A Help Desk tool for sporadically occurring inquiries. Based on Microsoft Outlook 2000 and Computas GTA.

Postgraduate thesis at
Agder College, Grimstad
Faculty of engineering,
Information and
Communication Technology

by
Bjørnar Sigve Bjørge
for Computas AS

Lysaker, 2007.06.03

1 Executive summary

A help desk is intended to support inquiries from users of one or several systems. The help desk could act as an external service point to customers, just as well as an employee-based service. Because of the movement towards more complex computer environments, support departments need to be more flexible and more proactive in handling service requests. The primary goals of a support department are to increase the quality of service and to lower the cost needed to provide it. The combination of more complex systems to support and the continual demand of lowering the Total Cost of Ownership (TCO), support the fact that help desk departments need to increase their efficiency.

Today a wide range of help desk software exits. They support help desk functions such as customer tracking, product problem tracking and statistical reporting. Common for these systems is that they are intended to support inquiries with relatively equal character. Small and medium-sized enterprises (SME) often find these systems too complex and tend to use ad-hoc solutions in conjunction with telephone and e-mail.

New help desk solutions have lately been introduced based on web-technology and GroupWare. Even though developers relatively easily can implement such systems, they do not tend to work effectively within the organization. This arises from a combination of how the help desk is managed and application design limitations that affect how work processes are performed. To actually improve how the help desk personnel are working, knowledge based work process support can be used. A work process support system guides the user through execution of a task and records all actions. Thus, helping the user to perform tasks in an effective and correct manner by reflecting existing rules and procedures incorporated in the system.

This postgraduate thesis focuses on a help desk tool that provides knowledge based work process support for Small to Medium-sized Enterprises (SME) with non-dedicated support personnel. The main issue is that in SME, help desk personnel do not have the quantity and the repetitiveness of large enterprises with an established help desk/call center function. For SME work process support enables non-dedicated support personnel to perform Help Desk related tasks. The help desk function then becomes a more integral part of the organization, and improves efficiency because of a more structured way of handling end-users by focusing on and standardizing the workflow through the organization.

The postgraduate thesis has the following main objectives:

Theoretical analysis

- Identify requirements and management areas that will enhance Help Desk success in IT-organizations, targeted at Small to Medium-sized Enterprises (SME) with non-dedicated support personnel.
- Establish structured Help Desk work processes in Computas AS.
- Comparison of Computas GTA and Microsoft Exchange.

Practical development

- Design a Help Desk application (prototype) based on Computas GTA and Microsoft Outlook.
- Design work processes in GTA, using the Sara Expert Workflow tool.

Table of contents:

Table of figures:

2 Computas AS

Computas was established in 1985, and is one of the leading knowledge engineering companies in Scandinavia [Appendix D8]. The company offers a wide range of products and services for computer-supported knowledge management, using state-of-the-art tools and techniques from the following fields:

- Artificial intelligence
- Object technology
- User interaction

Current customers include large industrial enterprises, government agencies and international organizations. Today Computas has 140 employees. Det Norske Veritas, TurnIT AB and employees own Computas AS. The Computas offices are located in Lysaker, and 40 employees are situated at Det Norske Veritas offices in Sandvika. Computas' mission is to help organizations make better use of their knowledge.

Computas AS
Vollsveien 9
Postbox 482
1327 Lysaker

Telephone: +47 67 83 10 00
Fax: +47 67 83 10 01

http://www.computas.com

3 Methodology

3.1 Quality Function Deployment (QFD)

QFD, also known as "House of quality" is a technique that was useful as a starting point to identify requirements, demanders and means used to meet requirements for a new Help Desk application. For more information about QFD see: link1 and link2

3.2 Activity Diagrams

Activity diagrams is a commonly known method, suitable for describing workflow. It is used to describe the main steps in handling a Help Desk inquiry in Computas AS (High-level work process). For more information about Activity Diagrams see: link

3.3 Interviews

To analyze the current situation and existing routines in Computas AS, interviews with Help Desk personnel have been done. To establish GTA work processes, I used a combination of reading internal documents and interviewing knowledge engineers in Computas AS.

3.4 Literature study

Different literature on the subject "Help Desk" was studied to be able to illustrate Computas' situation in the context of known theory. This literature was targeted at both Help Desk managers and developers and illustrated different aspect of the role of the Help Desk in an organization. A completed project investigating key factors in Help Desk success done by the British Library and Development Department, was found very useful. A paper, "The role of help desk in the strategic management of information systems", drawn on this research project can be found at: link

4 Theoretical analysis

4.1 Introduction: definition

Before going on with further analysis, a definition of the term help desk would be in its place. *"A help desk is an accessible service point which will provide on-demand service, information or action to aid the user carrying out an IT-related task", [11].* The help desk definition says nothing about how the Help Desk function is related to other macro services in the organization. This "accessible service point" can be a standalone application or integrated with other services in the organization. Help Desk relations to other business processes are covered in chapter 4.5.

A Help Desk can be further characterized by a number of different attributes [12].

- Internal vs. External
- 1.st line vs. 2.nd level
- Dedicated vs. non-dedicated staff
- Technical vs. Service desk (isolated vs. integrated)

The users of the service could either be internal (employees), external (customers) or both. An external help desk may often mean a better resource base, since it has a more apparent relationship to business and revenues. It would also lead to a greater focus upon marketing the service. This indicates that if the service is purely internal, more work is needed to gather financial and organizational backing to justify the Help Desk.

Next is the degree of 1.st line resolution versus 2.nd level problem solving. A Help Desk could just provide one level of support. That is, taking care of all inquiries. In other organizations 2.nd level problem solving is introduced to take care of complex tasks escalated from the 1.st level.

The Help Desk staff either works on the help service on a rotational basis while performing other duties, or they are working on a full time basis.

The main purpose of the Help Desk could be a purely technical service, or a more general service to the organization. A purely technical service has an evident mission, thereby easier to manage. A "complete" integrated service desk will require more communication and collaboration within the organization to achieve its goals. However, this could be the critical difference between "old style" and "new style" Help Desks.

Although the Help Desk has different characterizations and services, they all provide end-user support. A definition of end-user support is stated in [3]. *"A specialist function which retains, on behalf of the company's population, technical knowledge about IT and the way the company uses it, in order to deliver that knowledge in an focused form to solve specific technical and business problems on both a reactive and* *proactive basis, such that the user productivity is enhanced, thereby further enabling the user to contribute to the company's goals".*

As a starting point when developing a successful Help Desk, the following elements should be defined [6].

- The Client Profile
- Help Desk Mission
- Help Desk Objectives
- Help Desk Services

First one has to develop a client profile to know what the service has to support. A mission should then be stated to declare how the staff is going to work. The mission statement should accompany a formal service-level agreement. I.e. agreements on the level of service to be provided by the Help Desk. Areas to take into consideration when formulating the Help Desk mission include [6]:

- Resolve end users' technology problems through a single point of contact within a specified time period.
- Monitor, maintain and update overall performance, including change management, security and fault tracking.
- Prevent loss of data. Every inquiry has to be saved and all resolved tasks made backup of.

Help desk objectives is set to have something to work towards, i.e. to improve or change the Help Desk. The basic objective would then be to fulfill the Help Desk mission and support the evolving IT-organization. Finally, the Help Desk services must be brought into operation.

4.2 Help Desk Development: Current technologies and products

There exist a wide range of Help Desk systems that basically consist of query logging and tracking. Recently new systems based on web-technology and GroupWare have been introduced. PC SYSCOM Scandinavia is one of the leading suppliers of Help Desk and Network management products. For a list of their "Help Desk / Call center " products, take a look at link. For a relative complete list of "Workflow and GroupWare" products, take a look at link. I have chosen to categorize Help Desk software as follows:

1. Call Center tools

This type of Help Desk software focuses on problem management and automating the call logging/tracking and escalation/routing procedures. Capabilities for making reports and monitoring performance is usually added. An example is HEAT (Help Desk Expert automation tool) from Bendata. Detailed functionality can be found at: link

2. Enterprise Help Desk solutions

Enterprise Help Desk tools start with problem management functionality and integrate it with communications tools (telephone, email or fax), asset management and system management. Problem resolution is often further enhanced using expert technology. SQL database engines and client/server technology is used.

The Remedy Corporation is one of the leading solution providers for Help Desk applications. They have integrated macro processes such as problem management and resolution, asset management, measurement and reporting into a single application. For more information about the Remedy Help Desk see: link

3. Web based Help Desk

Web based Help Desk support has lately been introduced as a part of the Internet Expansion. A Web based Help Desk enables clients to directly access a knowledge base through a Web Browser. If the answer to a specific query cannot be found in the knowledge base, clients submit the query to the support center. Then clients can follow the status of their queries online. For information about web based modules see: link

Web based modules are also used for worldwide knowledge sharing based on the "concept" described above, see: link. Such modules can be a valuable information source for the Help Desk department.

4. Remote control systems

Remote control systems have been introduced because of new technology and the expansion of "Home offices". Symantec's PcANYWHERE, is a remote control system that enables access to applications and data from a remote location (Home office). For the support department this enables one to provide support remotely. For more information about PcANYWHERE see: link

For real-time communication between the client and support personnel tools like Microsoft NetMeeting can be used. NetMeeting is a voice, video and text-based program that provides videoconferencing capabilities over the Internet. It also includes a mail extension that can be used with Microsoft Exchange and Outlook. This feature provides the ability to make a call directly from an Outlook contact item, based on Microsoft NetMeeting entries in the mail address book. For more information about NetMeeting see: link.

4.3 Motivation for using work process support

As with any software, Help Desk packages have different functionality and sophistication levels. The major functionality is listed below [12]:

- query logging
- query retrieval
- automatic routing and escalation (workflow)
- knowledge bases and expert systems for problem resolution (via web or locally)
- automatic user recognition from database
- workload distribution and analysis
- statistical analysis
- automatic notification to problem originator
- reporting
- telephone integration
- remote control functionality

However, such features are targeted at relatively large enterprises with an established Help Desk /Call center function. Small and medium sized enterprises often find these systems too complex and tend to use ad-hoc solutions. Even though SME with non-dedicated support personnel do not have the quantity of large enterprises, it does not mean less complexity. Therefore, functionality for routing and escalation to a higher "knowledge level" are highly demanded. However, for SME organization with non-dedicated support personnel, routing, escalation and tracking are not enough. One has to support everyone with knowledge about how to perform a help desk related task. Thus, the need for work process support became visible. This approach focuses first on what is to be done and how (structured processes and its activities), then routing between participants (roles and rules). In other words, knowledge based work process support reflects knowledge about procedures and instructions within a specific domain (in this case the Help Desk), providing detailed support on every action the user must take during the execution of a task. This approach goes beyond standard routing of task between participants (se figure 1).

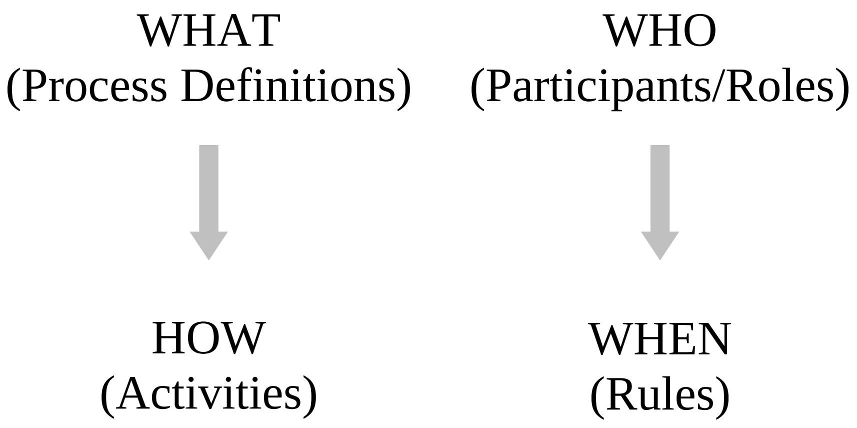


Figure 1 "What-How" vs. "Who-When"

4.4 A Case Study: The Help Desk in Computas AS

Today Computas has a central help desk service to support employees and customers. However, most customer inquires are initially received by a customer contact. The Help Desk in Computas AS is based on an Exchange Public Folder (see chapter 10). Inquiries are sent to a specific email address, and the answer provided by the support staff, is also sent to the public folder "Help Desk". The search-functionality provides a search by subject, and one can choose a more advanced search if one expresses which fields to search in. The information is not ordered systematically, which implies limited functionality for searching through the public folder. This is one conspicuous drawback of the existing solution.

Computas AS also has a CBR Express Help Desk. This is a more sophisticated application, based on Case Based Reasoning. However, the Help Desk department finds this application too complex. The most apparent drawbacks are complex methods for adding changes and the lack of an intuitive user interface. A training program is required to become proficient. In addition, using knowledge bases is mainly limited to first level (i.e. basic problems), because complex tasks often require additional work to be done before "case" reuse. Therefore, in the case of this help Desk too much work is needed to maintain the database in proportion to time saved in return with case reasoning.

4.4.1 The Client Profile

The number of staff, who may potentially use the service, is about 150 in addition to the customers. The knowledge engineers are split geographically, and about 40 users are situated at Det Norske Veritas.

Each user has a CX-PC that satisfies the minimum requirements set by the CX infrastructure manager. Each CX-PC has by default installed CX standard applications. Every knowledge engineer uses their PC for their everyday tasks. Current core working time is 9 A.M to 3 P.M, but some extra support is required to provide help in abnormal situations. All users have full access to the Internet, and e-mail is used heavily in the organization

4.4.2 Help Desk Mission

The Help Desk mission today is *"The CX Help Desk will provide a single point of contact for information, changes and problems associated with CX infrastructure".*

CX infrastructure includes all software and hardware in Computas AS.

4.4.3 Help Desk Services

The help desk personnel have to deal with a wide range of IT-products. This includes both networks and PCs. The help desk service is responsible for the following tasks:

- IT inventory management
- Installing or adjusting users' hardware/software
- Software application queries
- Hardware queries
- User training

Inquiries are received at the Help Desk either by e-mail, fax or telephone. Since inquiries arrive from different sources they may need to be forwarded to the Help Desk. The Help Desk staff solves problems as they arrive. Solutions are found either by the expertise held by the staff, or new knowledge is achieved by further study. Means used at the help desk to solve problems are staff expertise, product instruction manuals and online documentation. Problems are not assigned in accordance with any special routines. Each member of the staff observes the workload manually, and works on new tasks when they are free. This implies that there is a potential for a more effective way of assigning tasks (i.e. workload management).

4.4.4 Categorization of Help Desk Inquiries

Before describing detailed work processes in chapter 4.5, it is necessary to categorize different types of Help Desk inquiries. I have chosen to divide inquiries into two basic categories, change and help requests. (Inspired by [5]).

Change Requests:

- Changes: User generated inquiry resulting in changes to the IT infrastructure not caused by IT infrastructure failure
- Moves: User inquiry regarding the physical or virtual movement of IT resources or physical assets
- Adds: User queries on how to install or upgrade end-user IT resources
- Service request: IT and non-IT related inquiries regarding different support services

Help Requests:

- How To: User generated inquiries regarding how to operate and complete IT related tasks
- Failures: User generated inquiries caused by a problem working with an IT related task
- Access Control: Inquiries received from specific end users or groups that concern the establishing or reestablishment of access to IT resources
- Operating failure: An operating failure affecting a substantial group of end users

All Help Desk staff operations can be separated into three categories:

- Tasks completed by personnel at the Help Desk department
- Tasks completed by Help Desk personnel where the problem originated (for instance at the end-users office)
- Tasks completed by knowledge engineers others than Help Desk Staff.

4.4.5 Current problems and challenges

Based on analysis of the current situation in Computas AS, current problems and challenges (from my point of view) seem to be:

- CBR Express is too complex
- Exchange public folder, unused potential
- "Ad hoc" solutions initiated when needed, not planned for.
- Manage the greater quantity and complexity of hardware and software systems
- Manage sporadically and "unequal" inquiries. Inquiries are received from customers sporadically. That is, no special pattern of how often customers submit inquiries exists.
- Establishing effective routing mechanisms
- Establishing effective prioritization mechanisms
- Better structure of information generated through the Help Desk service, including information of who has done what, when.
- Make better use of information generated through the Help Desk service, e.g. for strategic planning of IT management

4.5 Help Desk Work Processes

When designing structured work processes, one has to understand the starting point and connections between the services that are focused on the end-user. Which services do the user initiate, and which involve the help desk department for resolution? Figure 2 shows a core set of services, which can be modeled as processes (change request, help request etc).

In Computas AS, the Help Desk has a clear relationship with other macro processes in the organization (see figure 2). The Help Desk function is a part of the macro process "Infrastructure". The process "Infrastructure" provides and runs all hardware and software. That is, work done by personnel responsible for the "Infrastructure" process affects the Help Desk and vice versa. Other macro processes e.q asset management and knowledge management also fires some events affecting the Help Desk department.

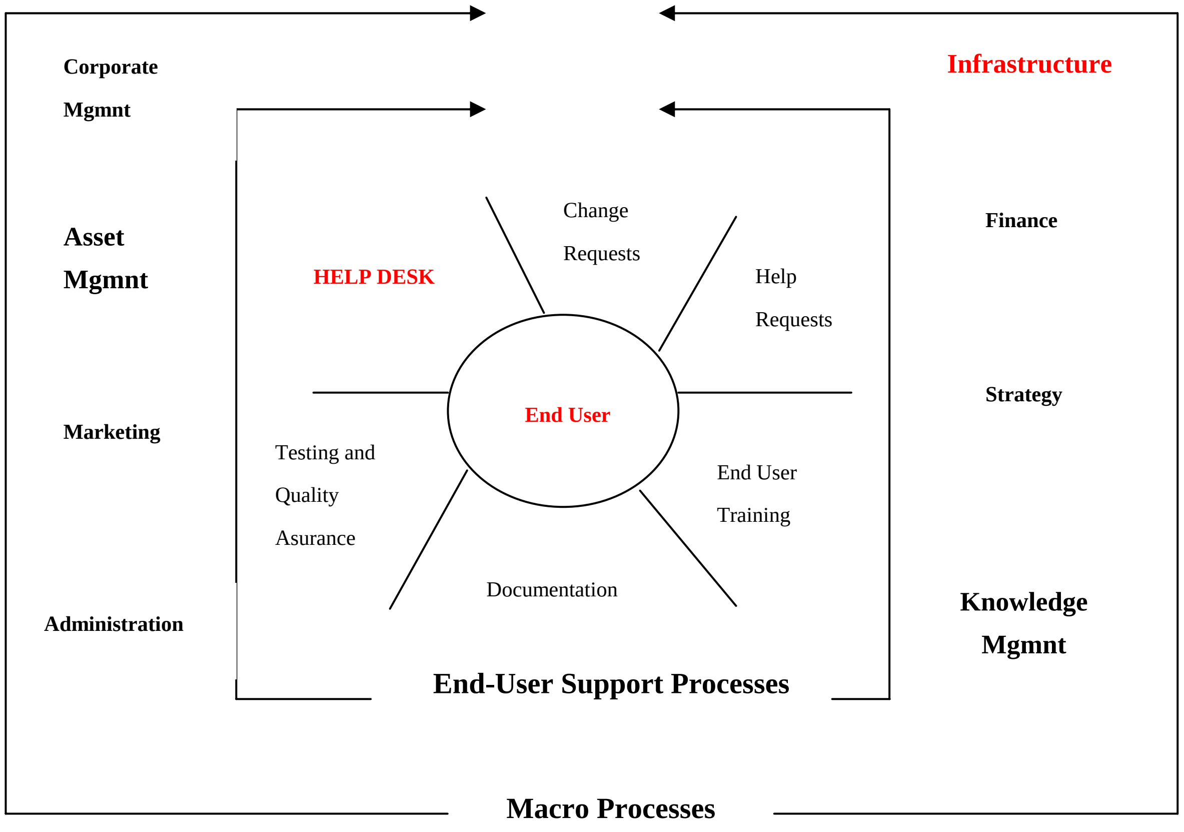


Figure 2 End-User Support Processes, inspired by [5]

The following identifies the high-level processes of the Help Desk department (inspired by [5]):

1. Problem Management and Resolution

This process is the main process since it involves resolving problems, or triggers the escalation process based on help inquiries received. If the problem is not described well enough by the user, the Help Desk personnel begins by collecting additional information from the end user to understand the problem. Next, allocation of priority is performed, also to determine the need for escalation. If the Help Desk can solve the problem, analysts attempt to resolve the problem using personnel knowledge and history information about past problems. If needed, the problem must be forwarded to another person at the Help Desk.

The problem-management and resolution process include the following main sub-processes:

- Problem identification

The problem identification phase involves information gathering and is generated by an end-user inquiry (e-mail, Phone or fax). The Help Desk analyst then identifies whether the task has to be forwarded to another knowledge engineer.

- Task escalation

Task escalation involves routing tasks to a higher level for resolution. This sub-process is targeted at passing the problem or request to the appropriate person. It is a key process, since it moves away from the initial contact point. A problem which cannot be solved by the Help Desk alone, has to be sent to another person representing the problem area, for instance, the project manager of a certain development project. Responsibility for routing this type of inquiries requiring competence not available locally, lies within the support department. Appropriate routing, such as transferring all information related to one inquiry is important because it affects the resolution process. Mechanisms that avoid situations where tasks jump from one person to another, should be established.

- Problem resolution

The problem resolution phase involves using all resources available to solve a given problem. It is required that problem solvers have access to open and resolved problems, to ensure effectiveness. Results are sent back to the end user and maintained in a knowledge base for tracking knowledge and experiences.

2. Change management

Change management involves allocation or reallocation of IT infrastructure. A change request does not require detail analysis like a problem request. The change management process mainly concerns deciding whether the change is needed, to allocate priority in relation to other tasks and to perform change request.

3. Follow-up and Closure

This is the final step when handling an inquiry in the support department. This process includes a review of the information generated to solve the problem (this could be a process itself), send feedback to the user, and a final closure of the problem or request. If the information generated to solve the problem is incomplete or the user is not satisfied, additional steps must be taken.

4. Reporting

Reporting can be triggered through several processes. It mainly involves use of acquired knowledge to report necessary information to the company, and use of technology and other resources to ensure a more proactive support department.

5. Education

Identification of training needs emerge in many situations. The Help Desk technicians can initiate their own training needs if they have limited knowledge in some specific area, or a more comprehensive training program could be organized by the Help Desk department to serve the whole organization.

6. Other Requests

These are processes that do not fall into a specific category.

