for two interconnected parties to be able to 'see' the same document or graphics file and for them both to work on it to reach a consensus. The application-sharing tool will enable any Windows application to be shared between users. A special case of application sharing is commonly referred to as whiteboarding. This function will allow co-working between users by employing a common 'whiteboard' or 'notepad' where text and imported images can be viewed, marked-up, agreed and stored. At the moment, these functions are done using Microsoft NetMeeting. These services are carried over TCP/IP (could be an ISDN connection). They conform to H.323 with the T.120 family used for file transfer and whiteboarding.

5.1.2 E-mail and Fax

The most used application on the Internet is e-mail. This functionality and more is available via the use of MS Outlook object libraries. The Contacts component uses these libraries to store contact data, and to send email and fax (based on MAPI, Windows' Messaging Application Programmers Interface).

5.2 Data Access

5.2.1 Web Browsing

For web browsing, Internet Explorer 4.0 and later releases are used. This will probably be optional in later releases, even though the Client Interface currently needs Internet Explorer 4 to function. A contacts web page can be accessed from the Contacts interface.

5.2.2 Database Access

Two forms of remote database access are being built into RITE; (1) advanced structured navigation and querying based on the Intuitive tools and (2) easy-to-use database interfaces with full drag-and-drop functionality, in several designs adapted for different user needs.

Both will be available via the RITE Client interface and can be seen as complementary ways of extending the information space available to a RITE user. The easy to use database interfaces are recommended for standardised tasks and well-defined work processes, while the Intuitive interface is recommended for advanced searches, free database navigation and system administration. While the specialised interfaces are easier to use, they must be especially made for each database and task. The Intuitive interface is possible to use with the whole database, and with databases for which no specialised interface components are written.

5.2.3 Intuitive

The database functionality in RITE builds on an existing set of software tools developed by SISU in previous research projects. We give a brief description of the Intuitive software architecture before explaining the adaptations made for the RITE system.

The INTUITIVE system provides the user with an overview of the available information together with facilities for formulating queries and, finally, evaluating the retrieved information. The integrated information space accessible through Intuitive can be organised as a collection of heterogeneous databases. The common foundation for these functions is the use of an extended Entity-Relationship (ER) model.

Fig. 3 below shows the INTUITIVE tools in action. In the upper left corner, a Navigator is displaying a conceptual overview of the available information space. In the context of the RITE applications Navigator will show an ER-model of music business related data. To the upper right there is a Selector that allows a user to formulate queries in a user-friendly manner without having to care about the underlying database structures or the formats.

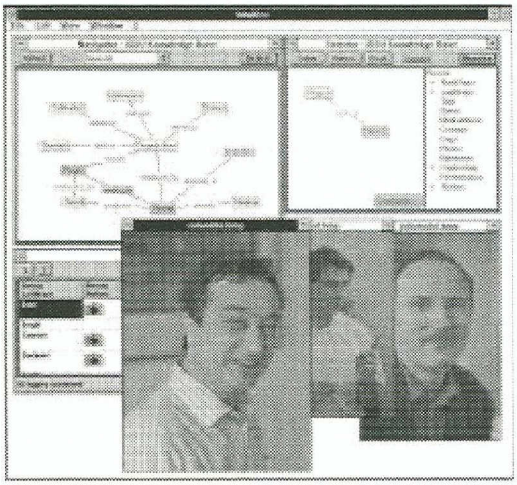


Fig. 3: Example of picture retrieval by means of conceptual maps using INTUITIVE tools.

Additionally it is possible to simultaneously have different representations of the same information space. One Navigator could show a conceptual information model, while another Navigator could instead show a media hierarchy, that explains what kind of media is available. This provides the user with facilities to navigate and discover information using the most suitable view of the information space.

Standard INTUITIVE includes a set of generic user tools that package the functionality outlined above. These user tools allow:

- Navigation within conceptual views of the whole information space (Navigator and Inspector tools)
- Formulation and refinement of pre-formed and ad-hoc queries (Selector tool)
- Browsing and presentation of results (Results Browser and Presenter tool)

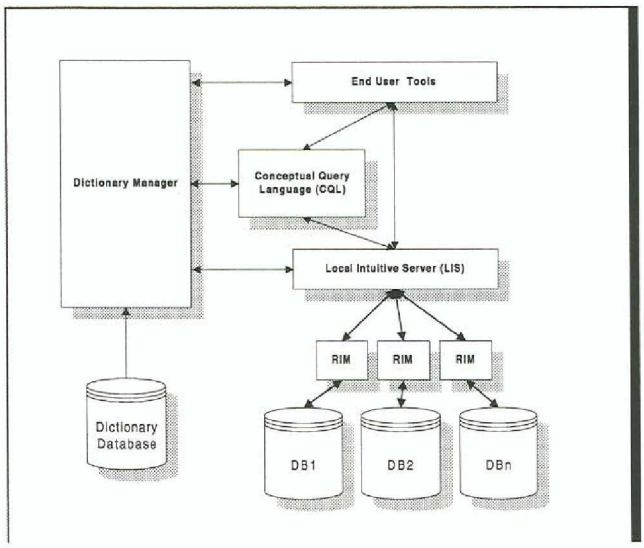


Fig. 4: Modular decomposition of basic Intuitive functionality

A modular decomposition of the basic Intuitive functionality is shown in Fig. 4. In the original version these modules are not independent software components and hence not accessible from other software.

