

Agent aided aircraft maintenance

Onn Shehory, Katia Sycara, Gita Sukthankar
The Robotics Institute
Carnegie-Mellon University
{onn,katia,gitars}@cs.cmu.edu

May 27, 1999

Abstract

Aircraft maintenance is performed by mechanics who are required, for non-standard discrepancies, to consult expert engineers for repair instructions and approval. In addition to their own experience, these engineers rely on external information sources, which are often inadequately indexed and geographically dispersed. The timely retrieval of this distributed information is essential to the engineers' ability to devise and recommend repair procedures in response to the mechanics' requests. This problem domain is well suited for a multi-agent system: it consists of distributed multi-modal information which is needed by multiple users with diverse preferences. In this paper, we describe an implementation of such a system, using the RETSINA multi-agent architecture. Such an implementation reinforces the importance of multi-agent systems, and in particular the usefulness of the RETSINA infrastructure as a basis for the construction of such systems.

1 Introduction

Agent aided information retrieval and decision support have been a focus of the agent research community for several years. Although theories and simulations of multi-agent information retrieval systems abound, real applications are scarce. One reason for this scarcity is the lack of real problem domains which are sufficiently distributed, rich in information and complexity, without being too complex. Such domains would have facilitated the development of application systems and allowed the examination of the applicability and usefulness of multi-agent technology developed in research labs.¹

In this paper we present such an application. Based on our experience with the RETSINA multi-agent infrastructure [7], we implemented a system to solve an existing real-world problem. Specifically, we developed a multi-agent framework that provides information retrieval and analysis in support of decision making for aircraft maintenance and repair in the U.S. Air Force. Although the solution was developed for a specific type of aircraft, the agents and the interactions among them were designed to apply to a range of similar maintenance scenarios.

Maintenance of complex vehicles such as aircraft and sea-craft is a complicated task. Usually, this task involves several people, including mechanics, inspectors, engineers and possibly other experts. In addition, the amount of information involved in the process is very large. This information is typically multi-modal and available in multiple storage media (e.g., hard-copy and electronic). Commonly, the information is also geographically distributed. A major goal of the maintenance process is to perform the most appropriate repair procedures in the most efficient way and in the shortest time. Since maintenance-related processes rely on relevant information, comprehensive and timely information delivery to the individuals involved in the maintenance can significantly benefit the process.

The paper is organized as follows: we first present the properties of the problem in section 2. Then, in section 3, we describe the RETSINA multi-agent infrastructure, which we used as the basis for the development of our solution. In section 4 we describe the solution we provide for the problem at hand. Finally, in section 5, we discuss the advantages and the limitation of

¹Such research is presented in multiple publications. For examples, surveys and pointers to additional research refer to [2, 3, 4, 5].

the solution and indicate future research directions.

2 The problem

Aircraft maintenance in the U.S. Air Force is performed at several different levels. The basic and intermediate levels are usually performed at the base where the aircraft is deployed, whereas periodic, comprehensive maintenance is performed at specialized depots. In both cases, maintenance is a complex process which involves several procedures, as described below. Initially, mechanics inspect the aircraft for discrepancies (and may also receive such information from pilots). For each discrepancy he/she finds, the mechanic looks-up the technical manuals for a standard repair procedure. In case that such a repair procedure is found and the resources and parts required for it are available, the mechanic proceeds with the repair and files the repair information. In cases where parts are not available or they are too expensive or require too much time and additional machinery for replacement, and in cases where a procedure is not provided in the technical manuals, a mechanic needs to consult an expert engineer. The engineer, in turn, may need to consult external sources of information. These include manuals, historical maintenance data and other, remotely-located experts. Until recently, no automation was introduced to the consultation processes of this information-rich environment, as detailed below. Hard-copy repair manuals are used (by both mechanics and engineers), thus search for relevant information may be time consuming and incomplete. Historical data (e.g., records of previous similar repairs) is scarcely used, since it is stored in paper format with no search mechanisms, usually only kept for short periods of time, and is distributed along remotely located service centers. Expert engineers may be located remotely, and their advice is available by voice or fax messages. These are usually delayed for hours or days. All of these factors contribute to a slow, inefficient inspection and maintenance processes.