The following figure illustrates the main steps in handling a Help Desk inquiry in Computas AS. This "high-level" process identifies GTA work processes described in chapter 4.5.1.

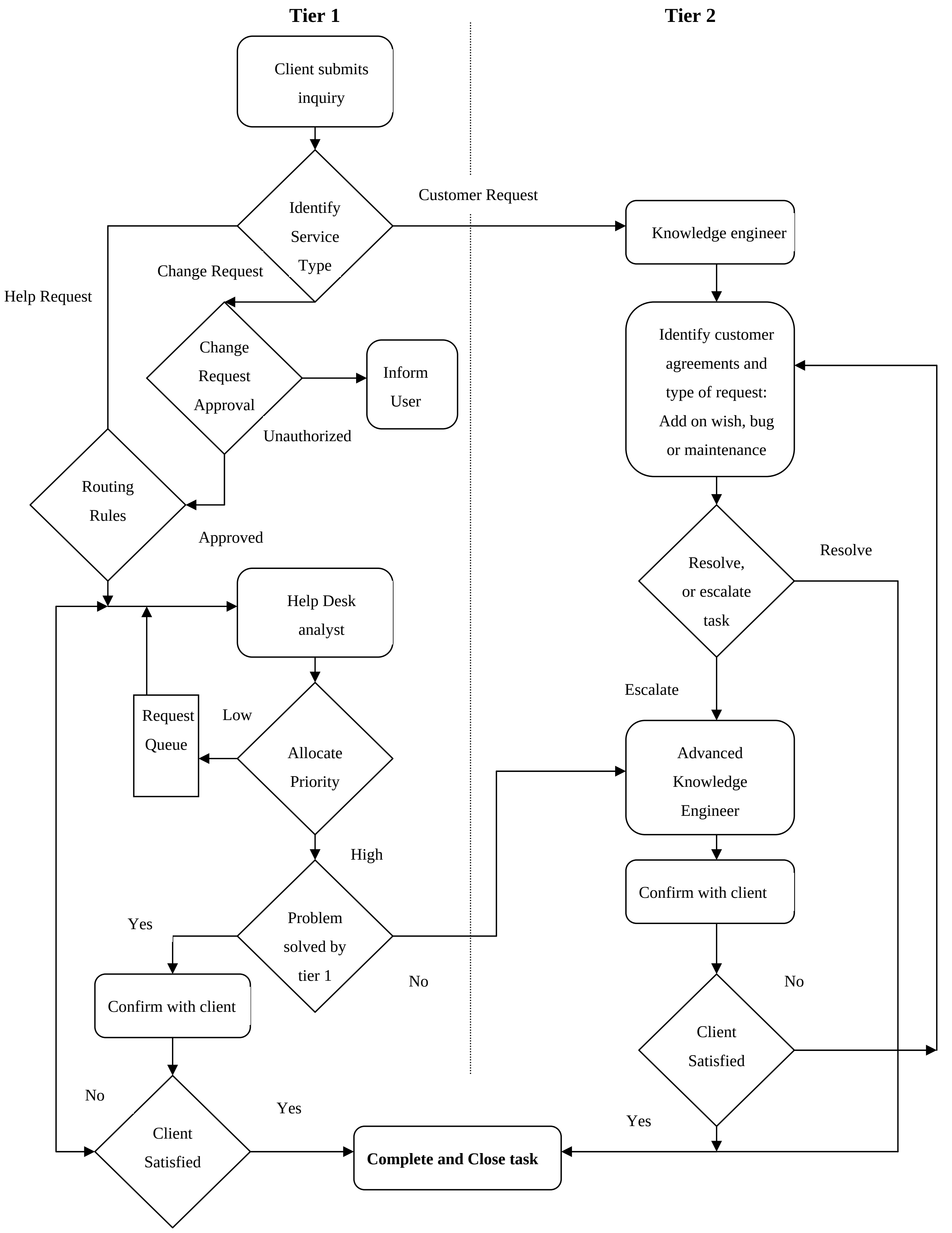


Figure 3 The major steps in handling a Help Desk inquiry in Computas AS

4.5.1 Establishing GTA Work Processes

Before continuing the path of Help Desk work process establishment a definition on the term “Work Process”, and related terms is in its place [Appendix D18].

Work Process:

A work process, is a tree of sub-processes and activities. These steps are collectively realizing a business objective. A work process has two aspects, that is the process definition and the process instance. The process definition is static, and includes definitions on process structure and behavior. The process instance is an executable instance of the process definition. It is possible to have many instances of a process definition at once. Processes can have conditions, that is information on whether they can be started, or are finished.

Sub-Process:

A subprocess is a process, that is a part of another process.

Activity:

An activity contains conditions and actions, which together define a unit of work to do. That is, all the actual work lies in the activities. These activities can either be automated, manual or a combination of both

Step:

A step could either be a process, subprocess or activity. Used as an informal name of all nodes in a process tree.

The relationship between basic workflow terminology is shown in figure 4. The Workflow management Coalition [17] has established this basic terminology. The work process definition language used in the Computas Process Framework is based on this terminology. GTA Work Processes are designed using the Sara Expert workflow tool.

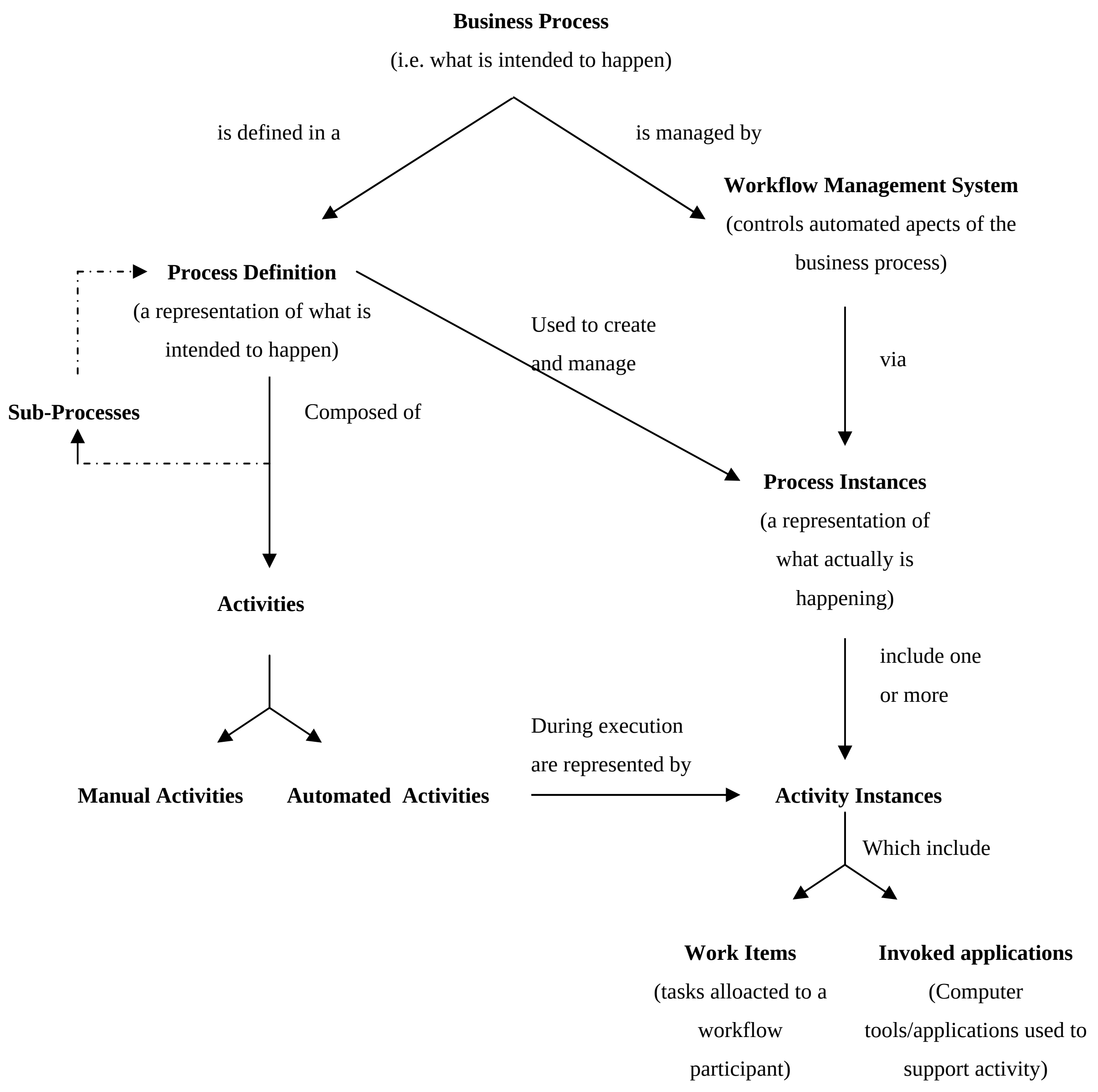

Business Process
(i.e. what is intended to happen)
is defined in a
is managed by
Workflow Management System
(controls automated apects of the business process)
Process Definition
(a representation of what is intended to happen)
Used to create and manage
via
Sub-Processes
Composed of
Process Instances
(a representation of what actually is happening)
Activities
include one or more
During execution are represented by
Manual Activities
Automated Activities
Activity Instances
Which include
Work Items
(tasks alloacted to a workflow participant)
Invoked applications
(Computer tools/applications used to support activity)

Figure 4 Basic workflow terminology and relationships

WfMC [17] defines a process as:

"The representation of a business process in a form which supports automated manipulation, such as modelling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc. The process may include both manual and workflow (automated) activities. The process definition may contain references to sub-processes, separately defined, which make up part of the overall process definition"

The following work processes assume that every inquiry is initially sent to the Help Desk. The work processes also covers management of the Assigned Help Task form. This form is a part of the Help Desk application prototype described in chapter 7.4.

1. Inquiry received, allocation of hardware – change request
2. Inquiry received, installation of software – change request
3. Inquiry received, creation of user accounts – change request
4. Inquiry received, establishment of database account – change request
5. Inquiry received form CX employee – help request
6. Self initiated, update work process – internal event
7. Inquiry received from customers– external event

The following work processes (tasks) handle the above events:

⚙ = process

✓ = executable step

➡ = mandatory step

1 ⚙	**Allocate hardware**
➡ ✓ Log allocation of hardware	You must register the hardware you have allocated. That includes PC, printer, fax, and telephone. An Excel worksheet containing all current and planned hardware allocations will be opened when you execute this task.
➡ ✓ Confirmed with user	Confirm with the user that everything is working like it's supposed to.
➡ ✓ Allocation of hardware is done	Execute this task when a final allocation of the specific hardware is done. Fill in information in the "new log entry" field and click the "Complete and Close" button.

2	**Installation of software**
Install software	Install the software according to the appropriate install description When you execute this task, a list of the available installation descriptions will "pop up". Select the appropriate installation description by clicking on the hyperlink. A word document then appears in the right frame.
Register software	Remember to register the licensed application that you have installed. This does not include internal applications. An Excel worksheet containing all current installations on your machine will be opened when you execute this task.
Confirmed with user	Confirm with the user that everything is working like it's supposed to.
Installation of software is done	Execute this task when a final installation of the specific software is done. Fill in information in the "new log entry" field and click the "Complete and Close" button.
3	**Create user accounts**
Read documentation	The documentation for establishment of a new user for any of the Computas' IT base systems is available by clicking the Hyperlink: "user accounts documentation"
Create Unix account Logged on the NIS server Home domain created Private domain created *.forward file added Password file updated User added to group: CX Aliases added	First, log on the NIS server ASLAN Second, create new home directory for user at aslan:/saturn/local/<User-ID>/. Then copy everything from /local/new.user to the users home directory using the following commands: **cd /local/<User-ID>;** **cp –r /local/new.user/*.;** The folders (including subfolders) ownership has to be changed by typing: **chown –R <User-ID> <folder/filename>;**

NIS system-files updated

chgrp –R cx<file/folder>;
chmod 0700 <file/folder>;

Third, create a new private home directory for user at aslan:/saturn/users/<User-ID>/. Then copy everything from /users/new.user to the users home directory using the following commands:
cd /users/<User-ID>;
cp –r /users/new.user/*.;
The folders (including subfolders) ownership has to be changed by typing:
chown –R <User-ID> <folder/filename>;
chgrp –R cx<file/folder>;
chmod 0700 <file/folder>;

Fourth, create a file *.forward that forwards Internet mail form the POP3 server to the Exchange server by typing:
Cd /local/<User-ID>;
vi.forward
Add the following string to the file:
<User-ID>.neptun.computas.no
If you want a local copy of the e-mail on the POP3 server add use this string instead:
<User-ID>.neptun.computas.no, <User-ID>
End the editing by typing:
<esc> :wq!

Fifth, edit the password file: etc/passwd. Add the following user entry:
<User-ID> :: User.nr : Group.nr : <full name> : local/<User-ID> : /bin/tsch
Select user-number among the free ones.

Sixth, open the group file etc/group/ and add **<User-ID>** to group CX

Seventh, add aliases by configuring /etc/aliases. Aliases for user Ola Normann would be:
O.Normann or Ola.Normann

Finally, update the NIS system-files. Go to /var/yp an type the following command: **make**

	This provide the other UNIX machines to identify the establishment of the new user.
Create NT/Exchange account User information added Memberships in groups set Profile, login script and Home directory set Exchange account created	First, start User manager for Domains and open a default user (template). Add the following information: User name: <User-ID> Name: <full name> Password and Confirm Password: <User-ID> + today's date Check off "User must Change password at Next logon" Second, set memberships in groups by clicking "Groups" Third, click the Profil dialog box. Set user profile to: **\\neptun\profiles\ <User-ID> .USR** Set login script to: default.bat and Home directory to: Connect to network drive: **T: \\neptun\users\<User-ID>** Click **OK** Finally create Exchange account: Click **Add** Registration of new users should be done at NEPTUN. A dialog box will then automatically "pop up". Fill in the following information: Name: First and last name Initials: <User-ID> Set primary Windows NT account to: **CX\<User-ID>** Click **Apply**, then **OK** Verify the following default settings: Internet: <User-ID>@computas.no X.400: c=NO:a= ;p=Computas AS; o=CX; s=Normann;g=Ola:i=on;
Confirmed with user	Confirm with the user that everything is working like it's supposed to.
All user accounts created	Execute this task when all user accounts is created. Fill in information in the "new log entry" field and click the "Complete and Close" button.

4	**Establish database account**
New user created	The following databases use CX system-database \\mimesis\databases\cx.mdw: employees, periodicals and library. To add a user in the system database select tools | Security | User and Group accounts in Access. Select new user. <User-ID> = login name (WN) <Personal-ID> = CXNNNNN Leave password blank. The user adds this on first login.
Confirmed with user	Confirm with the user that everything is working like it's supposed to.
Database account created	Execute this task when the database account is created. Fill in information in the "new log entry" field and click the "Complete and Close" button.
5	**Inquiry received from CX employee**
Task forwarded	If you don't have the expertise to solve the problem, you are responsible for forwarding the task to another knowledge engineer. Executing this task will confirm that you have sent the task to another person. This can be done by clicking the "Start Collaboration" button.
Type of inquiry identified	Identify type of Help request: How to, Failure or Access Control.
Problem description read	Second, read problem description closely. If you don't understand the problem or additional information from the user is required. Click on the "Send mail to user" button to open the correspondence mail item.

Prioritization made	Make a priority. If the problem affects a substantial group of end-user make a high priority.
Perform task	If you experience problems resolving a problem you can click the “Start Collaborate” button to start a online conversation about the problem Complete task and log results/solution in the “New log entry” field. Then click the “Complete and Close” button. A notification message is then automatically sent to the end-user.
6	**Update Work Process**
Analysis done	Refine every sub-process that must be changed caused by new working methods identified or configuration changes in the infrastructure.
Reminder sent	Send a reminder message to a knowledge engineer that manages the “Sara expert workflow” tool by clicking the “Start Collaborate” button.
Work Process updated	Fill in information in the “new log entry” field and click the “Complete and Close” button. A notification message is then automatically sent to the person who identified the need for refinement.
7	**Inquiry received from customer (e-mail, telephone or fax)**
Task forwarded	If you don’t have the expertise to solve the problem, you are responsible for forwarding the task to another knowledge engineer. Executing this task will confirm that you have sent the task to another person. This can be done by clicking the

	“Start Collaboration” button.
Type of inquiry identified Agreements read Problem description read Evaluation task assigned	For the person who receives this task (it can be yourself): First, identify type of inquiry (Change request, error message, add-on wish) Second, read the information available about the customer. (agreements) It is very important that a customer who has a change request agreement, gets an answer as quickly as possible! Third, read problem description closely. Fourth, “move” Taskreminder to a knowledge engineer for evaluation by clicking the “Start Collaboration” button.
Evaluation done	For the person who receives the task: Evaluate maintenance or change request. Log comments in the “new log entry field”. “Move” Taskreminder to a knowledge engineer for evaluation by clicking the “Start Collaboration” button.
Correction performed	For the person who receives the task: Perform Correction. Then log description of how it’s corrected (both technical and informal) in the Access database <project.mdb>. Next, Log comments in the “new log entry field”. “Move” Taskreminder to a knowledge engineer for testing by clicking the “Start Collaboration” button.
Testing done	For the person who receives the task: Test the application and log results in the Access database<project.mdb>. Next, send a notification message to the CX customer contact by clicking the “Start Collaboration” button.
Confirmed with user	Confirm with the customer that everything is working like it’s supposed to.
User Request handled	Execute this task when the user request is

handled. Fill in information in the “new log entry” field and click the “Complete and Close” button.

5 Key Factors for successful development and management of the Help Desk

The Help Desk is developing at two levels [12]:

- The "micro" level

Develop within the Help Desk department and make enhancements caused by identifications internally. Focusing on refining internal processes to enhance efficiency and enabling necessary resources to do so.

- The "macro" level

Defining the role of the Help Desk within the entire organization. Should it be integrated with other parts of the organization (change management etc)? Enterprise service desk solutions that exits on the market (Remedy), fully integrated with other business domains, is evidence of an expansion of the role of the Help Desk towards a more strategic management system.

Ensuring a "real" effective Help Desk requires both organizational and technical approaches. When establishing a Help Desk, focus is often on development, not on requirements after the Help Desk is implemented. The following process (figure 5) illustrates both development and management factors in Help Desk success. The process is illustrated as a sequence of steps, but every step is highly related to each other.

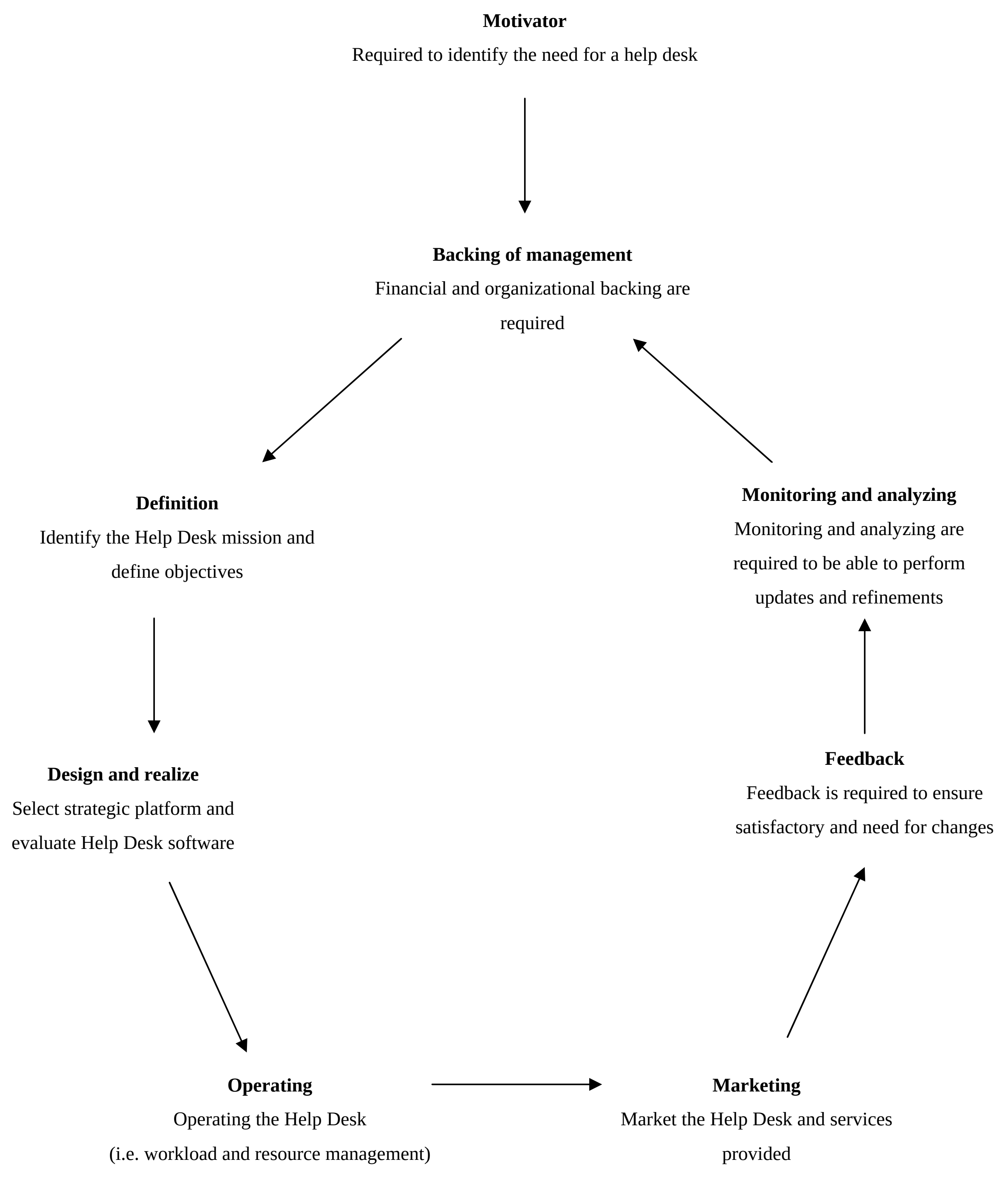

Motivator
Required to identify the need for a help desk
Backing of management
Financial and organizational backing are required
Definition
Identify the Help Desk mission and define objectives
Design and realize
Select strategic platform and evaluate Help Desk software
Operating
Operating the Help Desk (i.e. workload and resource management)
Marketing
Market the Help Desk and services provided
Feedback
Feedback is required to ensure satisfactory and need for changes
Monitoring and analyzing
Monitoring and analyzing are required to be able to perform updates and refinements

Figure 5 Key factors for successful development and management of the Help Desk

5.1 A motivator

A motivator is required in the first place to identify the need for a help desk or changes to an existing one. Generally a motivator is directly involved in IT management, thus recognizing user's problems through daily contact. The motivator in Computas AS, is Thor Nylander. He is highlighting the current Help Desk situation and the need for further development.