- Dictionary Manager is responsible of providing the system with the necessary meta dictionary data such as views, conceptual classes etc. The Dictionary Manager provides both conceptual and internal meta dictionary information. The Dictionary Manager uses an ordinary relational database as its persistent data store.
- End user tools, which provide the user with the means to interact with the system. The standard Intuitive system supplies the user with four basic tools, the Navigator, the Selector, the Browser and the Presenter tool.
- The Local Intuitive Server (LIS) is the main query engine of Intuitive. The LIS uses queries represented in CQL (the Conceptual Query Language) and delivers the result of the query to the End User Tools. If necessary the LIS split the CQL query in to several sub queries, i.e. this occurs when the CQL query spans multiple physical databases.
- RIMs ('Resource Interface Modules') implements database access that creates independence from particular database systems. Each RIM translates from INTUITIVE's conceptual query language to the specific database language. (e.g. ODBC SQL).

Meta Dictionary Structure

The Intuitive Meta Dictionary structure is based on a traditional three layer schema. Currently it mainly implements the database internal layer and the conceptual layer. The presentation layer is not implemented. The following describes the most central object types in the dictionary structure.

- ***Entity*** - represents an information object with a natural meaning in the users business. Entities have a set of ***Attributes*** that hold data values (name, salary, shoe size etc.). Entities are connected by means of ***Relations*** that reflect natural dependencies among the information objects.
- ***Datatype*** - all attributes have an associated datatype defining the domain of allowable values for that attribute. ***Valuesets/ enumerations*** - these are used as complements to datatypes. Valuesets can be either statically defined or dynamically generated. Valuesets are also used in the definitions of conditions and restrictions when formulating queries (in the Selector tool). ***Rules*** - these are a simplified form of regular expressions used for input validation when a user defines conditions for queries in the Selector tool.
- ***Model*** - a central object in the Intuitive system. It represents the collection of all Entities and their Relations describing information from one or more databases. ***View*** - a View represents a subset of some Model, a Model may have several possibly overlapping Views defined for it. ***Resource*** - a Model may describe information from several databases. Each such database is then referred to as a Resource.
- ***Viewgraphics*** - controls the presentation and positioning of Entities and Relations in the Navigator tool. Each graphical object belongs to a View, has its own co-ordinates and refers to a conceptual definition. ***Renderer*** - is an object that is used to present different types of resources in terms of images, documents, videos, URLs etc. in the Results Browser tool.

The tool set could in principle be used in the RITE environment as an independent application, without any significant modifications.

Meta Data Management

The Dictionary Manager (DM) has the overall responsibility for maintaining descriptive data (meta data) relating to databases accessed with the tool set. The DM is implemented as an OLE automation server. The API that had to be implemented is quite straightforward since the Dictionary Manager object was already implemented with a well-defined set of member functions.

6 System Design

The system design is based on independent software components. These are glued together by standard techniques like COM and ActiveX, communicating via standard Internet protocols like HTML, and built with well-known programming tools and languages like Visual Basic and Java.

6.1 Server System Design

The server part of the RITE system consists of several parts. First, a central Server/Control Component that handles interactions between the business objects that encapsulate specific data in the system. This component implements the business rules and access policies and also controls the other components. Then we have the RITE Server Components, which implement the different server side services mentioned above. These use the same business objects as and can be plugged in to the central Server Component. The third party components are interfaced via APIs that encapsulate the specifics of these servers and describe their functions in terms of the business objects.

The central Server Component models business rules and dispatches calls to the other components. It also handles all access to the databases, and modifies the query results so that the result appropriate for the current user is a returned. For instance, if the user is not an employee of the customer, a sleeve or song sample could be returned that is of lower quality. This way, it could not be used to just get an idea of what the sleeve looks like but not be suitable to be reproduced or redistributed. A simple way to plug in new Server Components without having to recompile will be developed. Probably, COM/DCOM will be used to combine the different components. That way the different components can be written in the language that fits the component best. They could also communicate with the Client Components and with each other even if the different server components are located on other computers.

What third party software to use can be decided later, but new API stubs would have to be written when new server software is purchased. Examples of third party software are mail servers, web servers, workspace servers, payment or transaction servers, or streaming media servers. It will also be possible to interface with third party services or servers located elsewhere on the Internet, e.g. music samples databases or Internet record shops. The idea is to be able to offer a complete set of business support services tailored for the music industry.

6.2 Client System Design

The Client Components are ActiveX components, web applications or Java applets that can be displayed within the Dynamic HTML document that constitute the "glue" for the user interface. The individual components can be tailored for specific work processes in co-operation with the users, while the framework for the interface remains constant. Each user can have his/her own selection of components, streamlining the interface so that the user has a minimum of unnecessary factors or unused features. Of course, a user not authorised to change the database should not have access to a component that supports database editing.

The development of new components should be guided by consistent interface design guidelines. All components should support drag and drop operations and behave in a consistent way. As we are currently using the Windows platform for the clients, the design principles we use are the same as for the Windows 9X operating system. This includes drag and drop, right-click to invoke operations on objects and double-click to open. When the prototype is made into a product, other clients can be

written for other operating systems; a Macintosh client for the artwork and design people will probably be necessary. Another interesting possibility is to implement a version for hand-held computers if the salespeople use such devices.

The components reside on the RITE server together with the DHTML page and are downloaded to the client when automatically when updated. This simplifies maintenance and administration. By combining components on a web page, the interface may easily be changed and many "cool" design solutions can be implemented, with JavaScript and VB Script. The web page may be placed on the user desktop by Active Desktop - a feature provided by Internet Explorer 4 or be accessed in a Browser window of Explorer 4. The interface can also be presented as the content of a folder – just as the shell extension. This is why the shell extension solution has been discarded.

Communication and data transfer functions are both combined in one component called *Contacts*, as they both are oriented towards the contacts (formerly known as "bods") that the users have. This control is using the features provided by MS Outlook (included in Office 97), and is tightly integrated with it. The Contacts component uses COM to access objects in the Outlook object library. The persistent contact objects are used to store information about email address, fax number, IP address and more. E.g., if a user right-clicks on a contact and selects "Videoconference", the Contacts control looks up the contacts IP address and starts NetMeeting with the instruction to start a videoconference with the selected contact.