5.2 Backing of management

Backing of management is highly required. For the Help Desk staff to justify their resources, expenditures has to be expressed in terms of money saved and benefits delivered as a result. For SME more work is often needed to gather financial and organizational backing to justify the Help Desk, because large enterprises often have a more apparent relationship to business and revenues (i.e. customer service).

To calculate the true cost-benefits of the user support function is a necessity. How to do it, is not within the scope of this thesis. The most conspicuous support cost justifications are [16]:

- Time saving
- Increased user productivity
- Prevent disaster

A starting point of performing cost-benefit analysis of Help Desk systems is found in [16].

5.3 Definition

The definition phase is extensive. In the first instance one has to analyze end-users requirements. This should involve representatives from every level in the organization, to define the role of the Help Desk in the organization. Then the services must be defined focusing on these requirements. In a management perspective the Help Desk personnel must focus on the services that gain real value to the company. Providing too many services could cause confusion and difficulties in prioritizing. Services need to be well-defined to ensure that the Help Desk personnel are focusing on their responsibilities and the right things to do next. Well-defined services also prevent the clients from having expectations far beyond what the Help Desk can deliver. The closer these expectations are to service delivered the more satisfied will everyone in the organization be.

Then, for SME organizations the next step is to establish structured work process descriptions, so they can deal effectively with the role of support (i.e. provide the services). During this step it is crucial to involve Help Desk personnel in order to ensure that the system provides "real" support. Work process support affects the functionality needed in the next step "Design and realize".

Finally, based on information and agreements in the definition of the services the Help Desk mission and attainable objectives should be stated to have targets and to be focused. These targets are used to compare performance measured when the service is up running (see step "Monitoring and analyzing").

5.4 Design and Realize

The help desk architecture must be designed with focus on automating all help desk processes, ranging from handling end-user inquiries (i.e. problem management and resolution) to strategic management (measuring and analyzing). For SME knowledge based work process support can be used to enable use of non-dedicated support personnel. Functionality needed depends on the target organization (see chapter 4.3). However, a help desk system must be a benefit to the organization. That is, reduce cost, improve client satisfactory and increase efficiency. Evaluate Help Desk software is crucial. How to do it, is not within the scope of this thesis but information of how to evaluate Help Desk software can be found in [16].

5.5 Operating

Once the Help Desk is up running the issue is to effectively manage the workload using all resources available. In addition, accessibility is a key issue. The end-users should be able to contact the Help Desk quickly and get an answer within a specific time period.

5.5.1 Workload management

How the Help Desk department manage their workload is essential for their total performance. This again affects the financial justification for further evolvement.

5.5.1.1 Reactivity and proactivity

There are two concepts of managing a workload, called reactivity and proactivity. If the support department deal with its workload by simply reacting to situations as they appear (i.e. acting in return), it is called reactivity. On the opposite, you are working proactively, if you control the workload and "make things happen".

In a support department reactivity is essential. That is, emergencies will appear, machines will fail, and mistakes will be made. Many situations cannot be forecasted, and it would not help to be proactive.

The organization benefits from both ways of working. The company benefits by having a central contact point, which can deal with most things as they crop up. The users benefit by getting help in emergency situations (it is often not an emergency in a corporate perspective). The Help Desk staff benefits too, because the situation raises their standing for having dealt with inquiries at hand.

However, the benefits of reactivity are limited to present situations. The point of solving any given problem is lost when the same problem is solved later by another technician for a different user. The work had to be done again, and no effort was made to prevent it. The benefits of proactivity are considerable. Proactivity gains self-respect due to having control over the workload. It means that the staff must examine their jobs and search for things to do next. That is, choices between options of what to do next to make changes and real improvements. Proactivity brings benefit to the Help Desk staff, because it can inject variety into their work. Variety can be a key to solve the known "help desk problem", namely burnout. Different types of work in the support department with tangible benefits can generate renewed motivation.

Proactive management can take many forms. One way is to regularly study user problems with a view to preventing that they occur in future. Another known method is to create an inventory of user equipment to help diagnose and solve the problems. Generally proactivity is to perform actions now, in order to influence the future. Users, technology and problems change, and it impose the Help Desk to change. In a management perspective proactivity actually involves to accomplish the path illustrated in figure 5 with emphasis on Marketing, Feedback, Monitoring and analyzing.

5.5.1.2 Escalation

Escalation is the industry term for "bumping it up the line". The person neither has the expertise or the resources to solve the problem, so it is "escalated". Escalation is important when managing the workflow. The key to success with escalated tasks is to ensure clear ownership within the Help Desk department, especially when they have left the Help Desk. If ownership is unclear, more work is needed to ensure that problems are solved, not just delayed.

5.5.1.3 Prioritization

In a Help Desk department there will always be a queue of work to do. This indicates the need for prioritization. Each member of the staff must therefore use some way of deciding which task to handle next. Some would just use their natural judgment or the classic non-prioritization "first in, first out" method. Regardless of method, priorities should always take into account the impact of the user problem on business.

Regarding prioritization "methods" one should note the difference between importance and urgency (see figure 6). The confusion of these two distinct attributes could cause that a task is handled first because it is urgent (i.e. time-critical) and not because of its relative importance. The essential difference is to determine whether the situation is urgent, can only be done when the problem occurs. That is, you then have the opportunity to decide whether the problem is time-critical or not. Importance on the other hand, means that you have the opportunity to examine the task and its importance compared with another, and its relevance to the business goals.

The meaning of task prioritization is to ensure that what is most important, gets dealt with first. This can cause an urgent task to be queued, because it is not important according to business goals.

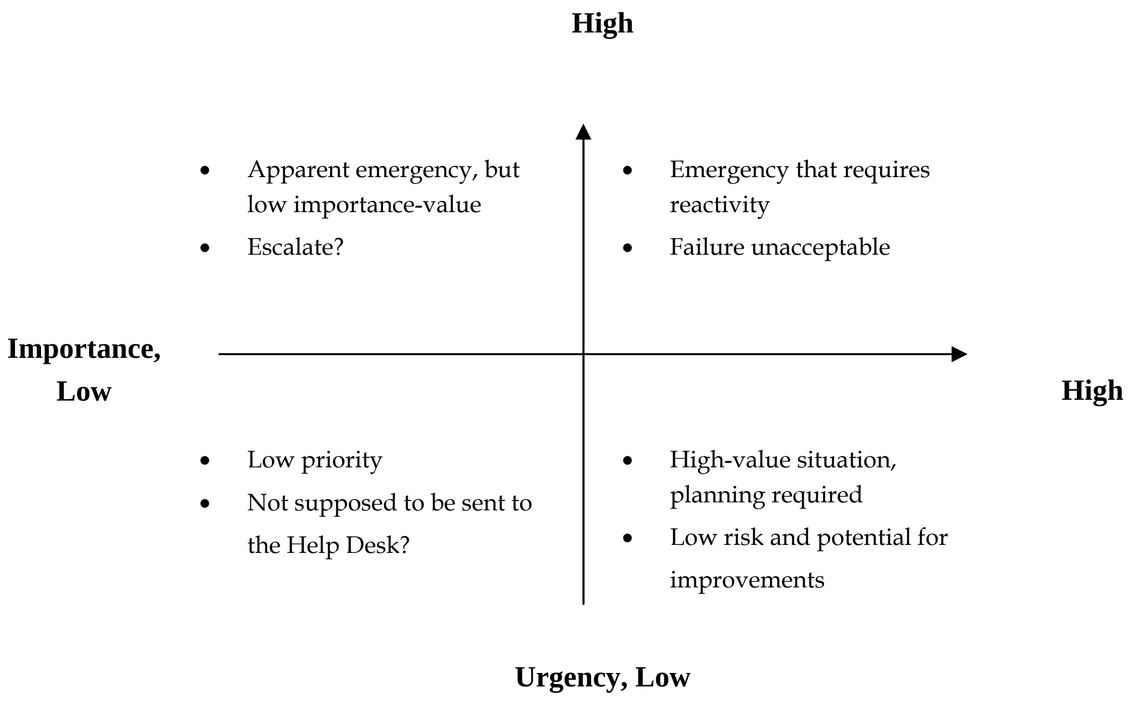

High
Apparent emergency, but low importance-value
Escalate?
Emergency that requires reactivity
Failure unacceptable
Importance, Low
High
Low priority
Not supposed to be sent to the Help Desk?
High-value situation, planning required
Low risk and potential for improvements
Urgency, Low

Figure 6 Importance versus urgency, inspired by [3]

As a starting point to prioritize one could use a simple comparison of importance and urgency as depicted in figure 6. The top priority becomes "important and urgent", second is "important and not urgent", third is "urgent and not important" and neither important nor urgent takes the lowest priority.

5.5.2 Resource management

For the Help Desk staff to carry out their work they need upgraded resources and tools. To manage their resources they also need a record of the total inventory. One way of keeping records of allocated hardware and software, is the allocation worksheet in Excel used in Computas AS. Because of the lack of registrations, another way could be to create an application that makes a full description of main components on a specific machine. On the other hand, using knowledge based work process support should ensure that registration of resources is done correct.

5.5.2.1 Knowledge

Knowledge is a key resource. The movement towards more complex systems and greater variety of systems result in increased task complexity. This put a pressure on the knowledge needed to perform support. Issues to concern [4].

- Knowledge levels

Different groups of clients need different levels of support.

- Relevant products

The Help Desk staff has to be up to date on new products.

- Relevant knowledge

To be able to provide all services, acquiring knowledge quickly is often a challenge. No single person in the Help Desk department can or is capable of acquiring and memorizing knowledge about every "problem". Therefore, an important role of the support specialist is to know how to find the solutions and distribute them to the clients. For effective knowledge retrieval and delivery, a central knowledge base plays an important role. A central repository to keep a history of each problem is required to move from the "fire-fighting" working pattern. However, using knowledge bases is mainly limited to first level (i.e. basic problems). Because, complex tasks require additional work to be done before "case" reuse. This requires assignment of responsibility of quality control and database maintenance. Without continuous maintenance of the knowledge base, the information becomes useless because of rapidly changes in the organization.

In order to ensure a usable knowledge base, work processes for capturing and maintaining knowledge are required (step "Definition"). As experienced in Computas' gathering of information should be done through work processes performed (i.e. everyday tasks).

5.6 Marketing

Next significant step is to market the service to end-users and executive managers. Establishing Service level agreements is one way of marketing the services to end-users and business managers. SLA is an agreement between the Help Desk and its "clients". A SLA specifies the Help Desk services and responsibilities. It also determines the responsibilities of the end-users using the service. The following identifies the main elements in a SLA [6]:

- Help Desk services
- Hours of operation
- Service Accessibility
- Client responsibilities
- Call priorities and response times

- Help Desk measures to be met
- Escalation procedures
- Reporting

5.7 Feedback

When the Help Desk service is up running, a significant step is to gather information about the clients' view on how the service is working. One could establish some sort of satisfactory surveys, talk with them in a more informal fashion or use information from earlier inquiries received. This information is important to identify need for changes, which is an ongoing process. [11] states: *"As a nexus for information, the help desk may ease the processes of change by maintaining continual contact between the groups concerned, insuring that plans are not executed without prior knowledge of those affected".*

5.8 Monitoring and analyzing

Monitoring and analyzing are required to ensure that the Help Desk it is meeting its objectives set in the "Definition" phase. [11] states: *"If the information which the help desk can gather is used effectively, requirements and performance may be monitored on a continual basis, making use of knowledge gained over time concerning users and systems."*

5.8.1 Analyzing service needs

The services provided by the Help Desk must be adjusted regularly as the organization changes. This could range from ordinary to a more detailed restructuring. Again it is crucial to focus on the services that deliver the best value to the organization. One could analyze the Help Desk inquires for the last month to identify user groups that seem to have the same sort of queries. Those users can be contacted to discuss solutions to provide the service they need. In addition, making reports on changes made to the computer network and how the organization will benefit from those changes enhance the possibility of satisfied clients.

For SME organizations work processes has to be refined. Because, organizational and technical changes affects the support department. Changes could either be internal or in business processes affecting support.

5.8.2 Help Desk measurement

To be able to measure the Help Desk performance, use of key statistics is a known method. These statistics measure the norms of general user support, in terms of providing information and solving tasks. [11] states: *"Statistics have a number of* *very valuable potential uses: as a source of information on the nature of problems encountered at present; to monitor usage of systems and the spread of users for each; to identify training needs; to identify gaps in provision and duplication of data input; and to assist in the mapping of the present pattern of information collection, dissemination and use within the client base."* There exist a number of measurements and I will comment the following ones [3]:

- Usage Rate
- Spot Rate
- Fix Time
- Fixes Per Resolver
- Response Time

The main one of statistic measurements is the "usage rate" (i.e. the number of times the user actually sends an inquiry to the Help Desk).

The next key statistic is the "spot rate" (i.e. the percentage of inquiries that are solved without further action). This measurement is essential, since it tells what level of immediate service the users can expect.

Another statistic is the "fix time" (i.e. how long a problem solver works before completing the task). If problems tend not to be fixed immediately it is important to find the cause. Was it too complex, or is it just delayed?

A measure of "fixes per Resolver" indicates the needed size of the staff.

Next is "response time". For some user "response time" is more important than "fix time". They just need a reassurance that somebody actually is working on the task.

6 Requirements for a new Help Desk tool

As a starting point to identify requirements, demanders and means used to meet requirements, I use Quality Function Deployment. Values 1, 3 and 9 are used to characterize the relative importance between each category. 1 = low, 3 = medium and 9 = high. It should be noted that values are assigned according to my point of view.

Requirements → **Demanders** ↓	Intuitive User-Interface	Enable use of non-dedicated support personnel	Configurable and easy to maintain	Stability and performance	Up to date information	Used by clients	Enable competitive advantages	Within price-range	Tracking functionality, who has done what when	Fit into existing work patterns
Help Desk personnel	9	9	9	9	9	9	3	1	9	9
Knowledge Engineers	9	9	9	9	9	1	3	1	9	9
Managers	9	9	3	9	9	1	3	1	9	3
Computas AS	9	9	9	9	9	9	9	9	9	3
Customers	3	3	1	9	9	1	1	1	3	1

Means used → **Requirements** ↓	Analysis of current situation	Integrate with existing applications	Use common known standards	Add work process support	Use exiting modules					
Intuitive User-Interface	9	3	3	9	1					
Enable use of non-dedicated support personnel	3	3	1	9	3					
Configurable and easy to maintain	3	3	9	1	3					
Stability and performance	1	3	9	1	3					
Up to date information	3	1	3	9	1					
Must be used by clients	9	9	3	9	1					
Enable competitive advantages	9	3	1	9	9					
Within price-range	3	9	1	1	9					
Tracking functionality, who has done what when.	9	3	1	9	1					
Fit into existing work patterns	9	9	1	9	1					

6.1 Organizational requirements

Computas AS wants to store all information about inquiries to make use of valuable information in a more systematic fashion than today. That is both inquiries initiated internally and externally. Valuable information is mainly problem description, problem solution and tracking information. A system which maintains and makes information easily available (i.e. includes an effective search-functionality), would gain the following benefits.

- Prevent information loss
- Reuse of key information
- Better view of how things are managed and control of workload
- Make reports with a view to potential of business improvement

Next, is using work processes to support the Help Desk personnel when performing tasks. A key issue is that customer inquiries are received sporadically and most of the questions have to be forwarded to a knowledge engineer responsible for a particular domain. In addition, SME like Computas AS do not afford having help desk personnel working exclusively with Help Desk inquiries. Therefore, well-documented procedures in the form of work processes will provide a more intuitive way of handling sporadically occurring tasks and enables use of non-dedicated support personnel. Having detail work processes will also improve the efficiency in the support department, and provide a more structured way of handling end-users. Another important issue is that adding new information into the system must not involve "additional" work. Collecting information as a part of the work process ensures correct logging of information.

Next is user-friendliness. Microsoft products have an important role in the organization. This implies that the staff is familiar with applications like Microsoft Outlook and supports the idea of developing a help desk tool in Microsoft Outlook.

6.2 Technical requirements

- Routing, tracking and collaboration

The Help Desk application must include functionality for routing of task to the right person and tracking of information such as who has done what when. Since most of the tasks involve several knowledge engineers, collaboration features is also required.

- Work Process support

For SME, knowledge based work process support functionality is required

- Registration of inquiry information

The following information must be saved for each inquiry received:

- Inquiry ID
- Customer name and ID
- From: Customer e-mail, CX employee e-mail.
- Computas Contact: CX Knowledge engineer.
- Received (date and time)
- Phone: Customer Phone, CX employee direct line
- Location: VV9, DNV.
- Status: Not Started, In Progress, Completed.
- Priority: Normal, Medium, Immediate deadline.
- Category: Customer support SW, Customer support MA, Internal support.
- Problem type and description
- Process (see chapter 7.4.2)
- Product
- History information (summary)

- Database

To provide searching and reporting capabilities, Outlook fields are mapped to an Access database.

- Search functionality

A key functionality is to be able to search for similar cases by subject or other fields. Outlook has a built in search function. One can either search in "Built in Fields" or "User Defined Fields" performing the following steps:

Tools | Find | Advanced Find button |Advanced Find Tab | Define more criteria: Field button |Add to list | Find now button |

7 A Help Desk tool based on Microsoft Outlook 2000 and Computas GTA.

A main part of this thesis concerns integrating the task functionality in Microsoft Outlook 2000 with GTA, an existing workflow system.

7.1 Outlook: Introduction

Microsoft Outlook is Microsoft's messaging and collaboration client. Outlook is designed as the e-mail client to Microsoft Exchange Server, but in addition to e-mail functionality, Microsoft Outlook has several management modules:

- Task management
- Calendar tool for setting appointments
- Contact management
- Journal tracking system

The Microsoft Outlook client can be used with a variety of e-mail servers and services including SMTP/POP3, IMAP or MAPI servers[1]. Outlook also works in conjunction with other message and information sources like Microsoft Mail and Fax, CompuServe, Lotus Notes and cc-mail, made possible through the MAPI extensibility interface in Outlook. However, different servers make different sets of features available. As distinct from MAPI, POP3/SMTP do not allow e-mail messages to be stored in multiple folders. Based on how companies intend to use Outlook, three main installations are available:

- No e-mail, Information management only
  using the contact, task, and schedule features without e-mail.
- Internet Only, Limited scheduling and collaboration
  using an Internet service provider for a SMTP/POP3 or IMAP server.
- Workgroup, Full e-mail, scheduling and collaboration
  using Exchange server or another third-party system and Internet mail.

For information about which feature areas supported in each configuration see [Appendix D10].

[1] Microsoft is working with third-party vendors to make their systems fully MAPI compatible

Outlook can be extended and customized using Visual Basic. This can be done in different ways, dependent on wanted functionality. Visual Basic is a programming language used to create stand-alone applications as well as DLL's to extend applications. Outlook provides the following programming opportunities:

- Visual Basic
  Create a separate application to control Outlook through its object model (This can also be done using C++). Used to develop applications that respond to events that occur in the application. (i.e. windows explorer, folders, all items)
- Visual Basic for applications (VBA)
  One can either automate Outlook from another application, or control how Outlook works within Outlook using a macro.
- Visual Basic Script
  Extend the functionality of Outlook forms. Used to develop applications that respond to events in specific items (i.e. e-mail, task, contact items). Since VB Script code is interpreted and not compiled, Outlook items are relatively small. Small in terms of bytes means less server resources consumed and faster performance. There exits no global repository of VB Script code as there is for VBA code.

The first step to understand Microsoft Outlook and its functionality is to work through the Outlook Object Model. Figure 7 shows the Outlook Object Model. At the top is the Application Object which represents the entire Microsoft Outlook Application. This application object is also used when automating Outlook. The object mainly does the following:

- Provides access to other objects through the Namespace and Folder objects
- Makes it possible to create new items using the CreateItem method, without having to traverse the object hierarchy
- Provides access to other interface objects such as Explorer, Inspector and COM AddIns

Next is the Namespace object which represents the message store. This object only contains MAPI Folders, but it could hold other folder types. This object has features that include:

- Methods for logging in and out
- Direct access to folders an items using an identification
- Access certain special default folders directly
- Access to data sources owned by other users

The MAPI Folder contains Items. An item represents a particular instance of Outlook data. It could be either an appointment, contact, journal, e-mail, note or Task-item.

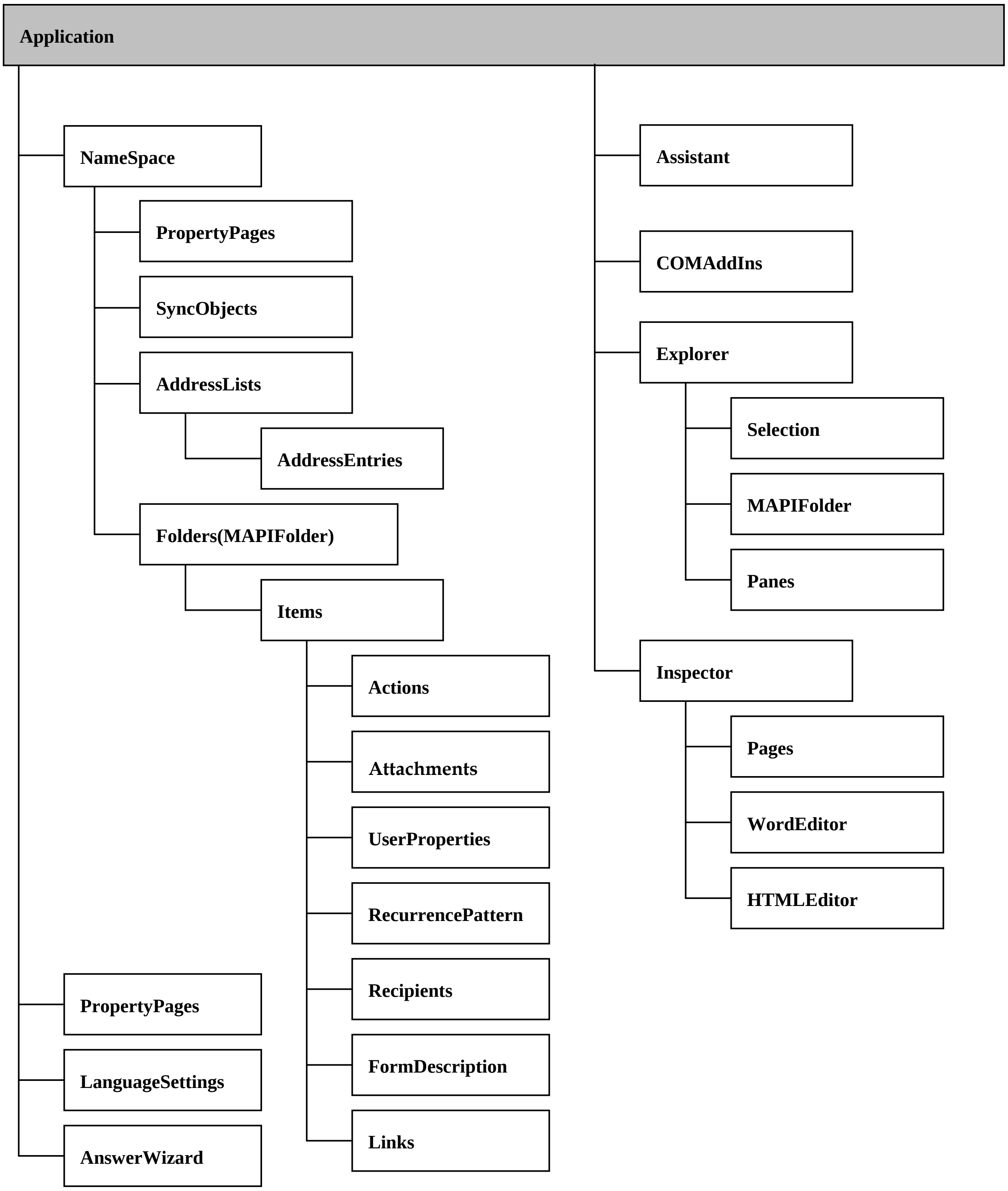
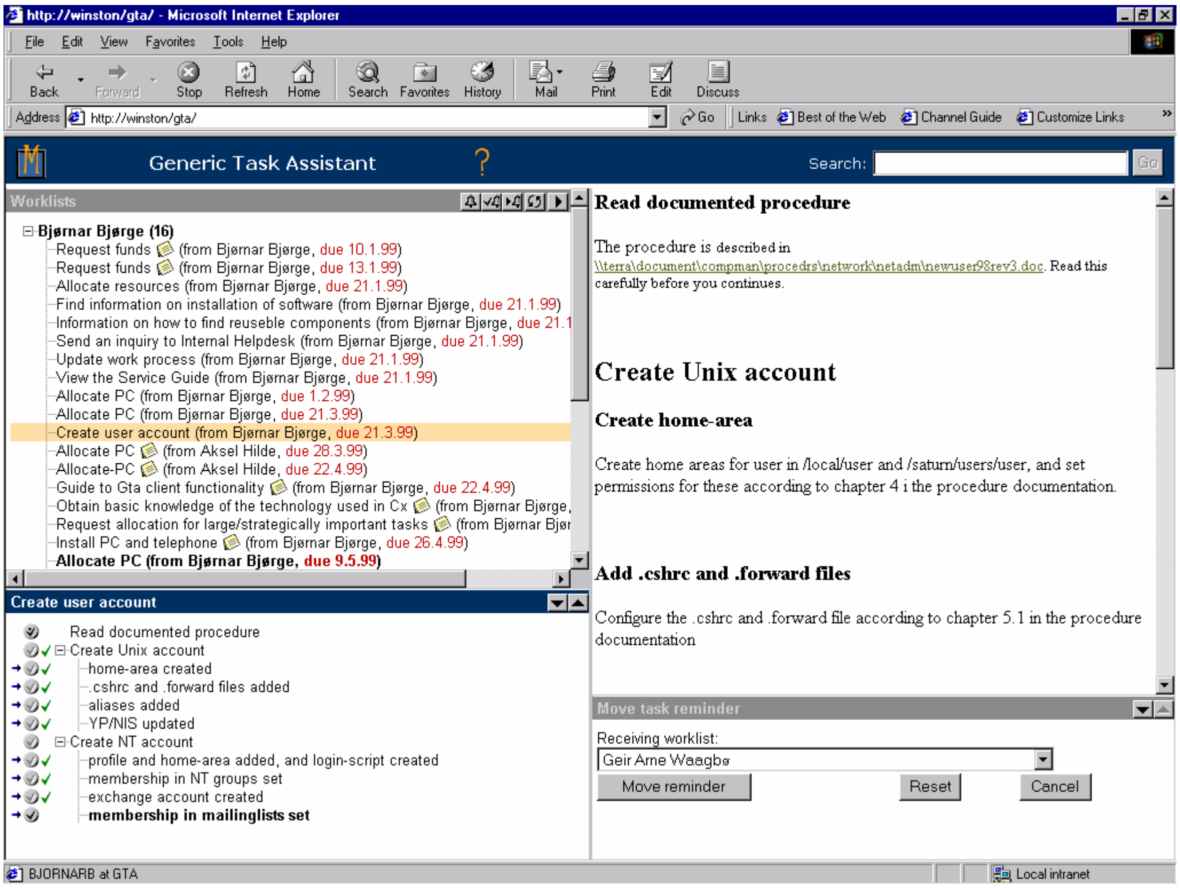


Figure 7 The Outlook Object Model [Appendix C]

7.2 GTA: Introduction

The GTA web client is a web based generic task assistant, connected to a Sara Server (see figure 10). The Sara web client framework supports two client implementations. When designing the Help Desk tool the XML client is used (see figure 8). The XML client provides capabilities for exploring and manipulating worklists and tasks. The user interface is based on XML (Extensible Markup Language) and the system architecture consists of two subsystems, the XMLExplorer and the XMLServer. The XMLExplorer runs inside a web browser and the XMLServer in context of a JavaWebServer. Data communication between the two subsystems of the XML client is based on the HTTP protocol. Every data request is initiated by an HTTP request from the XMLExplorer (i.e. the browser). A user logon request must be sent prior to any communication with the Sara Application Server. The following table [Appendix D15] gives an overview of the client to server protocol in GTA. The parameter combinations described are those relevant to the integration with Outlook (the server URL in front of each parameter combination is currently http://winston/servlet/GTAServlet).

Parameter String	**Explanation**
?type=login&username=[userid]&password=[password]	Logs a user into the server
?type=worklists	Returns the worklists and tasks of the current user. Not visible in Outlook
?type=process&process=[procID]	Result in a GUI representation of the process with the id [procID]. [procID] is a string that consist of the string `GtaProcess` and the Gemstone id of the process, e.q GtaProcess2616233
?type=createReminder&toMyself=true&processName=[Process]	Result in a task reminder sent to oneself without using the XML reminder form., e.q Create user account
?type=createOutlookReminder&toMyself=true&processName=[Process]	Same as previous. Returns HTML instead of XML for the purpose of getting the [procID] string used when type= process&process=[procID]
?type=logout	Logs the user out of the server


http://winston/gta/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://winston/gta/
Go Links Best of the Web Channel Guide Customize Links
Generic Task Assistant
Search:
Go

Worklists

Bjørnar Bjørge (16)
- Request funds (from Bjørnar Bjørge, due 10.1.99)
- Request funds (from Bjørnar Bjørge, due 13.1.99)
- Allocate resources (from Bjørnar Bjørge, due 21.1.99)
- Find information on installation of software (from Bjørnar Bjørge, due 21.1.99)
- Information on how to find reuseble components (from Bjørnar Bjørge, due 21.1
- Send an inquiry to Internal Helpdesk (from Bjørnar Bjørge, due 21.1.99)
- Update work process (from Bjørnar Bjørge, due 21.1.99)
- View the Service Guide (from Bjørnar Bjørge, due 21.1.99)
- Allocate PC (from Bjørnar Bjørge, due 1.2.99)
- Allocate PC (from Bjørnar Bjørge, due 21.3.99)
- Create user account (from Bjørnar Bjørge, due 21.3.99)
- Allocate PC (from Aksel Hilde, due 28.3.99)
- Allocate-PC (from Aksel Hilde, due 22.4.99)
- Guide to Gta client functionality (from Bjørnar Bjørge, due 22.4.99)
- Obtain basic knowledge of the technology used in Cx (from Bjørnar Bjørge,
- Request allocation for large/strategically important tasks (from Bjørnar Bjør
- Install PC and telephone (from Bjørnar Bjørge, due 26.4.99)
- **Allocate PC (from Bjørnar Bjørge, due 9.5.99)**

Create user account

- Read documented procedure
- ✓ Create Unix account
  - ✓ home-area created
  - ✓ .cshrc and .forward files added
  - ✓ aliases added
  - ✓ YP/NIS updated
- Create NT account
  - ✓ profile and home-area added, and login-script created
  - ✓ membership in NT groups set
  - ✓ exchange account created
  - **membership in mailinglists set**

Read documented procedure

The procedure is described in \\terra\document\compman\procedrs\network\netadm\newuser98rev3.doc. Read this carefully before you continues.

Create Unix account

Create home-area

Create home areas for user in /local/user and /saturn/users/user, and set permissions for these according to chapter 4 i the procedure documentation.

Add .cshrc and .forward files

Configure the .cshrc and .forward file according to chapter 5.1 in the procedure documentation

Move task reminder

Receiving worklist: Geir Arne Waagbø

Move reminder | Reset | Cancel

BJORNARB at GTA | Local intranet

Figure 8 The Computas GTA XML client

The main user functionality provided by the GTA XML client is as follows:

- Opening notes

If the task has a note string attached, the user can display the note by clicking on the note icon.

- Deleting and finishing tasks

Task can be deleted by pressing a delete button. This button is only enabled when a task is deletable.

- Loading processes

When the user selects a task with a process reference id, the process is loaded.

- Executing steps (processes and tasks)

The user can execute executable processes and tasks at any time.

- Creating and moving tasks

The user can create or move tasks to another person's worklist by entering data in a form rendered when clicking "Create taskreminder" or "Move taskreminder to another worklist".

- Search for work processes

Using a standard search engine, the user is able to search for work process definitions. By clicking on the search result link, a description of the work process appears. The user can then either click “Send to myself” (a taskreminder is sent to oneself) or “Send to someone else” and select a recipients.

7.3 Integrate Outlook and Computas GTA

The first issue in this section was to establish a setting for the task functionality in Outlook. Today Computas AS use an Exchange Public folder where users submit standard e-mail messages describing their problems. These messages are also automatically sent to the Inbox of every Help Desk staff member. After some research, I found that Microsoft have an Outlook Help Desk template available. This Help Desk application can be downloaded at no charge at Microsoft office-update web site, [Appendix D6]. This application has most of the wanted functionality, and I made changes to make it fit into Computas’ organizational and technical environment.

Second, and the main task in this section was to find an appropriate and consistent way of integrating GTA in an Outlook task item. After some research I found that the best way to do this, was to integrate an Internet Explorer browser and using its properties to control the XML/DHTML client of GTA. A detail overview of the OLE–based (Object Linking and Embedding) Internet Explorer architecture is shown in figure 9.

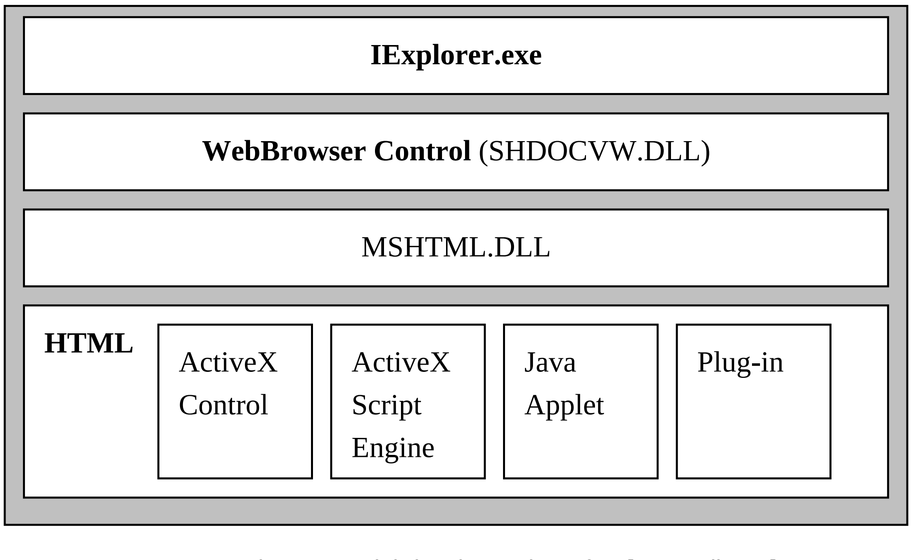


Figure 9 High-level overview of IE [Appendix D7]

IExplorer.exe is the application that's initiated when the Internet Explorer browser is loaded. It uses Internet Explorer components to perform navigation, history maintenance and HTML parsing and rendering. As shown in figure 9 IExplorer.exe hosts the shdocvw.dll component, which is, named as the WebBrowser control. Shdocvw.dll provides the functionality associated with navigation, in-place linking and history management. Shdocvw.dll in turn hosts the mshtml.dll component, which performs the HTML parsing and rendering in Internet Explorer. It also exposes HTML documents through the dynamic HTML object model. Mshtml.dll hosts the script engines, ActiveX controls, Plug-ins and other objects that might be a part of an HTML document. Shdocvw.dll also provides interfaces to its host to allow it to be hosted as a separate ActiveX Control. The WebBrowser Control is used in this way to control the GTA XML client as shown in figure 10. This GTA client is modified since Outlook handles the inbox (worklist). That is, the worklist frame in the original GTA XML client is removed. Thus the remaining functionality is rendering and execution of processes and tasks. However, it is still possible to search for GTA work process definitions. For a full description of GTA client functionality see chapter 7.2.

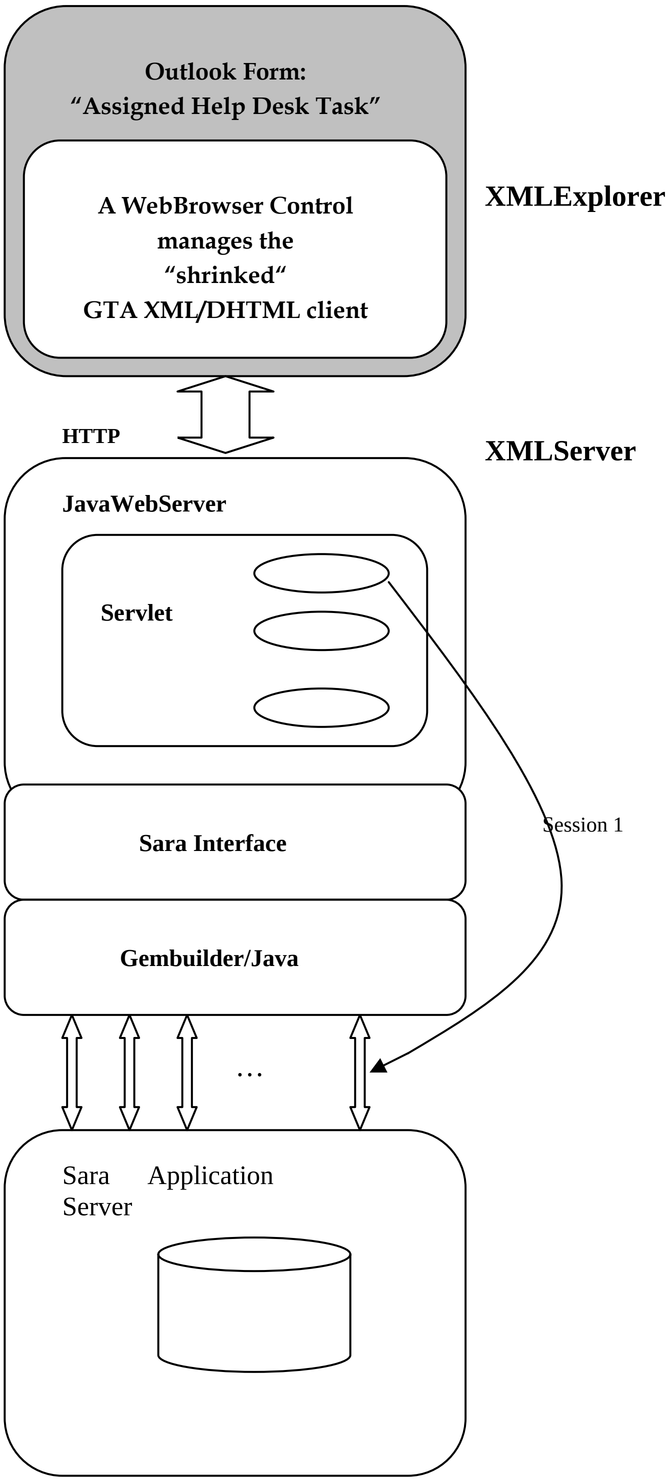


Figure 10 Computas GTA integration, technical overview, [Appendix D15]

The properties and methods of the WebBrowser Control are shown in below. To navigate to the XMLServer, I use the Navigate property. Then to be able to send several http requests, I needed a property that was set when the WebBrowser Control had successfully received requested data (e.q. the login procedure) from the XMLServer. An appropriate property called "ReadyState" was used. This property was not found in the "Iexplore.hlp" help-file, but discovered after some research. For more information about the WebBrowser Control see: link

Properties	Application, Busy, Container, Document, **ReadyState**, Height, Left, LocationName, LocationURL, Offline, Parent, Silent, RegisterAsBrowser, RegisterAsDropTarget, Top, TopLevelContainer, Type, Width
Methods	ExecWB, **Navigate**, Navigate2, Quit, Refresh, Refresh2, Stop, GoBack, GoForward, GoHome, GoSearch,

The following describes the syntax of the Navigate method. To navigate to an URL one uses this syntax:

Object.Navigate URL [Flags,] [TargetFrameName,] [PostData,] [Headers]

- **Object** = WebBrowser or InternetExplorer

  The WebBrowser object is an ActiveX Control that provides browsing capabilities to applications.

  The InternetExplorer object controls an instance of the Internet Explorer application through OLE automation.

- **URL** (Universal Resource Locator)

  A string that expresses the resource or file

- **Flags** =A constant or value that specifies the following: (Optional)

NavOpenInNewWindow **1**	Open the resource or file in a new window
NavNoHistory **2**	Resource or file not added to the history list. The new page replaces the current page in the list.
NavNoReadFromCache **4**	No read from the disk cache for this navigation.
NavNoWriteToCache **8**	No write of this navigation to the disk cache.

7.3.1 Outlook limitations

Even though using the WebBrowser control could seem relatively straight forward, limitations appeared after some research. I tried to find workaround methods, but I came up with the following limitations in Outlook:

- One cannot bind WebBrowser Controls properties to an Outlook field run-time
- One cannot bind WebBrowser Controls properties to an Outlook field design-time.
- Outlook does not support DoEvents, and therefore one cannot use Do..Loops to evaluate on WebBrowser.ReadyState

7.3.2 Creating ActiveX Components

To work around this situation I had to make two Visual Basic ActiveX components. One ActiveX Control called "CXGTAforOutlook.ocx" and one ActiveX DLL called "CXGTACreateOutlookReminder.dll".

The ActiveX Control manipulates the WebBrowser control to navigate the sequence of http commands needed to load GTA and the corresponding process definition (i.e. process GUI). To be able to navigate a sequence of http commands, I used the following event that occurs when the document that is being navigated to, reaches the READYSTATE_COMPLETE state:

object_DocumentComplete(ByVal *pDisp* As Object, *URL* As Variant)

- **Object** = WebBrowser
- **PDisp**

  I used this object to trap the final document complete event, because the XML client contains multiple frames. If a document contains multiple frames, this event is fired once for each frame. When the multiple-frame document has finished loading, the top-level frame fires the event one final time.

I then had to add properties. The properties of the "CXGTAforOutlook.ocx" control are shown below. The code is found in appendix B.

Login_Command	Used to navigate the GTA login procedure.
Worklist_Command	Used to navigate the GTA worklist. Not visible, but must be sent prior to a process request.
Process_Command	Used to navigate the GTA process.
Started	Used to start navigating when http commands are set by the application that uses the control.

To load the "CXGTAforOutlook.ocx" the following syntax is used:

Set Mypage = item.GetInspector.ModifiedFormPages("Ticket")

Set MyControls = Mypage.Controls

MyControls("CXGTA1").Login_Command = LoginString

MyControls("CXGTA1").Worklist_Command = WorklistString

MyControls("CXGTA1").Process_Command = GtaOutlookString

For more information about the "CXGTAforOutlook.ocx" see Appendix B

The ActiveX DLL "CXGTACreateOutlookReminder.dll" is used to create a GTA Taskreminder and populate the Outlook item with data. To do this the DLL is automating both Microsoft Outlook and Internet Explorer.

To create a taskreminder, the following http command was used:

"http://winston/servlet/GTAServlet?type=createOutlookReminder&toMyself=true&processName=Allocate PC"

GTAServlet code for type=createOutlookReminder was made to send an Html string back to the client with the belonging GTA ProcessID. (The original createReminder returns XML). The HTML string looks like this:

<html> <title>ProcessID</title> </html>

To get the unique GTAprocessID, which is the title element of the HTML document the following syntax was used:

Set MyHTMLobject = IE2.document

ProcessID = MyHTMLobject.Title

This Process ID is used in the "CXGTAforOutlook.ocx" control to load the process GUI:

**http://winston/gta/gtaApp/gtaOutlook.html?type=process&
process=GtaProcessID**

The "CXGTACreateOutlookReminder.dll" is called using the CreateObject method in Outlook:

Dim AssignTask

**Set AssignTask = _
item.Application.CreateObject("GTACreateOutlookReminder.GTA_OR")**

AssignTask.<properties> = item.userproperties.find("<field value>")

For more information about the "GTACreateOutlookReminder.dll" and its properties see Appendix B. For more information about creating ActiveX components see: link.