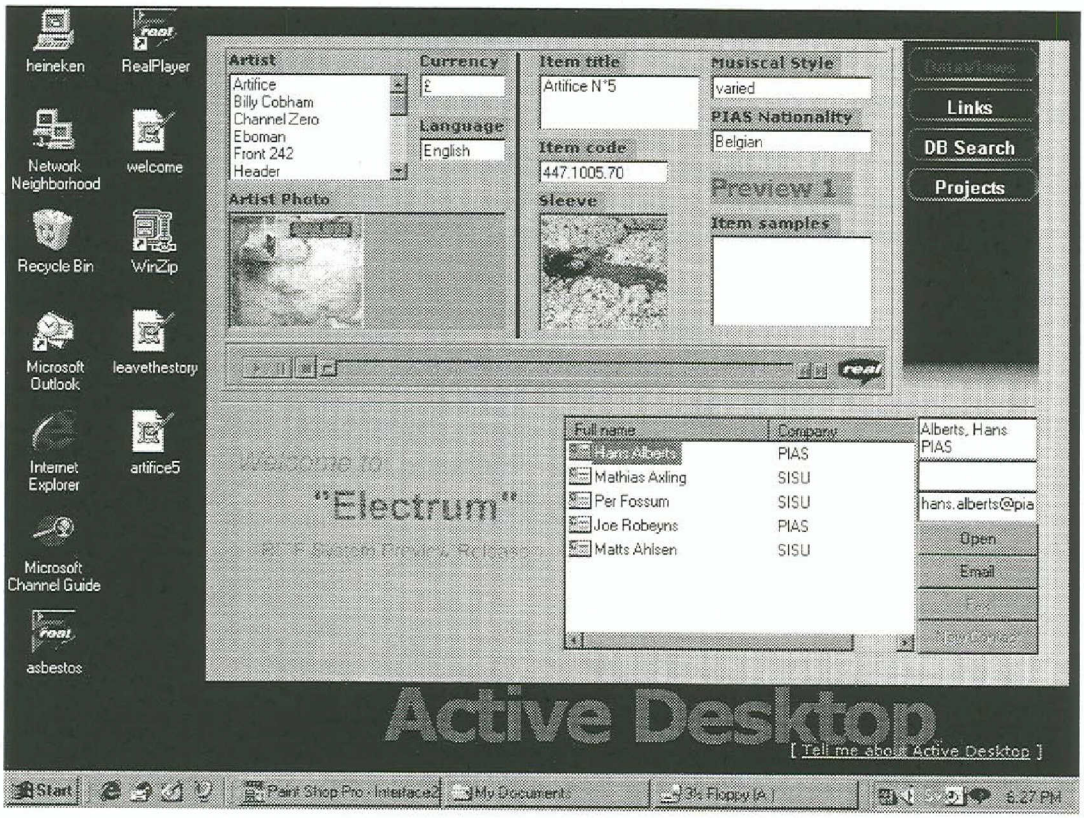


Fig. 5: The RITE Client with the Contacts component and the Artist&Item Data View. This is also an example of the use of Active Desktop functionality.

MS Outlook was chosen because it supported all necessary functions for a planner/address book, is widely used and is well integrated with the other components. For users with other address book preferences other contact components could be written, using the same interface. The free version of Outlook, Outlook Express, can also be used, although it runs slower at the moment. This is up to Microsoft to correct.

In order to support easy data access, specially designed components can be used to access specific parts of the RITE database. We have e.g. designed a sample component called *Artist and Item* containing multimedia information about artists and items, and that provides a different way of navigating the database than the "Intuitive way". Still, we can run Intuitive as e.g. a Java Applet, ActiveX or as a standalone application for the demanding user.

The components do not communicate directly with each other. Instead they are required to support Windows OLE drag and drop operations. This creates a very simple interface, is compliant with MS Windows interaction principles and makes the component developer independent of the other components. The developer does not have to know which other components there are or what operations they support. As long as the components can handle drag and drop with the relevant data types, all components can work together.

6.3 Technologies

6.3.1 Active Desktop

Active Desktop is a feature in MS Windows 98 and Internet Explorer 4.X that enables the user to display HTML (or DHTML) pages directly on the desktop. Several different HTML pages can be displayed at once and they can contain all objects a normal HTML page can contain, e.g. ActiveX components or Java applets. This is useful for information and functions that should be available at all time.

6.3.2 ActiveX

ActiveX controls are pre-compiled software components with a standardised interface that other components can call. An ActiveX component uses COM to publish this interface. The standard interface allows ActiveX controls to operate together with other COM-compliant software. ActiveX controls can be downloaded via the Internet and displayed on a web page, for instance. As they will be installed and run on the client machine and have access to the file system and registry, some caution is needed. (As opposed to Java Applets, which have no access to the file system.) In an application such as RITE, however, this is not a problem. An ActiveX control can be digitally signed to allow users to identify the developer before the component is installed or executed.

6.3.3 COM/DCOM

The Component Object Model (COM) is a software architecture that allows applications to be built from binary software components. COM is the underlying architecture that forms the foundation for higher-level software services, like ActiveX components. COM specifies how different software components call each other, and how the translation between different implementations of the same datatype is made. COM is also used to control and communicate with the Outlook objects.

6.3.4 DHTML

DHTML (Dynamic Hypertext Markup Language) is an extension of HTML (Hypertext Markup Language). As the name implies, DHTML allows the designer to construct web pages that exhibit dynamic behaviour. Used together with JavaScript or VB Script, the author can create pages where text, images, buttons and more can be changed, and moved around on the page without having to load another HTML page. This makes DHTML more of a language for constructing user interfaces than a language for document layout. Due to disagreements between developing companies (Netscape and Microsoft) DHTML is not yet standardised and the current version of the RITE Client runs only in

Internet Explorer. Since we are already committed to the MS Windows environment, however, this is not a problem and the language will hopefully be standardised in the near future.

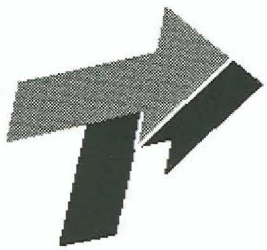
7 SUMMARY

The system design has been altered in some ways due to technological changes and a new project organisation. The changes mainly affect the user interface on the client side, as some technology has been made redundant and new technologies has been used in the design. Also, the server side has been given more weight in the revised architecture. This partly is due to the fact that the client side of the architecture has matured to the point that the effort now can be put into developing the server part of the RITE system. The system is more of a client-server architecture based on Internet technologies, although a dial-up connection will still be sufficient to use the system.

The system supports three main functions: communication, data transfer and data access. These functions are supported by software components in both the client and server part of the system. The system also plugs in third party software for standard functionality like video conferencing, email and address book. A presentation model has been adopted for publication and transfer of data, utilising the Internet and related technologies. The software components in the client are independent of each other and exchange data by supporting user interface operations like drag and drop. The clients communicate with the server via HTTP and COM is used to connect the server components. For access to the databases from the server, ODBC is used.

The revised architecture is modular, extensible, Internet standards compliant and builds on open, standards-based and widely used technology. It is a platform upon which to build different services supporting the record – or media – industry in their daily work. It could be used as a system internal to the company, or to sell outsourcing services to the smaller companies. It provides an interface to the business for managers, employees, customers and press.

Interim Report on the Demonstrator:
Initial Usability Evaluation



The RITE Project

Version 1

February 1998

SISU

The Swedish Institute for Systems Development

Interim Report on the Demonstrator: Initial Usability Evaluation

P22078

Abstract

This report describes the first user evaluation of the revised RITE system. The new systems design specifies a number of potential user functions and a subset of these has been built into a demonstrator. The first evaluation of that demonstrator is the subject of this report. The demonstrator was built following the participatory design approach adopted by the project team. The idea is to design and evaluate in an iterative manner, in close co-ordination to the end users.

Identifier	D67b-1.0- SISU-PF+MAX+MA
Class	**Deliverable (D)**
Version	**1.0**
Date	9803
Status	**Final**

DESTINATION

- CEC
- RITE Consortium

HISTORY

Version List of changes, Author(s)/Revisers		Date
Version 0.1, document created	Per Fossum	9801
Version 0.5	Per Fossum	9802
Version 1	Per Fossum	9802
Final editing	Matts Ahlsén	9803

TABLE OF CONTENTS

1 THE DEMONSTRATOR 1

2 USER EVALUATION OF THE DEMONSTRATOR 2

2.1 CO-OPERATIVE EVALUATION 2
2.2 SELECTING TASKS 2
2.3 TESTING THE EVALUATION 4

3 PERFORMING THE EVALUATION 5

3.1 THE USERS 5
3.2 THE CIRCUMSTANCES 5
3.3 THE EVALUATION PROCEDURE 5

4 INITIAL RITE EVALUATION RESULTS 8

4.1 GENERAL USER COMMENTS 8
4.2 INTERACTION ISSUES 8
4.3 MULTIMEDIA DATA ISSUES 9
4.4 MUSIC BUSINESS ISSUES 9

5 CONCLUSIONS 10

5.1 THE VALUE OF USER DRIVEN DESIGN 10
5.2 ARE WE ON THE RITE TRACK ? 10
5.3 FROM DATA COMMUNICATION TO MULTIMEDIA PRESENTATIONS 10
5.4 CONSIDERATIONS FOR THE FUTURE 11

6 REFERENCES 12

1 THE DEMONSTRATOR

The intended users of the RITE system are people working in joint production, marketing and distribution activities in the independent music industry. These users may be working in different geographical company locations, in-house or on the field, and they may communicate and exchange information both synchronously and asynchronously. These people work in a creative environment, where working relationships are informal (often with loud background music). The planning horizon varies and work is often driven by events, work load is unevenly distributed over time.

The RITE system is based on commercially available PC software and on state-of -the-art Internet technology. On a general level the final RITE system will include and integrate a set of functions for:

- Personal Information Management (incl. contacts mgmt with address data and communication channels)
- Unified Messaging Interface (email, fax and video conferencing)
- Generation of multimedia presentations of business data
- Shared workspaces for collaborative tasks
- Access to personal and shared databases
- Access to information sources and services on the Internet

The first demonstrator has been designed to illustrate a subset of these functions. The first evaluation of that demonstrator is the subject of this report. The demonstrator was built following the participatory design approach adopted by the project team. The idea is to design and evaluate in an iterative manner, in close co-ordination to the end users. Bringing the system designers closer to the users will create a mutual learning situation, where the developers learn about the users' world while the users learn what information technology can do for them and for their work. By bridging the gap between users and developers, a system that makes the users more efficient will be developed.

2 USER EVALUATION OF THE DEMONSTRATOR

In order to get reactions on what we had developed so far in the project (after the revised technical approach was adopted), we decided to meet with people from the business in a user evaluation workshop. By evaluating the demonstrator together with a record company we hoped to get feedback on our ideas and to make sure we keep the right focus of the system being developed. Are we doing what people within the music business needs? Is our product easy to use? Bringing the design people together with end users in a usability evaluation session seemed the right way to go. We chose to use co-operative evaluation (Monk et al, 1993), a methodology well-known (from the HCI society) for evaluating the usability of a system and good for gathering user comments.