7.4 The Help Desk tool

The Help Desk tool consists of four Outlook forms, two Public Folders and an Access database. All forms are published in the forms library of the Help Desk Folder. Thus when the folder is active, the forms are available from the "actions menu". Forms can also be published in the organizational Forms Library, thus available using the "choose form" command. Using the Help Desk tool, users submit requests using a Help Inquiry form. The Help Desk manager then assigns the tasks to the appropriate person from the "read" page of the Help Inquiry form", and the corresponding GtaTaskReminder is sent to the XMLServer (see figure 10). The assigned person then opens the Assigned Task Item with corresponding information and the GTA process window. For organizing and tracking of previous solved tasks, assigned and resolved tasks are kept in the "Assigned Help Tasks Folder". Fields in the Assigned task item are also recorded in an Access database. The figure below illustrates the solution:

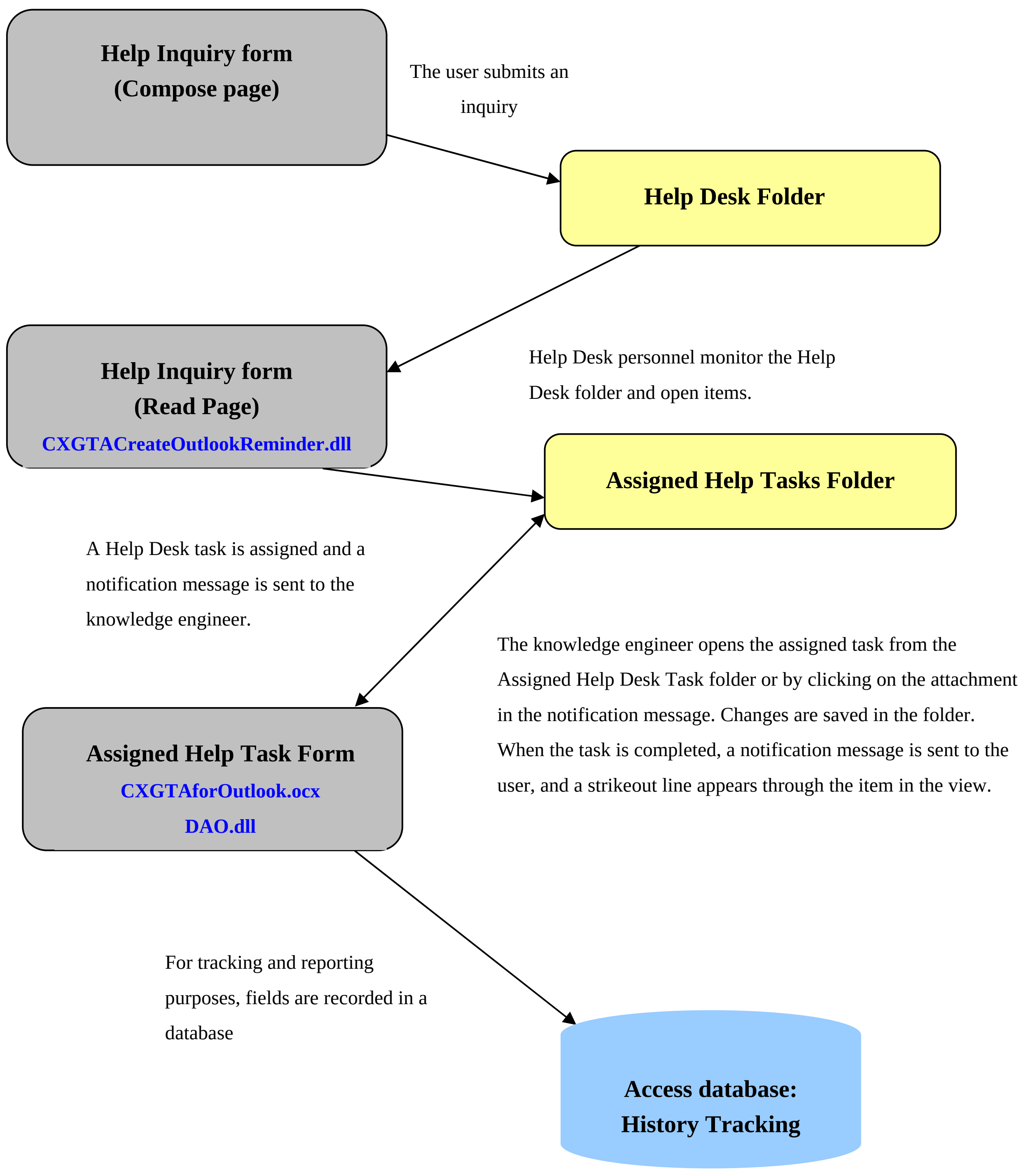

Help Inquiry form
(Compose page)
The user submits an inquiry
Help Desk Folder
Help Inquiry form
(Read Page)
CXGTACreateOutlookReminder.dll
Help Desk personnel monitor the Help Desk folder and open items.
Assigned Help Tasks Folder
A Help Desk task is assigned and a notification message is sent to the knowledge engineer.
The knowledge engineer opens the assigned task from the Assigned Help Desk Task folder or by clicking on the attachment in the notification message. Changes are saved in the folder. When the task is completed, a notification message is sent to the user, and a strikeout line appears through the item in the view.
Assigned Help Task Form
CXGTAforOutlook.ocx
DAO.dll
For tracking and reporting purposes, fields are recorded in a database
Access database:
History Tracking

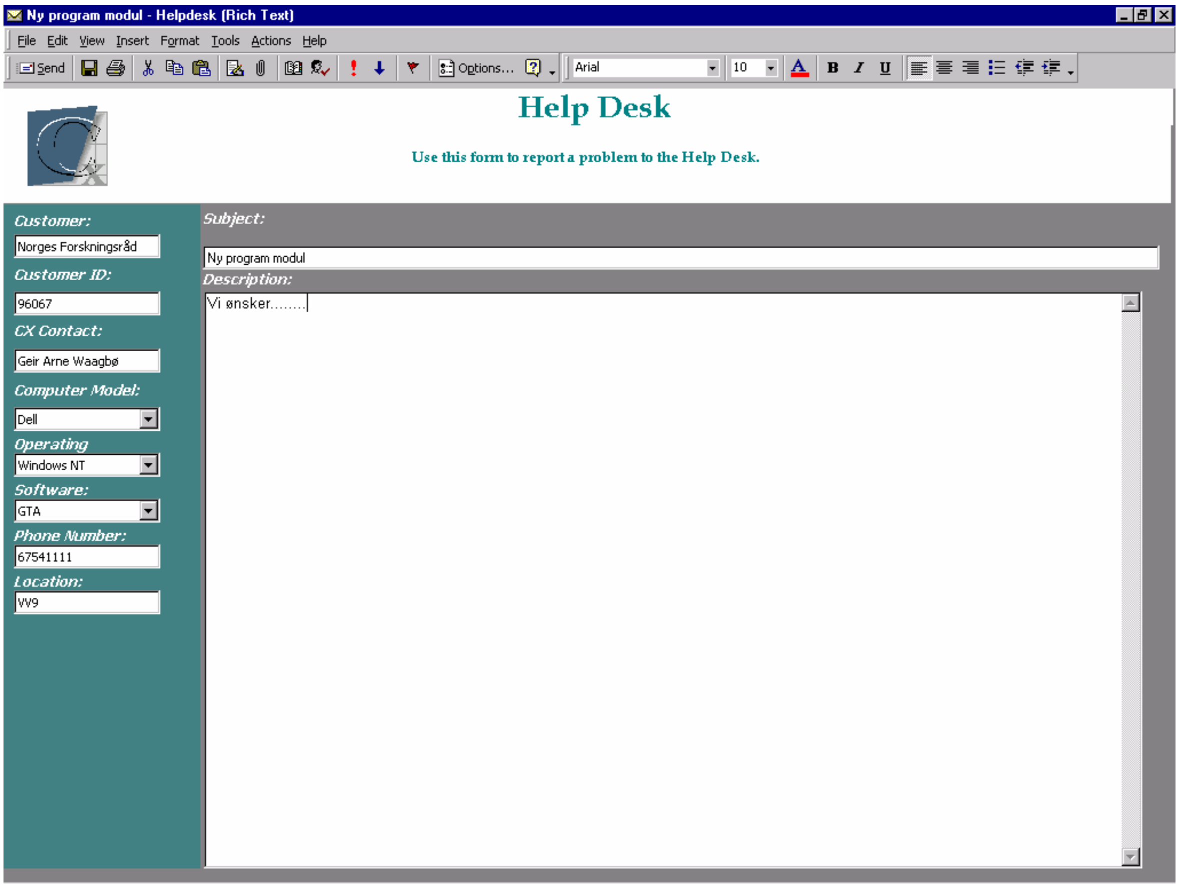
Figure 11 Help Desk Prototype, technical overview.

Detail functionality:

"Help Inquiry Form"

See figure 12 and 13. To open the Help Inquiry form, one has to click on the Help Desk Folder (create a shortcut). Then click New Help Inquiry on the Actions menu. The form is preaddressed to the Help Desk Folder, so the user does not have to specify the address. When the users have filled in all information, they just click "Send". When the inquiry is sent, the Help Desk administrator opens the form in the Help Desk Folder (its read page) and fills in additional information such as technician name and GTA process name. Subsequently he then clicks "Assign and Close". This action triggers the following:

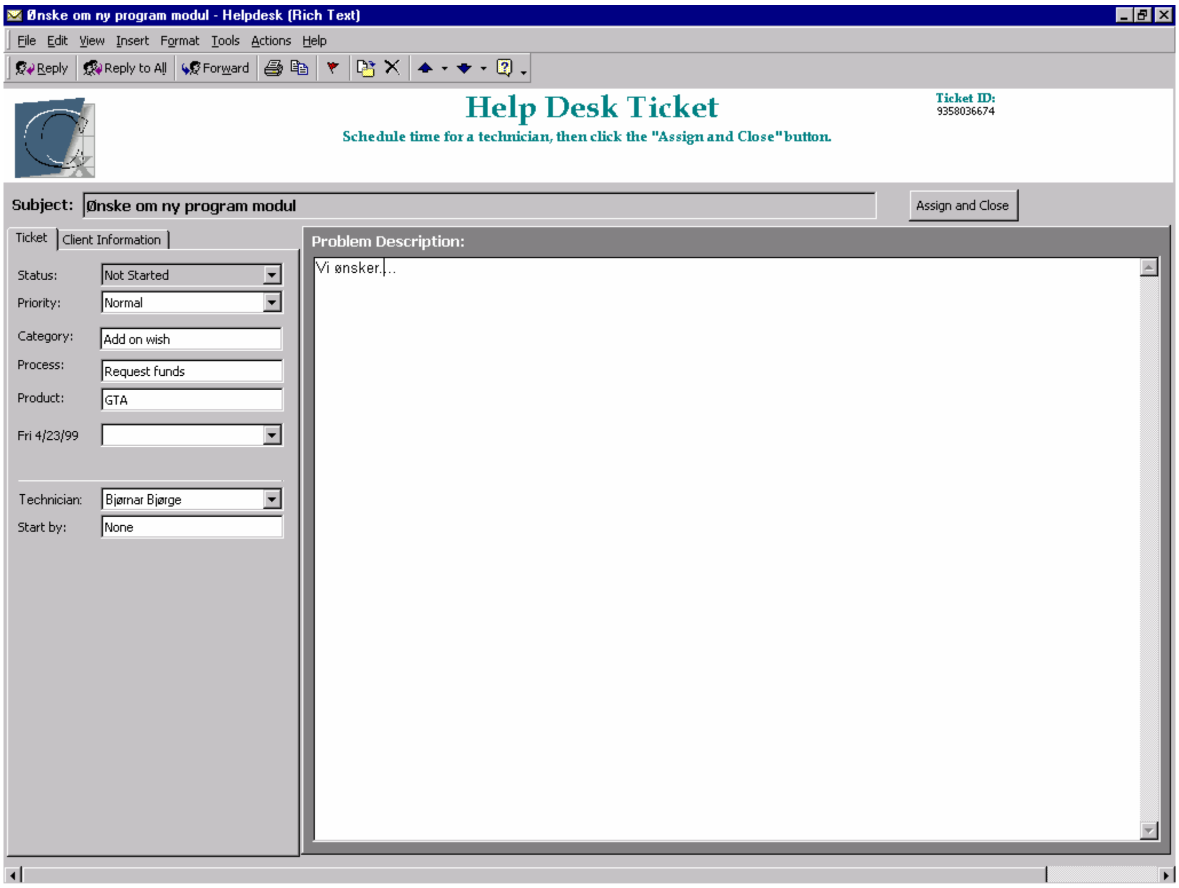
- Generation of an "Assigned Help Task " item in the "Assigned Help Tasks Folder". Information in the item is written from the "Help Inquiry Form", and the original item is deleted from the Help Desk Folder.
- HTTP commands are sent to the XMLServer to create a new GTAtaskreminder for the technician
- Notification mail with a shortcut to the task to complete is sent to the technician.


Ny program modul - Helpdesk (Rich Text)
File Edit View Insert Format Tools Actions Help
Send
Options...
Arial
10

Help Desk

Use this form to report a problem to the Help Desk.

Customer: Norges Forskningsråd
Customer ID: 96067
CX Contact: Geir Arne Waagbø
Computer Model: Dell
Operating: Windows NT
Software: GTA
Phone Number: 67541111
Location: VV9
Subject: Ny program modul
Description: Vi ønsker........


Ønske om ny program modul - Helpdesk (Rich Text)
File Edit View Insert Format Tools Actions Help
Reply
Reply to All
Forward
Assign and Close
Ticket
Client Information

Help Desk Ticket

Schedule time for a technician, then click the "Assign and Close" button.

Ticket ID: 9358036674

Subject: Ønske om ny program modul

Status:	Not Started
Priority:	Normal
Category:	Add on wish
Process:	Request funds
Product:	GTA
Fri 4/23/99	
Technician:	Bjørnar Bjørge
Start by:	None

Problem Description:

Vi ønsker....

Figure 12 The Compose page of the Help Inquiry form

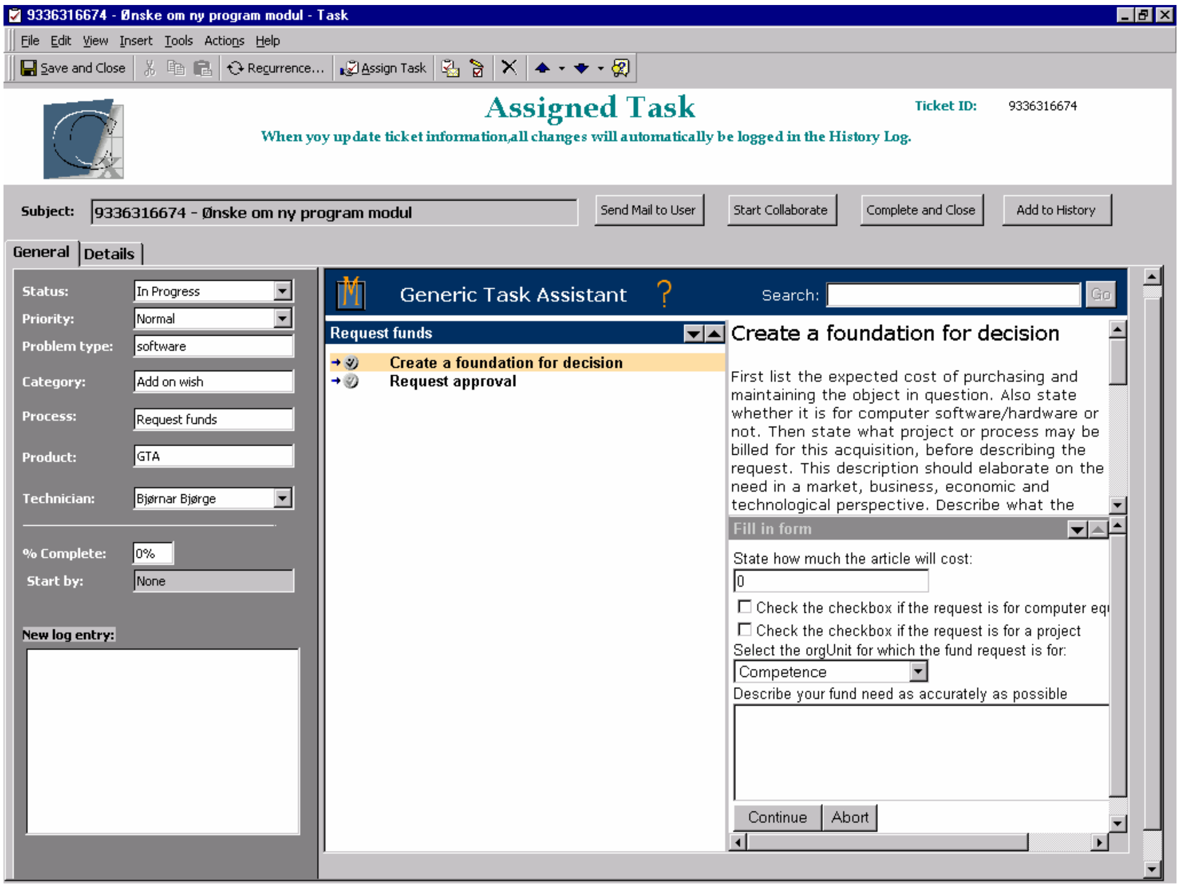
Figure 13 The Read page of the Help Inquiry form

"Assigned Help Task Form"

See figure 14. To perform an assigned task, the technician opens the item using the shortcut in the message field by opening it from the folder. This triggers the following:

- Field values are set according to values set in the Help Inquiry form.
- HTTP commands are sent to the XMLServer to view the GTA process description of the associated task assigned by the Help Desk administrator

The technician can read information automatically logged in the History field and subsequently add information to the field. During the completion of the task the technician can either send mail to the user, using the "Correspondence Mail Form" or post a discussion item to the "Assigned Help Tasks Folder" using the "Task Discussion Form" to gather information about specific problem areas.



Assigned Task

Ticket ID: 9336316674

When yoy update ticket information,all changes will automatically be logged in the History Log.

Subject: 9336316674 - Ønske om ny program modul

Status:	In Progress
Priority:	Normal
Problem type:	software
Category:	Add on wish
Process:	Request funds
Product:	GTA
Technician:	Bjørnar Bjørge
% Complete:	0%
Start by:	None

New log entry:

Request funds
- Create a foundation for decision
- Request approval

Create a foundation for decision

First list the expected cost of purchasing and maintaining the object in question. Also state whether it is for computer software/hardware or not. Then state what project or process may be billed for this acquisition, before describing the request. This description should elaborate on the need in a market, business, economic and technological perspective. Describe what the

Fill in form

State how much the article will cost:
0

☐ Check the checkbox if the request is for computer eq
☐ Check the checkbox if the request is for a project

Select the orgUnit for which the fund request is for:
Competence

Describe your fund need as accurately as possible

Figure 14 The Assigned Help Task form.

“Help Desk Main Public Folder”

Contains “Help Inquiry Forms” waiting to be assigned.

“Assigned Help Tasks Folder”

Contains completed tasks and discussions items for organizing, collaboration and tracking purposes.

“History Tracking Database”

Data from tasks are “mapped” to this database for tracking and reporting purposes.

For more information about the Help Desk prototype, see appendix A.

8 Comparisons

8.1 Computas GTA vs. Microsoft Exchange

Both Computas GTA and Microsoft Exchange based systems have workflow functionality, although based on different concepts. The Workflow Management Coalition (WfMC) detailed definition of a Workflow system:

"A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications. A Workflow Management System consists of software components to store and interpret process definitions, create and manage workflow instances as they are executed, and control their interaction with workflow participants and applications. Such systems also typically provide administrative and supervisory functions, for example to allow work reassignment or escalation, plus audit and management information on the system overall or relating to individual process instances."

The following figure illustrates the frameworks and components that GTA is based on (i.e. the SARA framework) [Appendix D14]:

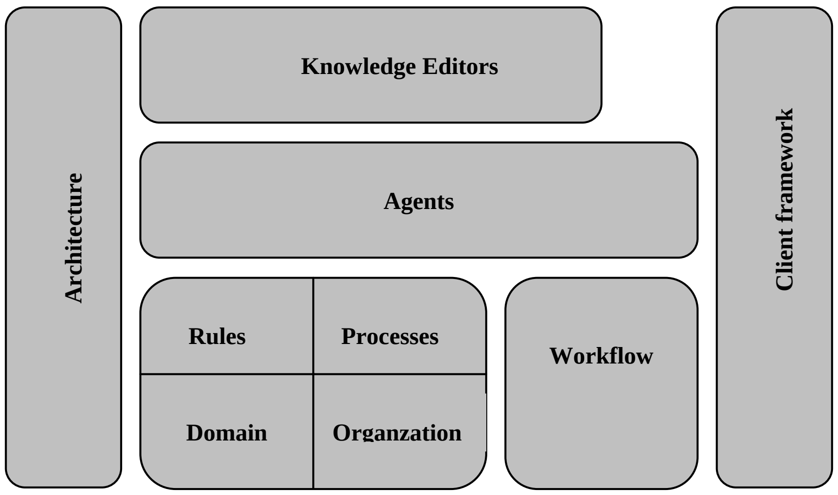


Figure 15 The Sara framework [Appendix: D14]

Workflow framework

The workflow framework consists of components and frameworks for routing functionality between participants in an organization. The framework consists of mechanisms for creating, assigning and routing tasks in addition to overdue handling, monitoring and control.

Domain framework

Using the Domain framework, one has the opportunity to create a domain model. A domain model represents the core of the business and is used together with the Workflow framework.

Organization framework

The organization framework offers authorized persons opportunity to model the structure of any organization. This structure usually concerns Persons, Groups and Roles.

Process framework

This framework contains a definition language for defining work processes as a network of activities, and sub-processes and mechanisms for executing such processes.

Agent framework

The Agent framework is used to develop system components, which handle knowledge. An Agent is responsible for acquiring, acting on and distributing knowledge about other objects. For instance, the process agents handle the process definitions.

Knowledge Editors

The knowledge editor is used to edit knowledge represented as rules, processes and organization structure.

Architecture

The architecture is based on a three-tier client server architecture with a client, application server, and enterprise and data storage tier.

Client framework

The client framework provides components for creating clients in Smalltalk, Java, HTML and XML. The XML client is used when integrating GTA with MS Outlook.

When confronting WfMC's definition of a workflow system, GTA covers workflow functionality as interpreting the process definition, managing and monitoring workflow instances and their interaction with participants and applications. In addition, GTA integrates business objects and knowledge bases (instructions of how to perform tasks). That is, GTA guides the user through every step of a process ensuring correct execution.