2.1 Co-operative Evaluation

The purpose of doing usability evaluations is to find usability problems and to gather information about the users work processes and concepts associated with it. It may seem strange but finding problems is good, cause once they are found they can be corrected, and the earlier you find them the cheaper it will be to correct them. Strangely enough, the more problems you encounter, the better.

There are a number of methods to use for evaluating the usability of a computer system, such as heuristic and standards evaluations, questionnaires and interface walkthroughs. Each of the methods are good for finding usability problems, but since we also wanted the system developers to learn about the users business, we decided to use co-operative evaluation. With this method, end users as well as members from the design team go through the system together (co-operatively) and look for usability problems.

During the evaluation, the end user is given a set of tasks to solve with the system, and while he/she is interacting with the system the system developers are observing and taking notes. If the end user is commenting or hesitating during a task, it may be due to a usability problem in the system. When the user has solved the tasks, the system developers ask the user about what he/she thought of the system. What was the easiest? What is missing? Is feature X any good? The good thing about this method is that the user feels that he/she is participating with the development of the system. Therefor the discussions often become creative brainstorming sessions where new ideas is likely to pop up.

The results of co-operative evaluations are not only a number of usability problems, but also comments about the system as a whole, and new ideas about the system. Not less important, since the system developers themselves are participating, the user comments can easier be considered in the further development. No energy will have to be spent on convincing the developers about why a certain feature of the system is needed (as is sometimes the case in software projects) – they have been told themselves by the users.

2.2 Selecting Tasks

By solving the tasks, the user gets a feeling for what could be done with the system. If there are too few tasks, the user will not evaluate all of the system. On the other hand, too many tasks may take too much time and be boring to the users, leaving less time and energy for the discussions that follow the tasks.

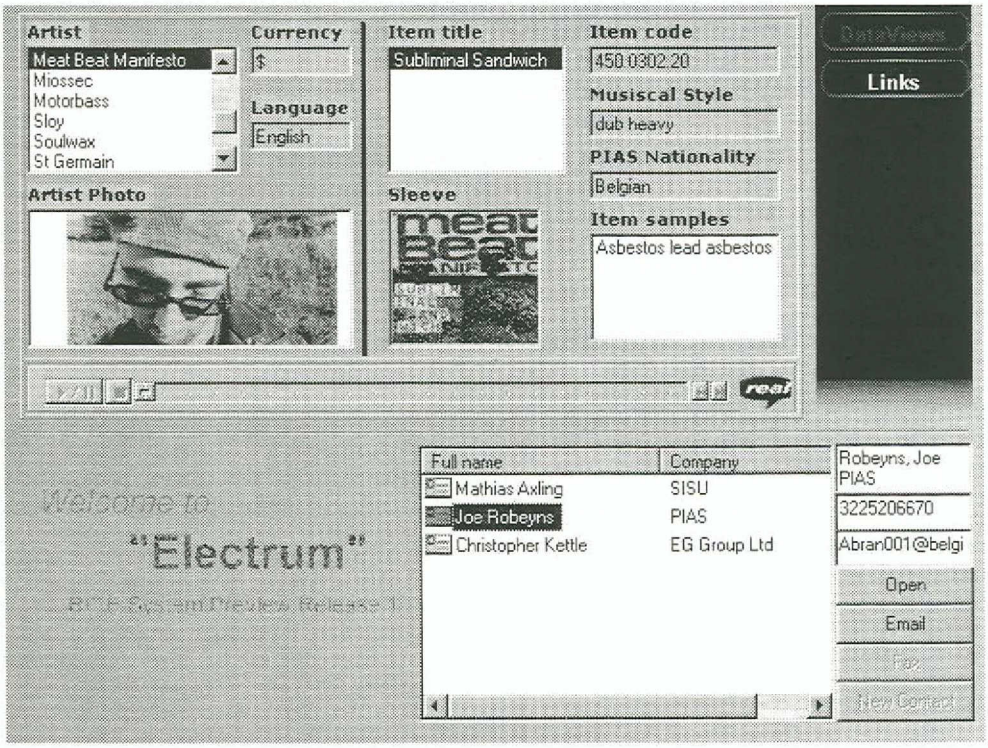


Fig. 6: The user interface consisted of three components: the artist data view, the contacts and the links components.

The user interface of the system demonstrator (shown in Fig. 6) consisted of three components,

- the artist data view, representing a simplified database interface
- the contact component, corresponding the a subset of PIM functions
- and the links component, representing a window to the Internet

In order to evaluate all components in a structured way, tasks were selected that would makes sense to the user population, ending up with fifteen different tasks. They were composed to a scenario, numbered and put together in a list. The tasks, shown in Fig. 7, involves database operations with artists that are well known to the users, communication of artist and item data and operations on links that are relevant to the music business.

1.Start RITE
2.Show information about the artist Miossec. In what language do they sing?
3.Show information from the Miossec item titled Baiser.
4.Change the information on the Miossec item Baiser, so that music style is "rock" instead of "chanson rock"
5.Play the sample called "La Fidelite" from the Miossec item Baiser.
6.Play the sample called "Asbestos lead asbestos" from the Meat Beat Manifesto item "Subliminal sandwich"
7.Find out what phone number Joe Robeyns has.
8.Send Joe an e-mail and attach some Miossec pictures to it.
9.Check out, from Mathias Axlings web site, what phone number he has.
10.Add an artist to the list and enter the appropriate information. (E.g. The Rite Project, English, £). Verify that it has been added.
11.Create an item for the artist you've just added and enter the appropriate information. (E.g. Logo, 450.1122.20, rock, UK)
12.Add a sleeve for the item you've just added (e.g. the file "Logo" from the desktop)

13.Add a sample to the item you've just added (e.g. the file called "thankyou" from the desktop) and give it a sample name.
14.Play the sample you've just added.
15.Use the RITE system links to go to "the Ultimate Band List".

Fig. 7: The tasks used during the co-operative evaluation.

2.3 Testing the Evaluation

In order to try out the evaluation methodology and the tasks we ran through the evaluation procedure once before meeting with the test users. During the run-though one of our colleagues acted as user. The evaluation run-through turned out to be a good idea, since we noticed that some of the tasks could be explained in a clearer manner. Also, we had a chance to discuss who should do and say what during the tests, i.e. who would give an introduction, who would take notes, who would act as the system expert during the evaluation. Thus, the results of this run through were a better and clearer task list, and good rehearsal for us.

3 PERFORMING THE EVALUATION

3.1 The Users

The users we met with came from different departments within the record company: management, IT department, sales department, marketing department, international department, label department and promotions department. They worked with different tasks: artistic or creative work, promotion and exploitation, IT support and management. Hence, they all had their own perspective on how they could make use of a system such as RITE, and what functionality that was the most important for us to implement.

The users were familiar with PCs and Windows 95 and they used them in their day-to day work. The main uses of the computers were searching the artist catalogue database, word processing, spreadsheet calculations and e-mail communication. Internet browsing was not widely used, mainly because it was only available on a few computers within the company.

Management had told us that they wanted to see the system first and then we were to follow a user scheme that was set up by the record company. As we went along with the tests, the list of people was modified. Initially we had planned to meet with eleven people. When we left, seventeen people, from the organisation had taken part of the co-operative evaluations of the demo. Still, we had got as much feedback from the users on our ideas that we could possibly get.

3.2 The Circumstances

The evaluation took place in a meeting room at the record company. Since this room served as smoking room and video room, people not participating in the evaluation would pop in every now and then. This happened even when we were evaluating the demo. Sometimes they commented on us and on what we were doing. However, the evaluation participants did not seem bothered by the other people. We had the impression that most users were very relaxed and focused on the evaluation.

Normally, during a co-operative evaluation there would be one or two system developers and one end user participating. However, during these evaluations we had to be more flexible. Sometimes there would be three users evaluating the demo, sometimes two and sometimes one. It all had its good and bad implications on the evaluations. When there were only one user, he or she would "think aloud" during the tasks, and when there were two or more users they could discuss the tasks among each other. The latter situation was probably more natural to them, and it was enjoyable for us to hear them discussing. Sometimes you could hear three languages at a time; French, Dutch and English. Also, the discussions that were following the tasks were somewhat more vivid when there were more than one user participating.

3.3 The Evaluation Procedure

The evaluations followed the same pattern, and each session took about an hour. At first we told the user(s) briefly about the RITE project and about who we were. Then we explained to them that we needed feedback to continue the development of the demo, and that we needed to learn more about the music business in general and of their work in particular. We also made clear that it was the *system* we were going to evaluate, not the *users*.

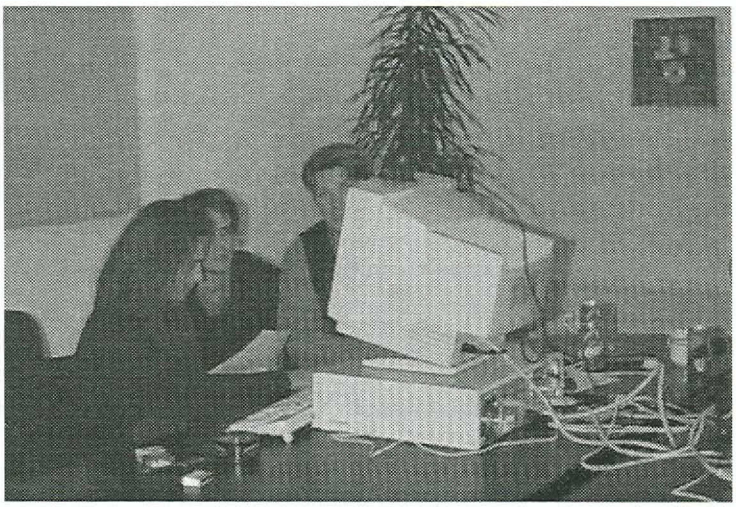


Fig. 8: During the tasks we carefully watched the user and system interaction.

Then we asked the users about their work. They told us what their main tasks within the organisation were and what their main work concepts were, on a day-to-day basis. This was very helpful for our understanding of how RITE could fit into their daily work. They all gave us one piece of the record company puzzle.

We then explained the basic idea behind the demo, that each user was supposed to get a specific user interface made up of a specific set of components. We told them that the interaction with RITE should follow the same interaction style as MS Windows (otherwise we had done something wrong) and that it should be very easy to use. Before handling them the task list, we went through the main interaction principles of the demo i.e. showed them what you could do with it. The users seemed to get a grip of the system quite fast and most of them seemed eager to start with the tasks.

The users would then go on with the tasks, one at a time. During the tasks we carefully watched the user and system interaction, as shown in Fig. 8. This was simplified when the users "thought aloud" or discussed the task solutions among each other. One of us served as a "living manual" to the system, and the users could ask him if they got stuck with a task. We sometimes would interrupt the users with questions, in order to better follow their thoughts, e.g. "What do you think now?", "Are you expecting something from the system when you click there?" etc.

When the users had finished the tasks, a discussion would naturally follow, see Fig. 9. We asked them what they thought of the concepts in the demo. What was good about the system? What was bad? How did it fit with their daily work? What was missing in the system? Usually they told us quite freely about their thoughts, and we got many valuable comments.