For workflow systems based on Microsoft Exchange the term "Infrastructure-Based Workflow has been used. The following figure shows the Exchange service architecture (i.e. workflow platform):

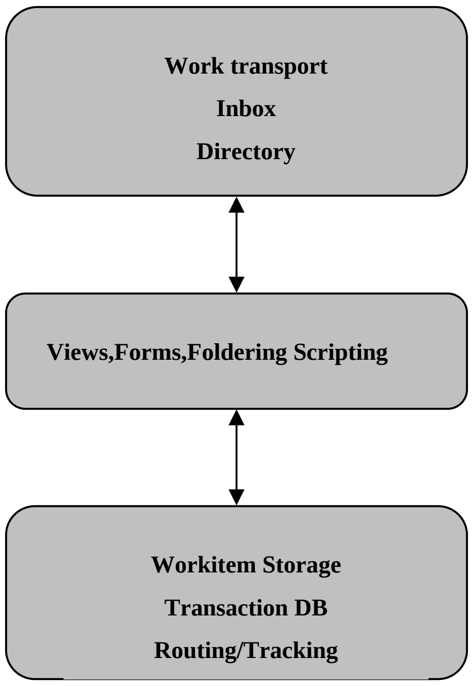


Figure 16 Exchange-based application architecture [7]

Directory services

These services hold all information about an organization's resources and users (i.e. addresses, mailboxes, public folders and configuration information). These services take care of security information and replicate it to other services.

Information store/transfer services

These services provide storage and transfer of "workitems". That is mail messages and other information (documents). This information could either be stored in public or private folders (unlimited storage in Exchange 5.5).

To be able to compare Computas GTA and Microsoft Exchange, I had to find software products layered on Microsoft Exchange Server. On the Giga Information Group's "Business Process and Workflow Conference" March 14-16 1999 in Florida, the following examples of so-called "Infrastructure-based" workflow was introduced [7]. The Giga Information Group provides knowledge and strategic technology and management advice to IT decisions makers. For more information see [Appendix D17].

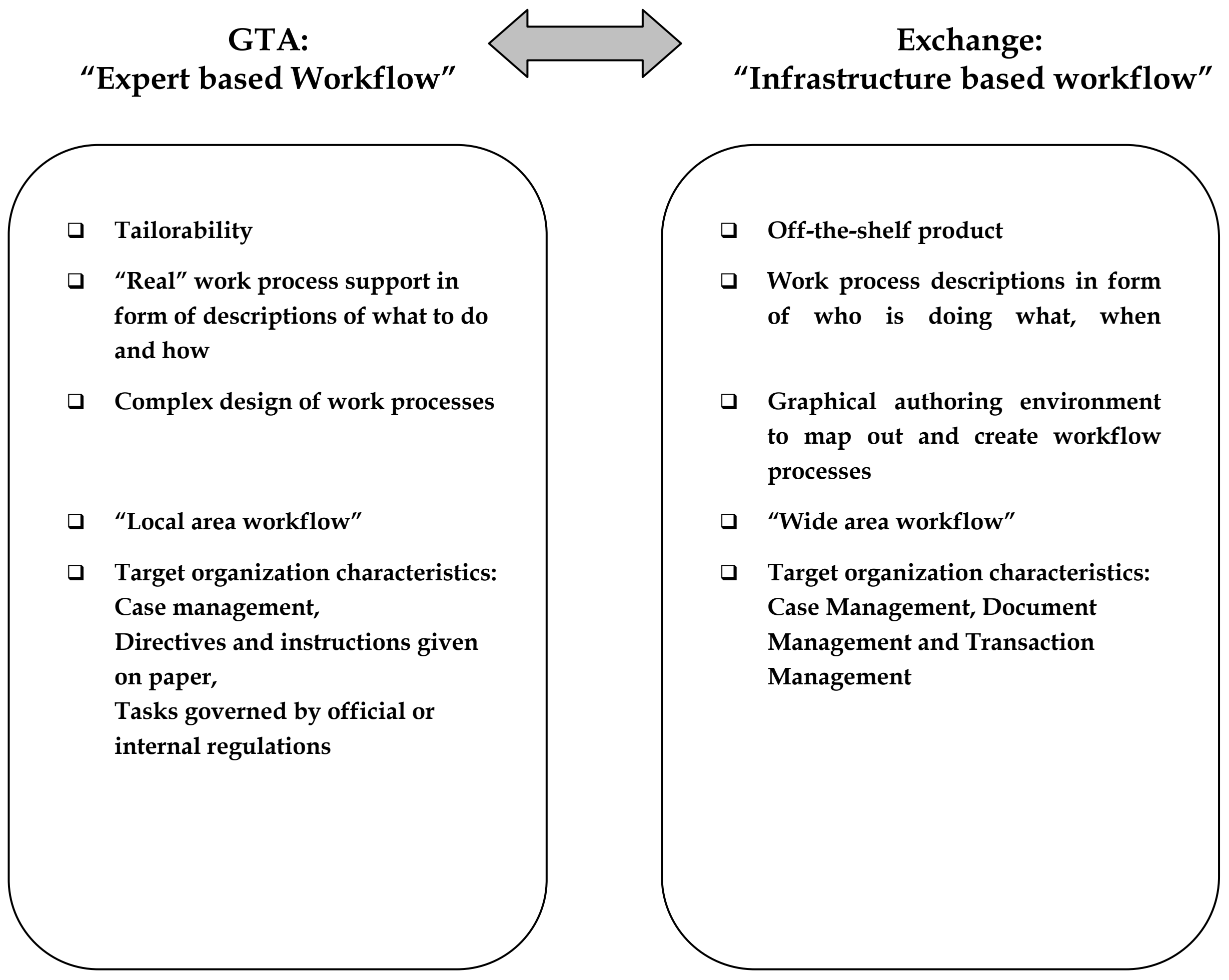
- Keyfile Keyflow (most mature product)
- Eastman Software WFX (work management, not flow)
- ONEstone Prozessware (based on Lotus Domino, but an Exchange version is also available)

For a relatively complete list of "Workflow and GroupWare" products, take a look at link. I wanted to focus on Keyfile Keyflow, and I had a Keyflow Evaluation CD that made it possible to view demonstrations and evaluation information.

Keyflow functionality:

- Emphasis on routing and tracking
- Share process definitions in public folders
- Server based work-tracking
- Simple containment (docs and application data)
- Standard and custom forms
- Graphical authoring environment to map out and create workflow processes (new maps can be defined via Outlook Compose form)
- Reporting capabilities to analyze process performance
- Attachment for review or process of specific information
- Thin client via Outlook Web Access

Confronting WfMC's definition against Keyflow's functionality, it provides standard workflow functionality as initiating, routing and tracking of workflow processes. However focus is on routing, not how workflow participants are going to perform their work (i.e. structured work process descriptions). The main differences between Computas GTA- and MS Exchange-based systems are pictured in figure 17.



GTA:
"Expert based Workflow"

- **Tailorability**
- **"Real" work process support in form of descriptions of what to do and how**
- **Complex design of work processes**
- **"Local area workflow"**
- **Target organization characteristics:**
  Case management,
  Directives and instructions given on paper,
  Tasks governed by official or internal regulations

Exchange:
"Infrastructure based workflow"

- **Off-the-shelf product**
- **Work process descriptions in form of who is doing what, when**
- **Graphical authoring environment to map out and create workflow processes**
- **"Wide area workflow"**
- **Target organization characteristics:**
  Case Management, Document Management and Transaction Management

Figure 17 Computas GTA vs MS Exchange

8.2 Ms Exchange and Internet Information Server

Computas GTA and Microsoft Exchange based applications have a relatively "in-house" focus. I.e. they focus on how help desk tasks are routed, performed and tracked within an organization. The customer is not an "active part". To create a more "Customer integrated" Help Desk, one could think of using the features of Microsoft Exchange Server and Internet Information server. The figure below shows a way of developing a "Help Desk solution" based on Microsoft Exchange Server, Internet Information server and SQL database:

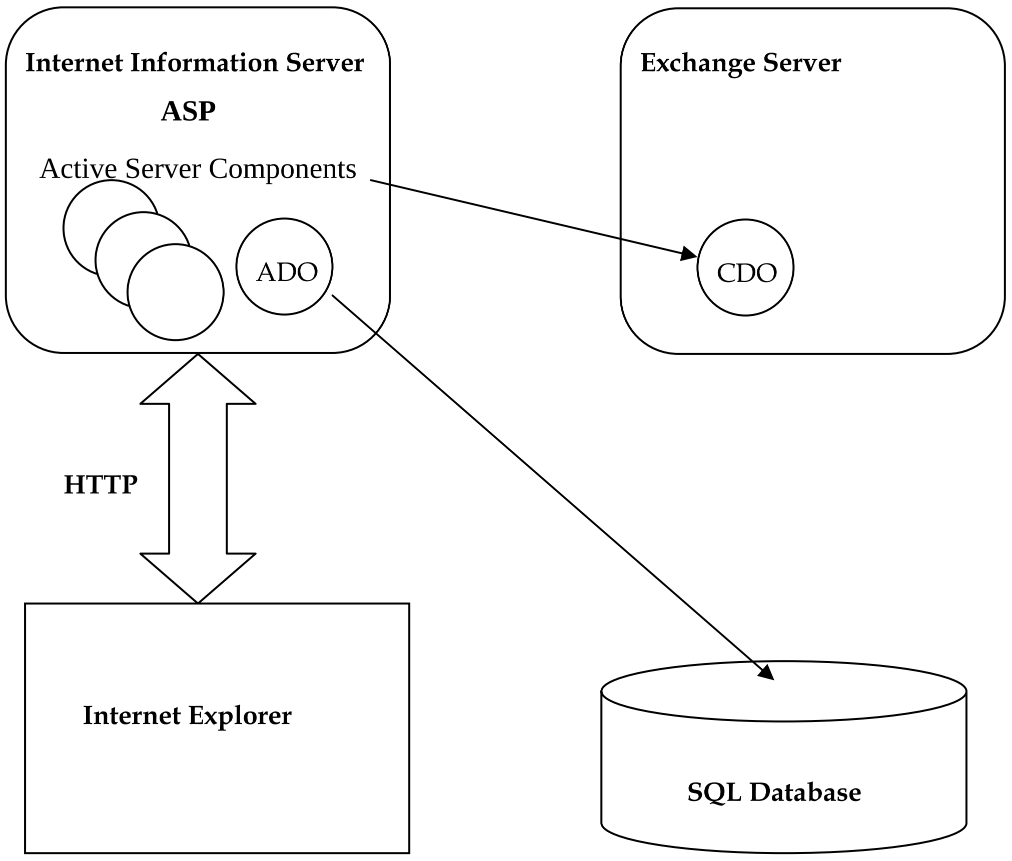


Figure 18 Exchange and IIS-based application architecture

Using this Web Based Help Desk, clients can submit inquiries, check on status and search for help in the database. E-mail notification of high priority inquires can be set up using Exchange features. The architecture is based on several features. I do not intend do go into detail, but some definitions are needed to illustrate the concept. More information can be found at link.

Active Server Pages

Active Server Pages is a feature of Microsoft Internet Information Server. In addition to standard HTML, an ASP page contains server side scripts. This script is used to generate an HTML page with dynamic content. "Server side" means that the HTML page is generated at the server and then sent to the client (i.e. a browser). For database access through IIS, ADO is playing an important role.

Active Server Components (ASC)

Active Server Components work in conjunction ASP. ASC enable the web server to interface with other server software components.

ActiveX Data Objects

ActiveX Data Objects are a part of the technologies that make up Microsoft's UDA (Universal Data Access) concept. ADO are used by an ASP page to retrieve specific information from a database, and then generate an HTML representation of the content and send it to the client (i.e. browser). In the figure above ADO are used in conjunction with OLE DB. Using the OLE DB provider for the specific data source, one can retrieve information from different types of databases (Access, Oracle and SQL server) using the ADO object model.

Collaboration Data Objects

Collaboration Data Objects is an ASC that provides features for calendaring, scheduling and workflow. CDO is delivered as a part of Exchange Server 5.5. For more information about CDO, see link.

This approach goes beyond "In-house" Help Desks by developing a relationship with customers via the Help Desk interface. [11] states: *"The results of the project illustrate the potential of the help desk to act as a link between IT and the customer, so that the help desk may act as a point of information interchange, feeding information from the customer into development teams and feeding information back out from the development teams to the customer."*

In addition, using this application architecture, enterprises can provide a 24-hour customer service.

8.3 Client vs. Server scripting

The Help Desk application is based on client scripts in a customized Outlook task form. Even though Exchange Public folder is used, this application does not require users to be connected to Exchange server. On the other hand, working with forms is tied to the Outlook client.

However, when developing workflow applications one has options regarding whether the routing processes should be executed on the server or client-side. The following illustrates some advantages/disadvantages regarding server vs. client scripting [4].

Outlook client

In the Outlook client application, VB script is used to develop workflow applications (customizing standard forms like e-mail, tasks or contact items).

Advantages:

- Outlook Object model used when automating Outlook
- The CreateObject method can be used to run other ActiveX automation objects.
- Forms available to any Exchange recipient or Outlook client (do **not** need to be connected to the Exchange server)
- Client based execution
- Applies to any folder in the recipients mailbox
- User defined and configurable
- User-friendly, minimal training required (i.e. appropriate for user who is familiar working with the Outlook client).
- Rules Wizard installed as an Outlook Add-In

Disadvantages:

- Very limited use of ActiveX controls (properties **ignored**). However one can make custom controls to achieve specific functionality.
- Not Scalable
- Rules Wizard only applies to inbox

Exchange event service

The Exchange event service is a new feature of Exchange 5.5 for developing workflow applications. Event Service scripts use Collaboration Data Objects (CDO) to create customized applications. For more information about CDO, see link.

Advantages:

- CDO provides new features for calendaring, scheduling and workflow
- The CreateObject method can be used to run other ActiveX automation objects. (i.e. database access using ADO or custom components created using Visual Basic or Visual C++)
- Server based logic, execution and administration
- Scalability dependent on server memory and configuration

- High level of security. Only users with permission can develop, install and debug scripts

Disadvantages:

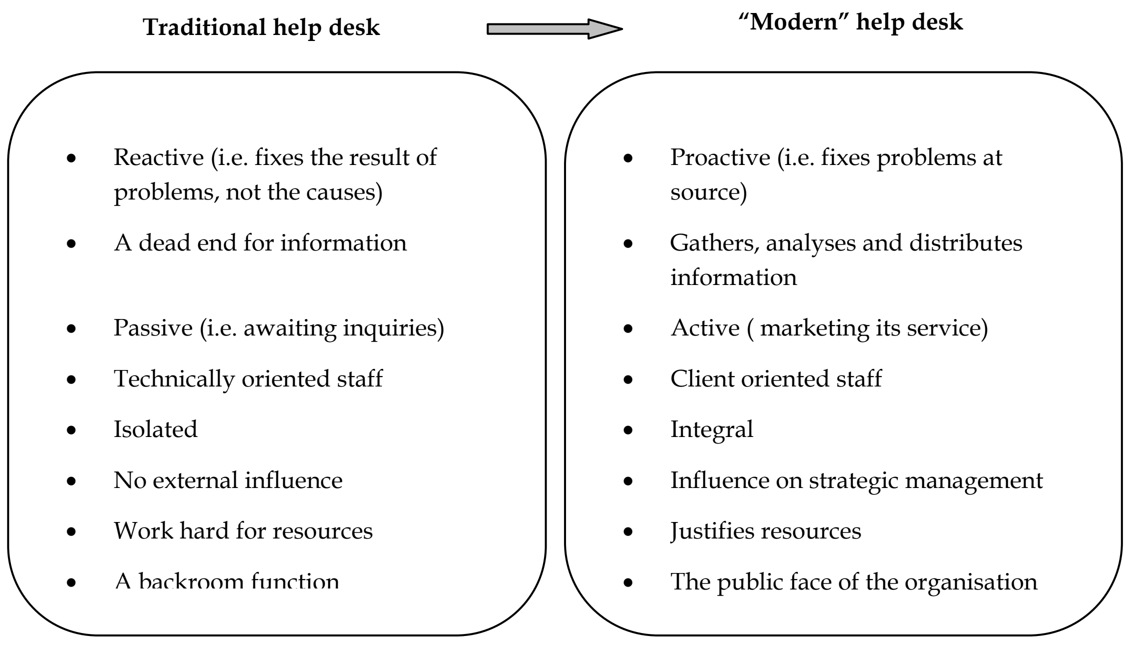
- Requires well trained developers using VBscript or JavaScript
- Interactive coding environment not available
- Applications take longer to develop and deploy
- Difficult to administer because scripts must be installed in each folder where they operate. (i.e. no centralized script administration tool)
- Inefficient on large number of mailboxes with much mail traffic

For more information about using the Exchange Event service see [4].

9 Conclusion

Improving Help Desk development and management is an ongoing process. To enhance Help Desk success, a path is identified (see figure 5). This path focuses not only on development, but also on management requirements after the Help Desk is implement.

Success with the Help Desk does not only require better organizing internally, but also recognition of how the Help Desk can be integrated with other macro processes in the organization. It has been stated [15] that while preventing problems from occurring, the Help Desk may also act as the front-line for IT with a direction towards more expert services through advising and decision making. Thus, one challenge of the Help Desk department is to gather and analyze information and use it in future strategic management within the organization. Figure 19 illustrates the direction that the most evolving and “modern” Help Desks are taking [12]. This approach to the Help Desk represents a movement towards a “macro” definition of the Help Desk role in the organization.



Traditional help desk	“Modern” help desk
Reactive (i.e. fixes the result of problems, not the causes)	Proactive (i.e. fixes problems at source)
A dead end for information	Gathers, analyses and distributes information
Passive (i.e. awaiting inquiries)	Active (marketing its service)
Technically oriented staff	Client oriented staff
Isolated	Integral
No external influence	Influence on strategic management
Work hard for resources	Justifies resources
A backroom function	The public face of the organisation

Figure 19 Traditional versus Modern Help Desks [12]

In other software development systems, business process support is used to improve how organizations are working. A paper on the subject "Object-Oriented workflow management systems" [10], illustrates an approach for business processes support. *"…Our experiences with applying a combination of object technology and expert systems technology in several real world systems show that it can add a knew level of support for business processes to the field of workflow management".* In the approach of this thesis it is clear that a real world system also includes a Help Desk system. The benefits of modeling processes are improved efficiency in the support department, and a more structured way of handling end-users by focusing on and standardizing the workflow through the organization. This is a necessity, especially in SME organizations. They receive inquires sporadically and often prefer using non-dedicated support personnel. Thus, work processes enables non-dedicated support personnel to perform Help Desk related tasks.

However, using work processes to support everyday tasks requires involvement in the process definitions phase by the Help Desk personnel to ensure that the system provides "real" support. And, not every task can be or is suitable to be modeled as work processes, especially in large enterprises that receive a greater quantity of help inquires and therefore have lots of repetitive tasks to do every day.

There is a main difference between a system based on Computas GTA and other workflow systems like Keyfiles' Exchange based system. The Computas GTA system approach is to focus first on what is to be done and how (i.e. the process and activities). Other workflow tools like Keyfiles' Keyflow, emphasis on routing of work between people in the organization. Which system to use depends on the client organization. In large enterprises with dedicated support personnel and a greater quantity of help desk related tasks, emphasis is on query logging, tracking, routing and escalation. In small and medium-sized enterprises with non-dedicated support personnel (in this case Computas AS) focus is more on what is to be done and how (i.e. structured work process descriptions), in addition to standard routing between participants. In this case a Computas GTA based system is a better choice.

Most help desk applications are focused on IT, not on how the support personnel are carrying out their tasks. That is, users must adapt to the application terminology. The Help Desk tool based on Computas GTA and Microsoft Outlook focus on how to support the employees in performing tasks.

The benefits of the Help Desk tool can be summarized as follows:

- Enables use of non-dedicated support personnel
- Rapid knowledge transfer and training
- Full control of who has done what when
- A more structured way of performing end-user support
- More time to focus on non-automated tasks

Integrating Microsoft Outlook and Computas GTA has involved some technical challenges and identified Outlook Application limitations. To work around these limitations two ActiveX components have been made. These ActiveX components have been used to integrate the Task functionality in Outlook with the XML client of Computas GTA. For communication between Outlook and GTA, the HTTP protocol has been used.

Modifications to the GTA client have also been done. The worklist frame in the original GTA XML client is removed, since Outlook handles the folders (worklist). To be able to load the GTA work process GUI in the Outlook custom Task item, additional GTA Server code was made. The integrated GTA client handles rendering and execution of work processes.

Based on the integration and the adaptions above the required Help Desk application prototype functionality is obtained. Finally, the Help Desk application has been tested, and the HTTP communication between Microsoft Outlook and Computas GTA works.

9.1 Further work

- Development and testing

The Help Desk application is a prototype. The user interface and database connectivity must be adjusted and tested.

- Expand GTA functionality

Since the GTA web client is based on the SARA web client framework, it is possible to expand GTA functionality through the SARA interface, for instance more work process automation and feedback. An internal Computas project named "GTA" is refining the GTA concept.

- Routing Help Desk tasks by subject

Using Exchange Server scripts a feature that routes the message item to the right person, is about to be shipped for use in the Help Desk department. This feature can be used in conjunction with the Outlook-GTA prototype.

- Adjust Help Desk work processes

For the Help Desk work processes to be effective, adjustments and refinements by the Help Desk personnel are needed.