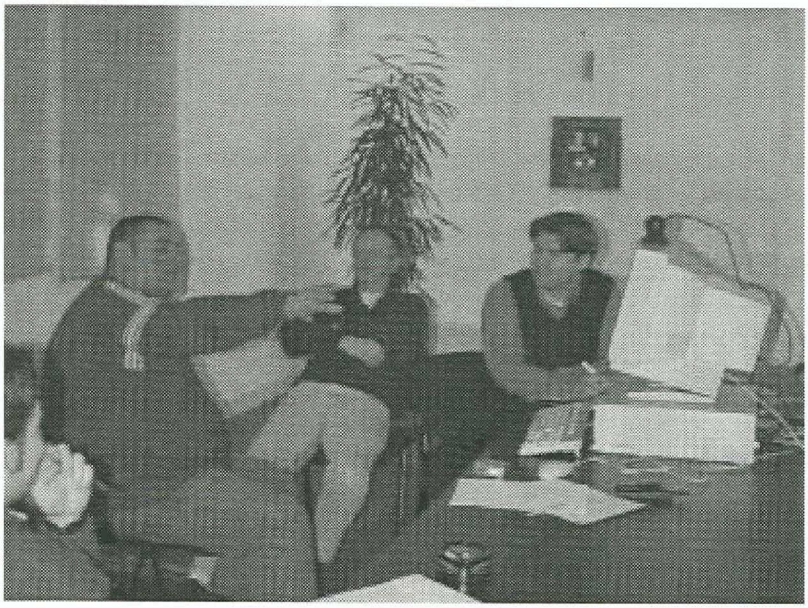


Fig. 9: When we had finished the tasks, a discussion would naturally follow.

Finally we presented a number of new ideas for the system that we needed to test on the users. They commented on its use and its importance compared to the functionality in the demo. Examples of such new ideas were the ability to share a project workspace with a number of internal or external users and the ability to drag and drop artist and item data from the database to the desktop thereby creating an "artist" or "item" file. The file would contain the data selected from the database and include easy presentation of its content. The new ideas we proposed were good as a starting point for the discussions of what functionality to focus on in the further development.

4 INITIAL RITE EVALUATION RESULTS

As mentioned, the evaluation generated a lot of feedback on the system demonstrator. Some of the information is hard to put into wording, but an attempt is made to summarise our impressions from the evaluation. The results have been characterised in four groups – general user comments, interaction issues, multimedia issues and issues specific to this particular music business and its work processes.

4.1 General User Comments

The general impression we got from the evaluation sessions was that the users liked what they saw. Most of them seemed to enjoy using the system, and told us that it was a potential good product. Some users pointed out features they wanted us to look more into, while others said they wanted the system "tomorrow, tomorrow". It was clear to us that people who work with promotion and exploitation of artists are the ones who could make the most use of the components that we showed them in the demonstrator. They use the business concepts of *artist* and *item data* in their daily work, and they were also the ones who liked the system the most. However, other components can be developed that better suit the other users, e.g. components with sales and stock data, or components designed for the artistic people.

The discussions that followed the tasks quite often involved areas of how to implement procedures around the use of RITE within the organisation. Example of such topics were; "Who is going to support the system with data?", "Who should be able to change data?" or "Will we all have Internet?". It seemed important to be able to allow certain features, e.g. updating the database, to a restricted part of the organisation.

Another topic that was often brought up was how the system would fit with their external partners and contacts. Questions such as "can my external contacts use data from it?", "How can I use this with my customers?", were raised by most user groups. As the music business is a business with a lot of contacts, external access is important to consider in the future. However, the demonstrator is well suited for both external and internal use since it is based on Internet standards and runs in a web browser.

4.2 Interaction Issues

The user and system interaction seemed to work fine. As the users were all using Windows 95 on a daily basis, they easily understood the interaction principles of the demonstrator. The tasks they were to solve involved pointing and clicking, dragging and dropping and the use of pop-up menus. Interestingly enough, all of the users solved all of the tasks. Sometimes we helped them a little bit, but the first time you use a system it is natural that you need some guidance. Anyway, when we asked the users they characterised the system as easy to use.

Finding the right functions was probably the most problematic part of the system. The functions available of a pop-up menu is not visible until you right-click, why it is rather hard to find compared to a button that is visible all the time. Many of our users were clicking around on the interface to solve some of the tasks, in search for the commands. We also observed some problems when the users were to edit data in the database. The way to do this in the demo was to bring up a dialog and edit the data fields in the dialog. However, almost all users had problems with this task and the dialog way didn't seem natural to them.

We also noticed a number of minor interaction problems, such as shortcut support and ability to tab between data.

4.3 Multimedia Data Issues

Some of the users saw problems in having everyone within the company accessing the multimedia in our database. According to them, it was a security risk in having all people accessing to the data that they are not allowed to have. One user said:

> *"Today if someone within the company wants a sleeve picture, he or she asks me and tells me why and then maybe I give him or her a copy. With this system I cannot control it anymore".*

This was an interesting point that shows features we may have to consider in the system. Also it shows some of the goals we had with RITE – easy access to low-cost multimedia - has some built in internal problems. Obviously some kind of access control/authorise copy mechanism is needed in the system if this is as important as the user claims.

In our database for the demo, we had low-quality pictures that we used. This does not seem enough, for people who are working with these pictures. We got comments that the quality must be better, maybe a full-screen version of the picture should be easily available even if only a thumbnail of it is shown initially in the data view.

4.4 Music Business Issues

Almost all the seventeen test users commented on the same issue in our artist data view component – the "item code" (the catalogue number in the catalogue database).

Our database was based on a datamodel designed during the first part of the project (and with other user representatives), and it turned out that the item codes mechanism had been misunderstood (despite several iterations in the previous database design). As people explained it to us it was clear that there was logic in this item numbering system, that we had not fully understood. The problem is easy to overcome (it is a matter of redesigning the database) but serves as a good illustration of how frustrating a system may appear to end users if it does not support their work concepts and procedures. It is worth noting that this design flaw was related to the contents of a specific database and not to the actual software design of the demonstrator, although it had serious implications for the users perception of the system.