10 Terminology

API	Application Program Interface. Provides the ability to programmatically make request of the operating system or another application
CBR	Case-based Reasoning
CDO	Collaborative Data Objects
CSS	Cascade Style Sheet
CX	Computas AS
DAO	Data Access Objects. An API to get programmatically access to a Microsoft Access database
DHTML	Dynamic HTML
DLL	Dynamic-link libraries
DTD	Document Type Definition
Exchange Public Folder	An Exchange feature that allows information in one folder to be shared
GTA	Generic Task Assistant.
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol. The application protocol used for communication between GTA and Outlook.
IE	Internet Explorer Application.
IMAP	Internet Message Access Protocol. A standard protocol for accessing e-mail from a local server
MAPI	Messaging Application Program Interface. A standard developed to provide complete system independence for messaging applications
OCX	OLE custom control
OLE	Object Linking and Embedding

POP3	Post Office Protocol 3. A standard protocol for receiving e-mail
QFD	Quality Function Deployment
SARA	A Software Reuse Architecture for Building Expert Workflow Systems
SLA	Service-level agreement
SMTP	Simple Mail Transfer Protocol. A standard protocol for transferring e-mail across the Internet
SQL	Structured Query Language
TCO	Total Cost of Ownership
VBScript	Microsoft Visual Basic Scripting, a subset of the Microsoft Visual Basic programming language. Used in Outlook to customize forms.
WebBrowser Control	An AxtiveX Control used to provide browser capabilities in an Outlook task item.
WfMC	Workflow management Coalition
XML client	A two tier implementation of the SARA Web Client framework. The user interface is based on the Extensible Markup Language.
XMLExplorer	A subset of the XML client. Runs in a web browser environment
XMLServer	A subset of the XML client. Runs in the context of a Java Web Server

11 References

[1]	**A. Barr** Software trends at the help desk *Cutter information corp. IIS Vol. 8, No. 8*
[2]	**A. Barr** Automating knowledge flow at the help desk *Cutter information corp. IIS Vol. 6, No. 11*
[3]	**N. Bruton** Effective User Support: How to Manage the IT Help Desk *McGraw-Hill, New York, NY 1995, ISBN :07506 3811 7*
[4]	**R. Byrne** Using the Exchange Event Service to develop Workflow applications *Micro Eye, 1998.*
[5]	**A. Cushman, T. Kirk, B. Keyworth** The Consolidated Service Desk: Enabling proactive Support Processes *GartnerGroup: Strategic Analysis Report MDC 1997*
[6]	**B. Czegel** Running an Effective Help Desk *John Wiley & Sons, Inc, NY 1998, ISBN: 0 471 24816 9*
[7]	**Giga Information Group** Business Process and Workflow Conference *Information about the conference and solution providers. March 14-16, 1999*
[8]	**P. Harmon** Experts systems for help desks *Cutter information corp. IIS Vol. 7, No. 9*
[9]	**P. Krebs** Building Applications with Microsoft Outlook 98 *Microsoft Press, 1998. ISBN: 1-57231-718-3*
[10]	**B.H. Kallåk, T.B. Pettersen, J.E. Ressem** Object-Oriented Workflow management: Intelligence, flexibility, and real support for business processes *Position paper on the OOPSLA '98 Workshop on : Implementation and Application of Object Oriented Workflow Management Systems*
[11]	**R.C. Marcella and I. A. Middleton**

	The Role of the Help Desk in the Strategic Management of Information Systems *OCLC Systems and Services*, Vol.12 No. 4, 1996, p4-19.
[12]	**R.C. Marcella and I. A. Middleton** Key Factors in Help Desk Success *British Library R&D Report number 6247, 1996*
[13]	**Microsoft** Microsoft Outlook 2000 Features and configuration guide *Microsoft corporation 1998*
[14]	**Microsoft** Microsoft Outlook 2000 Product Enhancements Guide *Microsoft corporation 1998*
[15]	**D. Pancucci** Internal Help Desk fuels Texaco's Business *Customer Service Management, Sept 1995, p 38-39*
[16]	**A.H. Thomas and R.M. Steele** The Virtual Help Desk *Strategic Management Center, 1996, ISBN: 1 850 32204 X*
[17]	**Workflow Management Coalition** Terminology & Glossary Number TC-1011, (URL: http://www.aiim.or/wmfmc/)

Appendix A

Help Desk Prototype code in Visual Basic Script:

1.Help Inquiry Form code

2.Assigned Help Task Form

1.Help Inquiry Form code:

```
'**************************************************************************
'*Help Inquiry code
'*Procedures:
'*      Submit_Click
'*      Item_Open()
'*      Item_CustomPropertyChange(byval string)
'*      Item_PropertyChange(byval string)
''**************************************************************************
Dim UserName            'script level variables used for the ticket ID
Dim Trimmed
Dim Munged
Dim TicketID
'******************************************************************************************
'* Procedure:  Item_Open()
'* Description: Upon open, create a unique ID if the item has not been
'*                   sent (done by checking the SenderName property.)
'******************************************************************************************
Sub Item_Open()
      Dim MyDate
      Dim NowID

      If (Item.SenderName = "") Then          'if item is submitted
            UserName = Application.GetNameSpace("MAPI").CurrentUser
            'Trim the user name of spaces
            Trimmed = TrimUserName(UserName)
            NowID = Now
            'Get date as numbers only
            MyDate = CreateDateAsNumber(NowID)
            'Set the ticket id for this item
            TicketID = MyDate & Trimmed
            Item.userproperties.find("TicketID").value = TicketID
      End If
End Sub
'*******************************************************************************
'* Procedure:  Item_CustomPropertyChange()
'* Description: If the property changed is "Problem type", call
'*                  ChangeProblemType().
'*************************************************************************
Sub Item_CustomPropertyChange(byval ChangeProperty)

      Select Case ChangeProperty
            Case "Problem type"
                  Call ChangeProblemType()
      End Select
End Sub

'*************************************************************************
'* Procedure:  ChangeProblemType()
'* Description: If the Problem Type is changed, clear and repopulate
'*                  the AssignedTo Technician box with the corresponding
'*                  names.
'*************************************************************************
Sub ChangeProblemType()

      Set MyPage = item.getinspector.modifiedformpages.add("Ticket")
      MyPage.controls("AssignedTo").Clear

      Select Case item.userproperties.find("Problem type").Value
            Case "Hardware"
                  MyPage.controls("AssignedTo").AddItem("Thor Nylander")
                  MyPage.controls("AssignedTo").AddItem("Per Johnsen")
                  Set item.userproperties.find("AssignedTo").value = "Thor Nylander"
            Case Else
                  MyPage.controls("AssignedTo").AddItem("Hans Christian Hanestad")
                  MyPage.controls("AssignedTo").AddItem("Thor Nylander")
                  Set item.userproperties.find("AssignedTo").value = "Hans Christian Hanestad"
      End Select
End Sub

'*************************************************************************
'* Procedure:  TrimUserName(ByVal UserName)
'* Description:        Trims the username so that it can be used in the Ticket ID
'*************************************************************************
Function TrimUserName(ByVal UserName)
Dim Counter          ' declare variables
Dim AscName
Dim KeepChar
Dim KeepName
Dim DiscardChar
Dim RightName
Temp = ""
KeepName = ""
For li = 0 To 2        ' for first three letters of user name
    CharName = Mid(UCase(UserName), li + 1, 1)
    AscName = Asc(CharName)
    If ((AscName >= 48) and (AscName <= 57)) or ((AscName >= 65) and (AscName <=90))
Then
        KeepName = KeepName + CStr(AscName)
    End If
Next
RightName = Right(KeepName, 6)
TrimUserName = RightName
End Function
'*************************************************************************
'* Procedure:   CreateDateAsNumber(ByVal NowID)
'* Description: Manipulates the date so that it can be used in the
'*              Ticket ID.
'*************************************************************************
Function CreateDateAsNumber(ByVal NowID)

    For li = 0 To 16            ' for all 17 charachters of the date stamp
        CharName = Mid(UCase(NowID), li + 1, 1)
        AscName = Asc(CharName)
        If ((AscName >= 48) and (AscName <= 57)) Then
            KeepName = KeepName + CStr(CharName)
        End If
    Next
    RightID = Right(KeepName, 6)
    CreateDateAsNumber = RightID
End Function
'****************************************************************************************************
'* Procedure:   Submit_Click
'* Description: When Submit is clicked, run the GTACreateOutlookReminder.dll,
'*              set properties and finally delete this item. The Dll automates both
'*              IE to get the GTA processString and Outlook to create a new Task item
'*****************************************************************************************************
Sub Submit_Click
Dim AssignedTask
```

```
        Assign = MsgBox ("This will create a task in the Assigned Tasks folder, delete this item, and
send mail to the technician indicating that he has a task to complete",1,"Create Task")
        If Assign = 1 Then
                'Set the GTACreateOutlookReminder.dll
                Set AssignedTask =
                item.Application.CreateObject("GTACreateOutlookReminder.GTA_OR")
                        If Err.Number <> 0 Then
                                msgbox err.description & " Could not find dll"
                                Exit sub
                        End if
                ' Create GTALoginString
                AssignedTask.Login_Command =
                "http://winston/servlet/GTAServlet?type=login&username=" &
                item.userproperties.find("AssignedTo").Value  & "&password=dummy"
                ' Create GTATaskReminderString
                AssignedTask.Reminder_Command =
                ("http://winston/servlet/GTAServlet?type=createOutlookReminder&toMyself=true&proces
                sName= & item.userproperties.find("Process").Value
                ' Set all dll property values from the item field values
                AssignedTask.Product = item.userproperties.find("Product").Value
                AssignedTask.Technician = item.userproperties.find("AssignedTo").Value
                AssignedTask.CompMod = item.userproperties.find("Computer Brand and Model").Value
                AssignedTask.AssignedOn = item.userproperties.find("Start by").Value
                AssignedTask.Description = item.body
                AssignedTask.Problemtype = item.userproperties.find("Problem Type").Value
                AssignedTask.Phone = item.userproperties.find("Phone Number").Value
                AssignedTask.OS = item.userproperties.find("Operating System").Value
                AssignedTask.TicketID = item.userproperties.find("TicketID").Value
                AssignedTask.Subject = item.subject
                AssignedTask.ReceivedDate = item.sentOn
                AssignedTask.CompSoft= item.userproperties.find("Software").Value
                AssignedTask.FromUser = item.sendername
                AssignedTask.UserLoc = item.userproperties.find("Request Location").Value
                AssignedTask.TaskPrior = item.userproperties.find("Task Priority").Value
                AssignedTask.Started = True

                Item.delete  ' after everything is set, delete this item. The GTACreateOutlookReminder.dll
                             ' creates a GTA taskreminder and populates data in the new Assigned Help
                             ' Task
        End If
    End Sub
```

2.Assigned Help Task Form :

```
'**********************************************************
'Assigned Task code
'Procedures:
'       Item_Write()
'       Item_Open()
'       Item_PropertyChange(byval string)
'       Item_CustomPropertyChange(byval string)
'***********************************************************
dim Item_Changed  ' Sets if any property changed
dim Property_Changed
dim Priority_Changed
dim Status_Changed
dim Complete_Changed

dim orig_Priority     ' stores the values upon opening the form
dim orig_Status
dim orig_Complete
dim MButton_or_StatClosed
dim SentMail_Flag

dim dbe          'script level database variables
dim MyDB
dim RS
dim dbOpenTable

dim OriginalFolder
dim Ticket_Resolved

Item_Changed = False         'must initially set global variables
Priority_Changed = False
Status_Changed = False
Complete_Changed = False
MButton_or_StatClosed = False
SentMail_Flag = False
orig_Priority = "False"
orig_Status = "False"
orig_Complete = "False"



'*******************************************************************************
```

```
'* Procedure:  Item_Open
'* Arguments:
'* Description: Upon opening, stores the current folder location
'*              into a variable. On creation of the item, set the
'*              history log to list opening information.  Store the
'*              initial values of certain fields for use in later
'*              functions. Load GTA work process
'*******************************************************************************
sub Item_Open()
dim MyPage
dim Resolved

call GTAload

Ticket_Resolved = False

'checks the value of some fields that raise "changes" in other fields:

    if (item.userproperties.find("Flag").Value = "Opened") then
        item.userproperties.find("History Text").Value = chr(164) & " Ticket Received on " &
        item.userproperties.find("Received Date").Value

        If (item.userproperties.find("Description").Value <> "") Then
            item.userproperties.find("History Text").Value = chr(164) & " User Description of
            Problem - " & item.userproperties.find("Description").Value & chr(10) &
            item.userproperties.find("History Text").Value
        Else
            item.userproperties.find("History Text").Value = chr(164) & " No User description of
            problem" & chr(10) & item.userproperties.find("History Text").Value
        End If
        item.userproperties.find("Flag").Value = "Done"
        item.Status = 0
    End If

    Resolved=item.Status

    ' lock all values for the fields when the task is "Resolved"
    If (Resolved=2) Then    ' 2=closed
        set MyPage = item.GetInspector.modifiedformpages.Add("Ticket")
        MyPage.controls("Status").Locked=True
        MyPage.controls("Priority").Locked=True
        MyPage.controls("CompletePercent").Locked=True
```

```
                MyPage.controls("NewLogEntry").Locked=True
                MyPage.controls("TechnicianName").Locked=True

        End If

' sets variables to store the initial information
        orig_Priority = item.userproperties.find("Task Priority").value
        orig_Complete = item.PercentComplete
        orig_Status = item.Status
End Sub

'****************************************************************************
'* Function:     Item_PropertyChange
'* Arguments:
'* Description: Looks if any non-user defined Task fields have
'*               changed. In this case, Status is the only
'*               field affected.
'*****************************************************************************
Sub Item_PropertyChange(byval myProperty)  ' Checks if Status or % complete changed
        Property_Changed = True

        select case myProperty
                case "PercentComplete"
                        call Set_Complete_Changed()
                case "Status"
                        call Set_Status_Changed()
                case else
                        Property_Changed = False   ' if it wasn't changed, set
        end select                               ' flag to FALSE
end Sub

'*******************************************************************
'* Function:     Item_CustomPropertyChange()
'* Arguments:  myProperty - string
'* Description: Checks if any user-defined fields have changed.
'*               Only Task Priority and Complete Percent are
'*               affected.
'*******************************************************************



sub Item_CustomPropertyChange(byval myProperty)
```

```
        Item_Changed=True           ' initially sets the Changed flag

        select case myProperty
            case "Complete Percent"
                call Set_Complete_Changed()
            case "Task Priority"
                call Set_Priority_Changed()
            case else
                Item_Changed=False    ' if nothing changes, set flag
        end select                    ' as FALSE

    end sub

    '*****************************************************************************************
    '* Function:     Item_Write()
    '* Arguments:
    '* Description: Writes all changed fields into the History Log
    '*               and creates a record in the History tracking database.
    '*****************************************************************************************
    sub Item_Write()
    dim Priority_Conversion

        'if the status, priority, %complete, or severity have changed, enter these into the log

        if (Item_Changed or Property_Changed) Then
            LogEntry = ""                 'Clear the initial LogEntry

            if (Priority_Changed) then
                if LogEntry <> "" then
                    LogEntry = LogEntry & ", "  'commas in LogEntry after each entry
                end if

                ' assign the Task Priority value
                select Case item.userproperties.find("Task Priority").Value
                    Case "Normal"     ' to display the default Priority icon(! or ) upon save
                        item.importance = 1
                    Case Else
                        item.importance = 2
                End Select
                LogEntry = LogEntry & "Priority Changed to " & item.userproperties.find("Task
                Priority").Value
                orig_Priority = item.userproperties.find("RPriority").Value
                Priority_Changed = False

            end if

            if (Complete_Changed) Then   ' check %Complete
                if LogEntry <> "" then
                    LogEntry = LogEntry & ", "
                end if
                LogEntry = LogEntry & "%Complete Changed to " & (item.PercentComplete) & "%"
                orig_Complete = item.PercentComplete
                Complete_Changed = False
            End If

            if (Status_Changed)  then
                if LogEntry <> "" then
                    LogEntry = LogEntry & ", "
                end if
                If (Orig_status = 2) Then
                    item.userproperties.find("Close Date").Value = "None"
                End If

            Select Case item.Status ' convert the enumerated value of Status into a text
    representation for the History log
                Case 0
                    Status_Conversion = "Not Started"
                Case 1
                    Status_Conversion = "In Progress"
                Case 2
                    Status_Conversion = "Completed"
                Case 3
                    Status_Conversion = "Waiting on someone else"
                Case 4
                    Status_Conversion = "Deferred"
            End Select

                LogEntry = LogEntry & "Status Changed to " & Status_Conversion
                orig_Status = item.Status
                Status_Changed = False
            end if
```

```
        MailEntryLog = ""

        If (SentMail_Flag) Then ' it resolution mail was sent, report in log
            MailEntryLog = "Resolution Mail sent to user on " & Now() & chr(13) & chr(10)
            SentMail_Flag = False
        End If

        'write history log

        HistoryLog = item.userproperties.find("History Text").Value
        NewHistoryLog = Now() & ": " & Application.GetNamespace("MAPI").CurrentUser & " - "
        & LogEntry
        NewHistoryLog = chr(164) & " " & NewHistoryLog & chr(10)
        item.userproperties.find("History Text").Value = MailEntryLog & NewHistoryLog &
        HistoryLog

        If (Ticket_Resolved) then      'when ticket is Resolved
            item.userproperties.find("History Text").Value = chr(164) & " Ticket Closed on " &
            Now() & chr(10) & item.userproperties.find("History Text").Value
            Ticket_Resolved = False
        End If

        Item_Changed=False
    end if

    ' checks if a record is already created, then add or uppdate record in the database

    if item.userproperties.find("Database record created") = "No" then
        Call AddNewDatabaseRecord
        item.userproperties.find("Database record created") = "Yes"
            else
            if item.userproperties.find("Database record created") = "Yes" then
            call UpdateDataBaseRecord
            end if
    end if
end sub
```

```
'*********************************************************************************
'* Procedure:  Set_Priority_Changed(), Set_Status_Changed()...
'* Arguments:
'* Description: These procedures compare the values of specific
'*              fields with their original values (upon opening).
'*              If they are different, then the History Log is affected
'*********************************************************************************
Sub Set_Priority_Changed()
'if the user changes the value of priority
'flag it as dirty and to be logged.
dim x           'temp variables
dim y

     x = orig_Priority
     y = item.userproperties.find("Task Priority").Value

     Priority_Changed = (x <> y)

End Sub


'****************************************************************
Sub Set_Complete_Changed()
' same as above function, but check %Complete
****************************************************************
dim x           'temp variables
dim y

     x = orig_Complete
     y = item.PercentComplete

          Complete_Changed = (x <> y)

     if (y > 100) Then
          item.PercentComplete = "100"
     Else If (y < 0) Then
          item.PercentComplete = "0"
     End If
     End If

End Sub
```

```
'*****************************************************************
Sub Set_Status_Changed()
*****************************************************************
dim x           'temp variables
dim y
     x = orig_Status
     y = item.Status

     Status_Changed = (x <> y)

     If (y = 2) then
          item.PercentComplete = "100"
     End If

End Sub

'***************************************************************************
'* Procedure:  SendButton_Click
'* Arguments:
'* Description: Sends mail to end user. Calls the Correspondence
'*              Mail item and populates with the neccessary fields,
'*              then displays the form.
'***************************************************************************
Sub SendButton_Click

     Item.save
     Set AssignedFolder = item.parent
     Set CorrespondenceItem = AssignedFolder.Items.Add("IPM.Note.Correspondence Mail")
     CorrespondenceItem.To = item.userproperties.find("From User").Value
     CorrespondenceItem.Subject = Item.Subject
     CorrespondenceItem.UserProperties.Find("Ticket ID").Value =
     item.UserProperties.Find("Ticket ID").Value
     CorrespondenceItem.UserProperties.Find("AssignedEntryID").Value = Item.EntryID
    CorrespondenceItem.UserProperties.Find("AssignedStoreID").Value = Item.Parent.StoreID
     CorrespondenceItem.Display

End Sub
```

```
'*************************************************************************
'* Procedure:  SendMailToUser()
'* Arguments:
'* Description: Upon exiting when resolved, this
'*              automatically sends a mail message to the end
'*              user, stating their problem is now resolved.
'*************************************************************************
Sub SendMailToUser()
Dim myFolder
Dim MailMsg
Dim myItem

     Set myItem = Application.CreateItem(0)   'creates native Mail form

     myItem.To = item.userproperties.find("From User").Value          'populate the native fields
     myItem.Subject = "Problem - resolved"
     myItem.Body = Item.Subject & " -- Your request was resolved by the Help Desk at
     " & Now() & "."
     myItem.Send

     SentMail_Flag = True
End Sub
'**************************************************************************
'* Procedure:  Resolve_Click
'* Arguments:
'* Description: When technician finally decides to mark the task
'*              as resolved, "Status" and "%Complete" are set to
'*              Closed and 100, respectively, the form is saved,
'*              and mail is sent to the user.
'**************************************************************************
Sub Resolve_Click
dim User_item

     User_item = MsgBox("This will send a mail message to the user indicating that the problem is
     resolved." & chr(13) & Chr(10) & chr(13) & chr(10) & " Would you like to save and close this
     form?",4,"Completed")

     On error resume next

     If User_item=6 Then           ' item 6 is answer "YES" to
                                  ' above Message Box question
```

```
                item.Status = 2
                item.PercentComplete = "100"
                item.userproperties.find("Close Date").Value = Now()
                Ticket_Resolved = True
                Call SendMailToUser()
                item.Save

                item.close(0)
        End If

End Sub

'***************************************************************************
'* Procedure:  AddtoHistory_Click
'* Arguments:
'* Description: Each time the "Add to History" button is clicked,
'*              the contents of the New Log Entry are copied into
'*              the History Log, and the New Log Entry is cleared.
'***************************************************************************
Sub AddtoHistory_Click
dim HistoryLog          'Temp for the old history log
dim NewHistoryLog       'Temp for the new History log
dim LogEntry            'Temp for the new history log entry line


'if the New Log Entry field is not empty, record the description in the log
        'and clear out the description

        if len(item.userproperties.find("New Log Entry").Value) > 0 then
                HistoryLog = item.userproperties.find("History Text").Value
                NewHistoryLog = Now() & ": " & Application.GetNamespace("MAPI").CurrentUser & " - "
                & item.userproperties.find("New Log Entry").Value
                NewHistoryLog = NewHistoryLog & chr(13) & chr(10)
                item.userproperties.find("History Text").Value = chr(164) & " " & NewHistoryLog &
                HistoryLog
                item.userproperties.find("New Log Entry").Value = ""
        end if

End Sub
```

```
'***************************************************************************
'* Procedure:  Discussion_Click
'* Arguments:
'* Description: Starts a discussion on the problem or request in
'*              the task folder.
'***************************************************************************
Sub Discussion_Click
      item.Save
      item.actions("Discussion").Execute
End Sub

'************************************************************************
'* Procedure:  Item_CustomAction
'* Arguments:
'* Description: Executes when a custom action is executed
'************************************************************************
Function Item_CustomAction(ByVal myAction, ByVal myResponse)
   Select Case myAction.Name
      Case "Discussion"
            myResponse.Body = "For more information about this ticket, double-click the link." &
            chr(10)
            myResponse.display
      Case Else
   End Select
End Function

'****************************************************************************
'Procedure: AddNewDatabaseRecord
'Description: Adds a new record to the History Tracking database
'for the Document Tracking item.
'****************************************************************************

Sub AddNewDatabaseRecord
  dbOpenTable = 1
  on error resume next
  Set dbe = item.Application.CreateObject("DAO.DBEngine.35")
      if Err.Number <> 0 then
            msgbox err.description & " -- Some functions may not work correctly" & Chr(13) &
            "Please make sure that DAO 3.0 or greater is installed on this machine"
            exit sub
      end if
      Set MyDB = dbe.Workspaces(0).OpenDatabase("c:\my documents\history tracking.mdb")
```

```
        Set Rs = MyDB.OpenRecordset("History Tracking Information", dbOpenTable)
        Rs.Addnew
        'RS("Status") = item.status
        Call FieldValues
        Rs.Update
        Rs.MoveLast
        Rs.Close
        MyDB.Close
End Sub
'*************************************************************************
'Procedure: UpdateDataBaseRecord
'Description: Adds a new record to the History Tracking database
'for the Document Tracking item.
'*************************************************************************
Sub UpdateDataBaseRecord

  dbOpenTable = 1
  on error resume next
  Set dbe = item.Application.CreateObject("DAO.DBEngine.35")
        if Err.Number <> 0 then
        msgbox err.description & " -- Some functions may not work correctly" & Chr(13) & "Please
        make sure that DAO 3.0 or greater is installed on this machine"
        exit sub
        end if
        set MyDB = dbe.Workspaces(0).OpenDatabase("c:\my documents\history tracking.mdb")
        Set Rs = MyDB.OpenRecordset("History Tracking Information", dbOpenTable)
        'rs.Index = "Ticket ID"    ' Define current index.
        'Rs.Seek "=", Item.userproperties.find("Ticket ID").value
        Rs.Edit
        'RS("Status") = item.Status
        Call FieldValues
        Rs.Update
        Rs.MoveLast
        Rs.Close
        MyDB.Close
End Sub
```

```
'*************************************************************************************
'Procedure: FieldValues
'Description: Calls the CheckValue Function and passes it the name of the
'form field and the data field.
'*************************************************************************************

Sub FieldValues
        on error resume next
        CheckValue "Ticket ID", "TicketID"
        CheckValue "Technician Name", "Technician"
        CheckValue "% Complete", "% Complete"
        CheckValue "Task Priority", "Priority"
        CheckValue "Problem Type", "Problem Type"
        CheckValue "Description", "Description"
        CheckValue "Received Date", "Received"
        CheckValue "Customer", "Customer"
        CheckValue "CustomerID", "CustomerID"
        CheckValue "Cxcontact", "Cxcontact"

End Sub

'************************************************************************
'Procedure: CheckValue
'Description: Checks the field for valid data. It it exists, write the
'field value to the database.
'************************************************************************

Sub CheckValue (formfield, dbfield)
  if not userproperties.find(formfield) is nothing then
    if userproperties.find(formfield).value <> "" then
      if isdate(userproperties.find(formfield).value) then
        if userproperties.find(formfield).value <> "1/1/4501" then
                rs(dbfield) = userproperties.find(formfield).value
           end if
        else
          rs(dbfield) = userproperties.find(formfield).value
        end if
    end if
  end if
End Sub
```

```
'******************************************************************************************
'Procedure: GTAload
'Description:   Called by Item_Open. Uses the CXGTAforOutlook.ocx
'*              to load the GUI of the GTA process
'******************************************************************************************

Sub GTAload

    Dim LoginString     'Http commandString variables
    Dim WorklistString
    Dim GtaOutlookString
    ' Creates LoginString, use the technician name
    LoginString = "http://winston/servlet/GTAServlet?type=login&username=" &
    item.userproperties.find("Technician Name").Value  & "&password=dummy"
    WorklistString = "http://winston/servlet/GTAServlet?type=worklists"
    ' Creates GTAOutlookString, use the ProcessID
    GtaOutlookString = "http://winston/gta/gtaApp/gtaOutlook.html?type=process&process=" &
    item.userproperties.find("ProcessID").Value
    ' Set CXGTAforOutlook.ocx  properties
    Set Mypage = item.GetInspector.ModifiedFormPages("Ticket")
    Set MyControls = Mypage.Controls
    MyControls("CXGTA1").Login_Command = LoginString
    MyControls("CXGTA1").Worklist_Command = WorklistString
    MyControls("CXGTA1").Process_Command = GtaOutlookString

End Sub
```

Appendix B

1.“CXGTAforOutlook.ocx” code

2. “GTACreateOutlookReminder.dll” code

1.“CXGTAforOutlook.ocx” :

```
Option Explicit
Private LoginVar As String
Private WorklistVar As String
Private ProcessVar As String
Private StartedVar As Boolean
Private Sub UserControl_Resize()
    ' Resize the WebBrowser3 Control (I.e the work process window)
      'to fill MyControl's visible surface area
  WebBrowser3.Move 0, 0, ScaleWidth, ScaleHeight
  Debug.Print "Resize"
End Sub
Private Sub UserControl_Initialize()
      Debug.Print "Initialize"
End Sub
Private Sub UserControl_InitProperties()
   Debug.Print "InitProperties"
End Sub
Private Sub UserControl_WriteProperties(PropBag _
     As PropertyBag)
  Debug.Print "WriteProperties"
End Sub
Private Sub UserControl_Terminate()
  WebBrowser3.Navigate ("http://winston/servlet/GTAServlet?type=logout")
  Debug.Print "Terminate"
End Sub
Private Sub UserControl_ReadProperties(PropBag As _
     PropertyBag)
 Debug.Print "ReadProperties"
End Sub
Public Property Get Login_Command() As String
```

```
   Login_Command = LoginVar
End Property

Public Property Let Login_Command(ByVal NewLogin As String)
   LoginVar = NewLogin
   PropertyChanged "Login_Command"
   End Property
Public Property Get Worklist_Command() As String
   Worklist_Command = WorklistVar
End Property
Public Property Let Worklist_Command(ByVal NewWorklist As String)
   WorklistVar = NewWorklist
   PropertyChanged "Worklist_Command"
End Property
Public Property Get Process_Command() As String
   Process_Command = ProcessVar
End Property
Public Property Let Process_Command(ByVal NewProcess As String)
   ProcessVar = NewProcess
   PropertyChanged "Process_Command"
End Property
Public Property Let Started(ByVal NewValue As Boolean)
   StartedVar = NewValue
   PropertyChanged "Started"
      Call NavigateStart
 End Property

Private Sub NavigateStart()
      WebBrowser1.Visible = True
      WebBrowser1.Navigate Login_Command
End Sub
Private Sub WebBrowser1_DocumentComplete(ByVal pDisp As Object, URL As Variant)
 If (pDisp Is WebBrowser1.object) Then
      WebBrowser2.Visible = True
      WebBrowser2.Navigate Worklist_Command
```

```
        Debug.Print "Login is finished loading."
 End If

End Sub

Private Sub WebBrowser2_DocumentComplete(ByVal pDisp As Object, URL As Variant)
 If (pDisp Is WebBrowser2.object) Then
        WebBrowser3.Visible = True
        WebBrowser3.Navigate Process_Command
        Debug.Print "Worklist is finished loading."
  End If

End Sub
Private Sub WebBrowser3_DocumentComplete(ByVal pDisp As Object, URL As Variant)
    If (pDisp Is WebBrowser3.object) Then
        Debug.Print "Process is finished loading."
        Exit Sub
    End If
End Sub
```

2. “GTACreateOutlookReminder.dll”:

```
Option Explicit
' Automate Internet Explorer and receive events
Dim WithEvents IE1 As InternetExplorer
Dim WithEvents IE2 As InternetExplorer
Private LoginVar As String
Private ReminderVar As String
Private StartedVar As Boolean
Private FinishVar As Boolean
Private ReadyStateVar As String
Private ReadyStateVar2 As String
'Variables for the new Assigned task item
Private ProcessIDVar As String
Private TechnicianVar As String
Private CompModVar As String
Private ProductVar As String
Private ProblemtypeVar As String
Private PhoneVar As String
Private OSVar As String
Private TicketIDVar As String
Private SubjectVar As String
Private DescriptionVar As String
Private ReceivedDateVar As String
Private CompSoftVar As String
Private FromUserVar As String
Private UserLocVar As String
Private TaskPriorVar As String
Private AssignedOnVar As String
Private AssignedToVar As String

Public Property Get Login_Command() As String
   Login_Command = LoginVar
End Property
Public Property Let Login_Command(ByVal NewLogin As String)
   LoginVar = NewLogin
End Property
Public Property Get Reminder_Command() As String
   Reminder_Command = ReminderVar
End Property
Public Property Let Reminder_Command(ByVal NewReminder As String)
   ReminderVar = NewReminder
End Property
Public Property Let Started(ByVal NewValue As Boolean)
   StartedVar = NewValue
   ' When all properties is set, start navigate
   Call NavigateStart
End Property
Public Property Let Finished(ByVal NewValue As Boolean)
   FinishVar = NewValue
   ' When the processID is "received" from IE, create new Assigned task.
   Call CreateAssignedTask
End Property

Public Property Let ProcessID(ByVal NewProcessID As String)
   ProcessIDVar = NewProcessID
End Property
Public Property Get ProcessID() As String
   ProcessID = ProcessIDVar
End Property
Public Property Let Technician(ByVal NewTechnician As String)
```

```
    TechnicianVar = NewTechnician
    End Property
Public Property Get Technician() As String
    Technician = TechnicianVar
End Property
Public Property Let CompMod(ByVal NewCompMod As String)
    CompModVar = NewCompMod
End Property
Public Property Get CompMod() As String
    CompMod = CompModVar
End Property
Public Property Let Product(ByVal NewProduct As String)
    ProductVar = NewProduct
End Property
Public Property Get Product() As String
    Product = ProductVar
End Property
Public Property Let Problemtype(ByVal NewProblemtype As String)
    ProblemtypeVar = NewProblemtype
End Property
Public Property Get Problemtype() As String
    Problemtype = ProblemtypeVar
End Property
Public Property Let Phone(ByVal NewPhone As String)
    PhoneVar = NewPhone
End Property
Public Property Get Phone() As String
    Phone = PhoneVar
End Property
Public Property Let OS(ByVal NewOS As String)
    OSVar = NewOS
End Property
Public Property Get OS() As String
    OS = OSVar
End Property
Public Property Let TicketID(ByVal NewTicketID As String)
    TicketIDVar = NewTicketID
End Property
Public Property Get TicketID() As String
    TicketID = TicketIDVar
End Property
Public Property Let Subject(ByVal NewSubject As String)
```

```
  SubjectVar = NewSubject
End Property
Public Property Get Subject() As String
  Subject = SubjectVar
End Property
Public Property Let Description(ByVal NewDescription As String)
  DescriptionVar = NewDescription
End Property
Public Property Get Description() As String
  Description = DescriptionVar
End Property
Public Property Let ReceivedDate(ByVal NewReceivedDate As String)
  ReceivedDateVar = NewReceivedDate
End Property
Public Property Get ReceivedDate() As String
  ReceivedDate = ReceivedDateVar
End Property
Public Property Let CompSoft(ByVal NewCompSoft As String)
  CompSoftVar = NewCompSoft
End Property
Public Property Get CompSoft() As String
  CompSoft = CompSoftVar
End Property
Public Property Let FromUser(ByVal NewFromUser As String)
  FromUserVar = NewFromUser
End Property
Public Property Get FromUser() As String
  FromUser = FromUserVar
End Property
Public Property Let UserLoc(ByVal NewUserLoc As String)
  UserLocVar = NewUserLoc
End Property
Public Property Get UserLoc() As String
  UserLoc = UserLocVar
End Property
Public Property Let TaskPrior(ByVal NewTaskPrior As String)
  TaskPriorVar = NewTaskPrior
End Property
Public Property Get TaskPrior() As String
  TaskPrior = TaskPriorVar
End Property
Public Property Let AssignedTo(ByVal NewAssignedTo As String)
```

```
    AssignedToVar = NewAssignedTo
End Property
Public Property Get AssignedTo() As String
    AssignedTo = AssignedToVar
End Property
Public Property Let AssignedOn(ByVal NewAssignedOn As String)
    AssignedOnVar = NewAssignedOn
End Property
Public Property Get AssignedOn() As String
    AssignedOn = AssignedOnVar
End Property

Private Sub NavigateStart()
    ' Automates IE, send login request
    Set IE1 = New InternetExplorer
    IE1.Visible = False
    IE1.navigate Login_Command
    Do Until IE1.readyState = READYSTATE_COMPLETE
        ReadyStateVar = IE1.readyState
        If ReadyStateVar = 3 Then
            Call NavigateReminder
        End If
    DoEvents
    Loop
End Sub

Private Sub NavigateReminder()
    ' Automates IE, sends the createOutlookReminder request
    Set IE2 = New InternetExplorer
    IE2.Visible = False
    IE2.navigate Reminder_Command
    Debug.Print "Login is finished loading."
    Do Until IE2.readyState = READYSTATE_COMPLETE
        ReadyStateVar2 = IE2.readyState
        If ReadyStateVar2 = 3 Then
            Call GetProcessID
        End If
    DoEvents
    Loop
End Sub
```

```
Private Sub GetProcessID
' Get the unique GTAprocessID replyed from the createOutlokkreminder request
        Dim MyHTMLobject As HTMLDocument
        If Not IE2.document Is Nothing Then
            Set MyHTMLobject = IE2.document
            'Get the unique GTAprocessID, which is the title element of the HTML document
            ProcessID = MyHTMLobject.Title
            Debug.Print "Reminder created"
            Finished = True
       End If
End Sub

Private Sub CreateAssignedTask()
    'Automates Outlook. Create a new assigned task in the "Assigned Task Folder"
    'and send a notification message to the receiver
    'Important! The assigned task folder must be copied to the Public Folders Favorites folder
    'If not, the code has to be changed to point at "Assigned Task folder"
    Dim myOlApp As Object
    Dim myNameSpace As Object
    Dim PublicFolders As Object
    Dim AllPublicFolders As Object
    Dim AssignedTaskFolder As Object
    Dim AssignedTaskItem As Object
    Dim myItem As Object
    Dim myAttachments As Object

    Set myOlApp = New Outlook.Application
    Set myNameSpace = myOlApp.GetNamespace("MAPI")
    Set PublicFolders = myNameSpace.Folders("Public Folders") ' Note!
    Set AllPublicFolders = PublicFolders.Folders("Favorites") ' Note!
    Set AssignedTaskFolder = AllPublicFolders.Folders("Assigned Help Tasks")
    Set AssignedTaskItem = AssignedTaskFolder.Items.Add("IPM.Task.Assigned Help Task")

    AssignedTaskItem.UserProperties.Find("Computer Brand and Model").Value = CompMod
    AssignedTaskItem.UserProperties.Find("Problem Type").Value = Problemtype
    AssignedTaskItem.UserProperties.Find("Phone").Value = Phone
    AssignedTaskItem.UserProperties.Find("Computer OS").Value = OS
    AssignedTaskItem.UserProperties.Find("Ticket ID").Value = TicketID
    AssignedTaskItem.Subject = Subject
    AssignedTaskItem.UserProperties.Find("Description").Value = Description
    AssignedTaskItem.UserProperties.Find("Technician Name").Value = Technician
    'AssignedTaskItem.UserProperties.Find("Received Date").Value = ReceivedDate
    AssignedTaskItem.UserProperties.Find("Computer Software").Value = CompSoft
    AssignedTaskItem.UserProperties.Find("From User").Value = FromUser
    AssignedTaskItem.UserProperties.Find("User Location").Value = UserLoc
    AssignedTaskItem.UserProperties.Find("Task Priority").Value = TaskPrior
    AssignedTaskItem.UserProperties.Find("AssignedOn").Value = AssignedOn
    AssignedTaskItem.UserProperties.Find("Product2").Value = Product
    AssignedTaskItem.UserProperties.Find("ProcessID").Value = ProcessID


    'If TaskPriority is something other than normal, set the Importance icon.

    Select Case TaskPrior
       Case "Normal"
          AssignedTaskItem.Importance = 1
       Case Else
          AssignedTaskItem.Importance = 2
    End Select

    AssignedTaskItem.save

    ‘ Send notification message
    Set myItem = myOlApp.CreateItem(0)  ' 0 = message
    myItem.Subject = "Assigned Help Desk Task at  " & Now()
    myItem.body = "You have an assigned Help Desk task: " & Chr(13)

    On Error Resume Next
    myItem.To = AssignedTo
    Set myAttachments = myItem.Attachments
    myAttachments.Add AssignedTaskItem, 4 ' Type of attachment. 4 = olByReference attachment
    myItem.send
    If Err Then
       MsgBox ("The mail did not send because the technician was not recognized. Take a look in
       your address list")
    End If

 End Sub
```

Appendix C

Tools:

- Microsoft Outlook 2000 BETA
    - Microsoft Outlook Forms Help
    - Microsoft Outlook VB Script Editor
- Microsoft Office Test Platform & Development Tools BETA
    - Office 2000 developer
    - Office 2000 Evaluation Materials and
    - MSDN Library – Office 2000 Developer
- Visual Basic 5.0 to create DLL and OCX
- Sara Expert Workflow Tool to design work processes
- Keyfile Corporation: Keyflow evaluation CD

Appendix D

Web resources:

1. http://www.helpdeskinst.com/
2. http://www.helpdesk.com/
3. http://www.rgu.ac.uk/~sim/research/helpdesk/
4. http://www.philverghis.com/
5. http://www.pcsyscom.no/produkt.htm
6. http://officeupdate.microsoft.com/downloaddetails/helpdesk.htm
7. http://www.microsoft.com/workshop/browser/overview/overview.asp
8. http://www.computas.com/
9. http://www.microsoft.com/office/
10. http://www.microeye.com/outlkb.html
11. http://www.hub.nih.gov/exchinfo/default.htm
12. http://www.brsilver.com/
13. http://www.keyfile.com/
14. http://ikaros/sara/ (intranet)
15. http://ikaros/taskassistant (intranet)

16. http://ikaros/mimesisgta/ (intranet)
17. http://www.gigaweb.com/
18. http://ikaros/framesolutions (intranet)

Appendix E

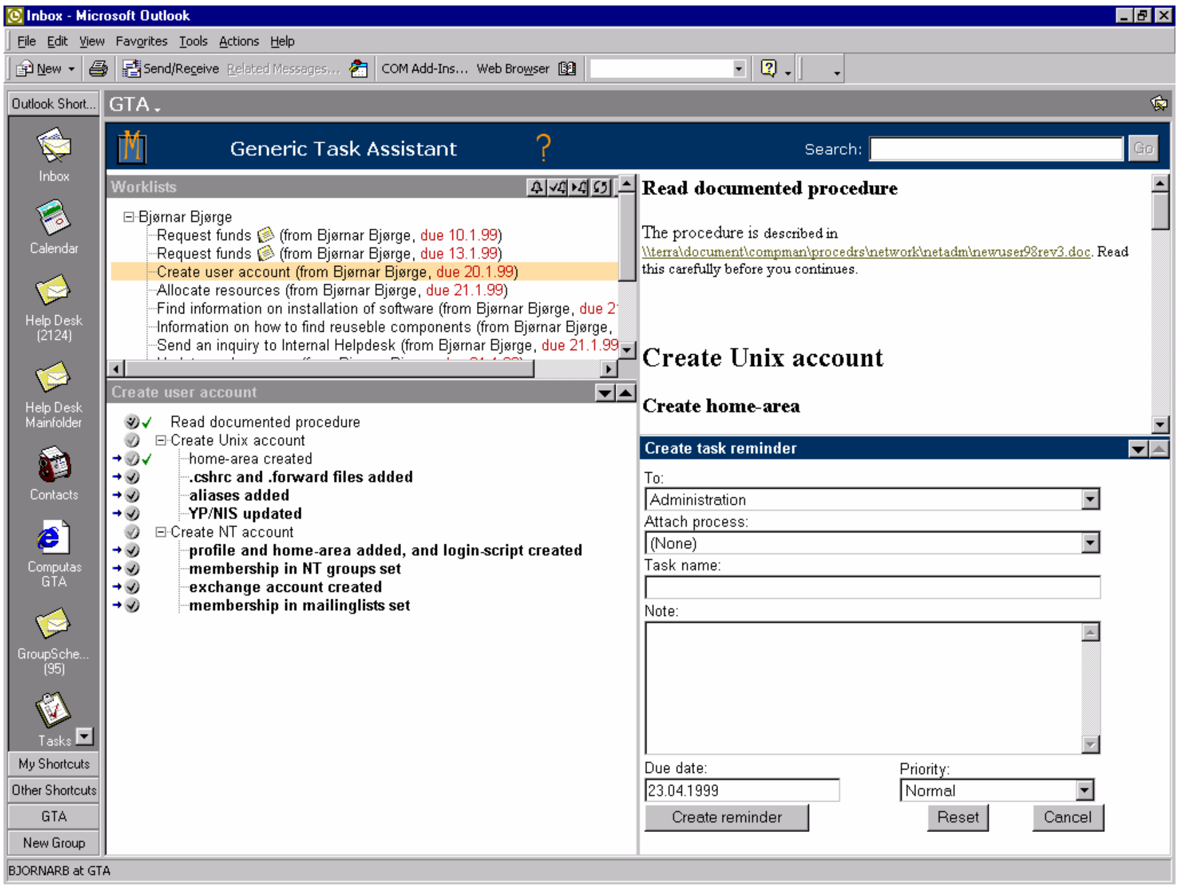
Newsgroups:

1. Microsoft.public.outlook
2. Microsoft.public.outlook.program_forms
3. Microsoft.public.scripting.vbscript
4. Microsoft.public.activex.programming.controls.webdc
5. Microsoft.public.activex.programming.controls.webwiz
6. Help desk: LISTSERV@WVNVM.WVNET.EDU
7. http://experts-exchange.com

Appendix F

GTA in Outlook

Outlook 2000 has a built in Internet Explorer browser, and therefore GTA can be loaded as an application in the Outlook window. For description of GTA client functionality see chapter 7.2.



Bjørnar Bjørge
- Request funds (from Bjørnar Bjørge, due 10.1.99)
- Request funds (from Bjørnar Bjørge, due 13.1.99)
- Create user account (from Bjørnar Bjørge, due 20.1.99)
- Allocate resources (from Bjørnar Bjørge, due 21.1.99)
- Find information on installation of software (from Bjørnar Bjørge, due 2
- Information on how to find reuseble components (from Bjørnar Bjørge,
- Send an inquiry to Internal Helpdesk (from Bjørnar Bjørge, due 21.1.99

✓ Read documented procedure
Create Unix account
- ✓ home-area created
- **.cshrc and .forward files added**
- **aliases added**
- **YP/NIS updated**

Create NT account
- **profile and home-area added, and login-script created**
- **membership in NT groups set**
- **exchange account created**
- **membership in mailinglists set**

Read documented procedure

The procedure is described in \\terra\document\compman\procedrs\network\netadm\newuser98rev3.doc. Read this carefully before you continues.

Create Unix account

Create home-area

To: Administration
Attach process: (None)
Task name:
Note:
Due date: 23.04.1999
Priority: Normal

Create reminder Reset Cancel