By finding a problem as early as in the demonstrator, the costs of correcting it is much less than finding it later on when more work has been done on e.g. the components and the system help documentation.

Some of our users had recently seen the Liquid Audio system and wondered what the difference was between Liquid Audio, Real Player and the RITE system. We explained to them that RITE contains so much more than Liquid Audio and Real Player, who mainly are digital music players. RITE can include Liquid Audio if you want, and today we already use a Real Player component to play our music samples. We can easily plug any music player into RITE – with almost no coding required.

The database in our demo system is fairly small, and even though it has been designed together with people from the record company it needs some modifications. Some of the data we stored in the database did not seem important any more, and may be replaced by other data fields in further on, e.g. release texts and bar codes. Some users foresaw a search need as the database grew to the size of the catalogue database today used within the record company. A database storing all the titles in a record company's catalogue will obviously need more sophisticated search facilities. The Intuitive visual maps (that will be one of the additional components in RITE) provide very powerful search facilities, and is therefor very useful when searching big databases.

5 CONCLUSIONS

5.1 The Value of User Driven Design

This first co-operative evaluation gave us, the designers, valuable knowledge of these users work situation, including issues related to main business concepts (e.g., *artists* and *item content*) and processes. It also added to our understanding of the music business in general. We were shown around the record company and met with a lot of people from different departments within the record company. Just being in the environment of this company gave us valuable impressions for the further development of RITE. The users also have a better understanding of the possibilities of the technology and the capabilities of the designers.

During our stay, people tried to explain key issues regarding what a computer system could do for facilitating their work. We got many interesting and valuable comments, from different perspectives. We know have a far better understanding of the problems involved with designing an IT system for this target group than we had before the evaluation.

In this type of workshop like evaluation there is always a risk of not being able to capture all the relevant views and impressions. However, a very concrete result of this exercise is a list of points for improvement which was compiled by the users themselves immediately after the test.

5.2 Are we on the rite track?

Even if not everything was perfect in the demo (it could impossibly be), the general impression we got was that most of the users liked what they saw. All of the users managed to solve the tasks we had selected, after a ten minutes introduction to the system. And far more important, we now know there is need for a system for managing and communicating multimedia data in an easy to use way. Thus, the development goals of the RITE project are still valid.

The flexible design approach we have chosen by using components and dynamic HTML, seems to suit this user group very well. All users do not use or work with the same data in their daily work, so the personal interface idea to a central database (where each user interface is made up of a unique set of components) seems very useful

5.3 From Data Communication to Multimedia Presentations

Meeting with so many people and discussing the demonstrator gives you many new perspectives and reactions on the system. Each user group has its own needs with a system and it is hard to encapsulate everybody's needs in a system. However, many of the comments we got on the demo pointed in somewhat similar directions. One of the most important features that we concluded after the demonstrator evaluation was the need to easily make and communicate multimedia presentations, both internally and externally.

It seems that a lot of the work being done in such an organisation is focused around presentations, such as newsletter, and the communication of such to external contacts, e.g. the new releases newsletter that is sent to all customers. We believe there is a tremendous amount of efficiency to be made in simplifying this work. Many users said they wanted support to easy import these data to a newsletter and to store the newsletter data in the database, such as bar codes associated with an item. Others wanted support for easy publishing of the multimedia data on the Internet, i.e. to create WWW newsletters or web sites. Yet others wanted to be able to mail newsletters directly to external contacts,

instead of as today when the newsletter documents are sent by mail or fax.

One user stated the need to be able to "dump" the new releases in the database to a portable computer that the sales people could bring to the customers. Also, many of the users liked the idea of dragging data from the database thereby creating e.g. an "artist" presentation file. When we discussed these topics with the users, we soon ended up discussing communication of such "artist files" and the demands on the computer that receives such files. Will special software have to be installed on the client to see the presentation of the file? Again, since the RITE system is based on Internet standards we can make such "artist files" viewable in a browser, without installing any software on the receiving machine.

Communication of music multimedia presentations involves security and IPR issues. It is not only good that this kind of data can be communicated easily. Access rights restrictions, both for internal and external use, must be considered when developing such features. Remember the data "ownership" idea, that one user told us about during the evaluation. In order to get a copy of a sleeve, people within the record company must seek permissions from the owner to get a copy. This is a good illustration of the security needs that has to be considered when dealing with such sensitive multimedia data that a record company deals with.

5.4 Considerations for the Future

The demonstrator shown in this test was intended to illustrate the potential of the revised RITE system. For the continued work there are two, partly conflicting, aspects that must be tackled. That of general usability of software components and combinations thereof, and that of providing the most useful tool for a specific user group.

The first aspect implies that the project should continue to provide simplified user interfaces to do away with the aeroplane cockpit like user interfaces of modern PC software. The other aspect deals with how the RITE-project should focus to provide tailored functionality to its users. The essence of this is to make sure that RITE as a software tool can preserve the prevailing work culture, as it is perceived in the independent music sector. This work culture includes strong elements of artistic creativity and processes are often ad hoc or in many cases "chaotic"(in the positive sense of the word) as pointed out by the projects user partners. One implication of this is that software tools should be designed with a focus on maintaining "relations" between people rather than "structure" and "data flows" between systems. It is doubtful whether traditional so called enterprise-wide groupware solutions or generic web services can live up to this.

The next step for the project is to scale up user evaluations based on the design improvements resulting from this first evaluation. Here, a careful balance must be sought between the introduction new functions and the improvement of existing ones.

6 REFERENCES

MONK, A., WRIGHT, P., HABER, J. & DAVENPORT, L. 1993. *Improving your Human-Computer Interface - a practical technique*. Prentice Hall. ISBN 0-13-010034-X.