The inspection, consultation and repair process consists of the following steps:

1. An aircraft arrives at a maintenance service center, either at its home base or at a depot (depending on the type of maintenance required). In both cases, the maintenance procedures must be completed within a

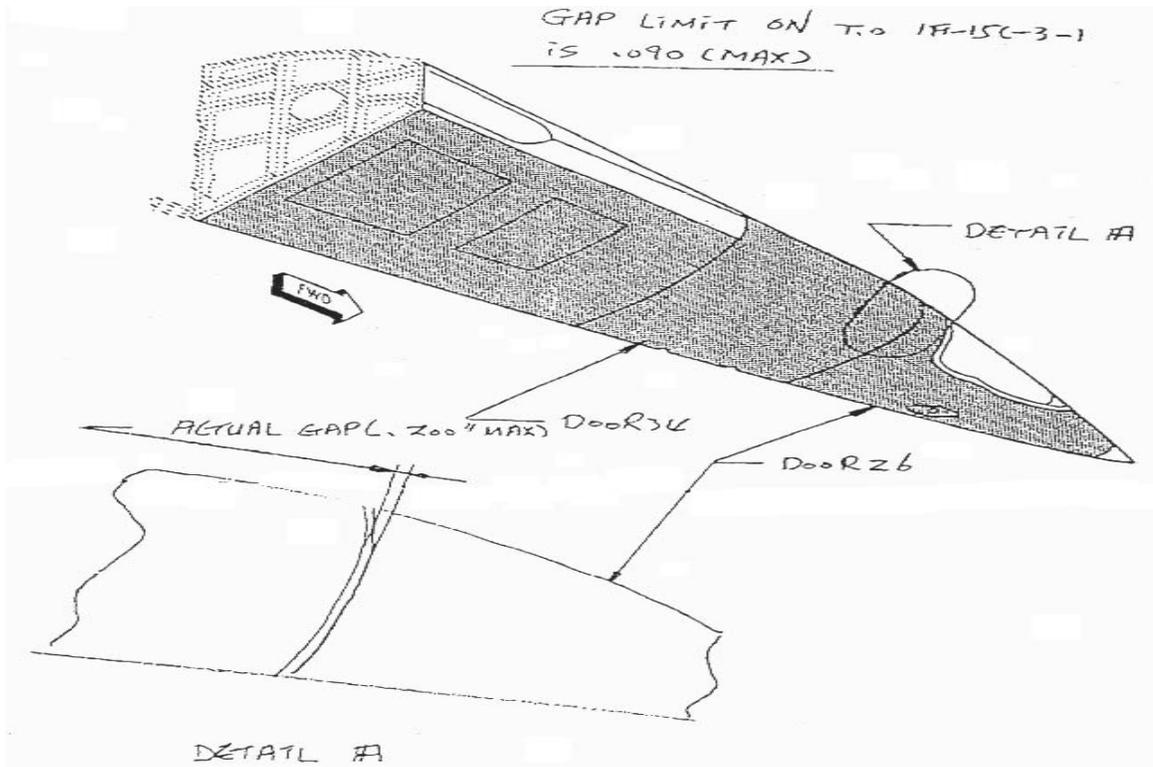


Figure 1: Part of a graphical description attached to a form.

limited time period. This period varies: basic and intermediate maintenance must be completed within hours of a few days, whereas depot maintenance may be scheduled for several weeks (the exact time allocation depends on the aircraft type).

2. Mechanics inspect the aircraft and locate discrepancies. For each discrepancy a mechanic finds, he/she performs the following:
 - Browse the technical manual for repair procedures.
 - In case that an appropriate procedure is located, the mechanic needs to verify whether it can be completed given limitations on repair time and parts' availability. The mechanic may also need to consider the price of the repair (For instance, the technical manual may require replacing a whole wing if a crack in the wing is greater

than some given threshold. This may take too long and be too expensive, and delay or hamper operational activity or readiness. There may be alternative, simpler and cheaper procedures that can provide a solution).

- If the procedure found in the technical manual can be performed, the mechanic performs it (skips to item 4). Otherwise, the mechanic proceeds through consultation, as follows.
- The mechanic fills in a 202a form, which is a standard Air Force form for reporting aircraft discrepancies and requesting advice. To the form, the mechanic may attach supporting information such as graphical illustrations (as in Figure 1). In order to describe the discrepancy in an adequate manner, the mechanic, in addition to relying on his/her experience, consults Illustrated Part Breakdown (IPB) technical manuals (and possibly other, more experienced mechanics).
- The 202a form is sent to an engineer at a depot for advice on the suggested repair and authorization to perform non-standard repair procedures. Since inspection and repair are performed both in field units and in depots, engineers may be located remotely.

3. An engineer, upon receipt of a 202a form, performs the following:

- Uses own experience, historical repair information and technical manuals to find an appropriate repair for the discrepancy described in the 202a form.
- Fills in a 202b form, which is a standard Air Force form for discrepancy repair instructions. To this form the engineer may attach graphical illustration to clarify the required repair procedure.
- Files 202a and 202b forms for future use as historical repair information.

4. When a standard repair procedure is found, or upon receipt of a 202b form from an engineer, the mechanic performs the repair as instructed.

The current inspection, consultation and repair processes, as described above, have several problems. The MAS implementation reported in this paper attempts to address these problems.

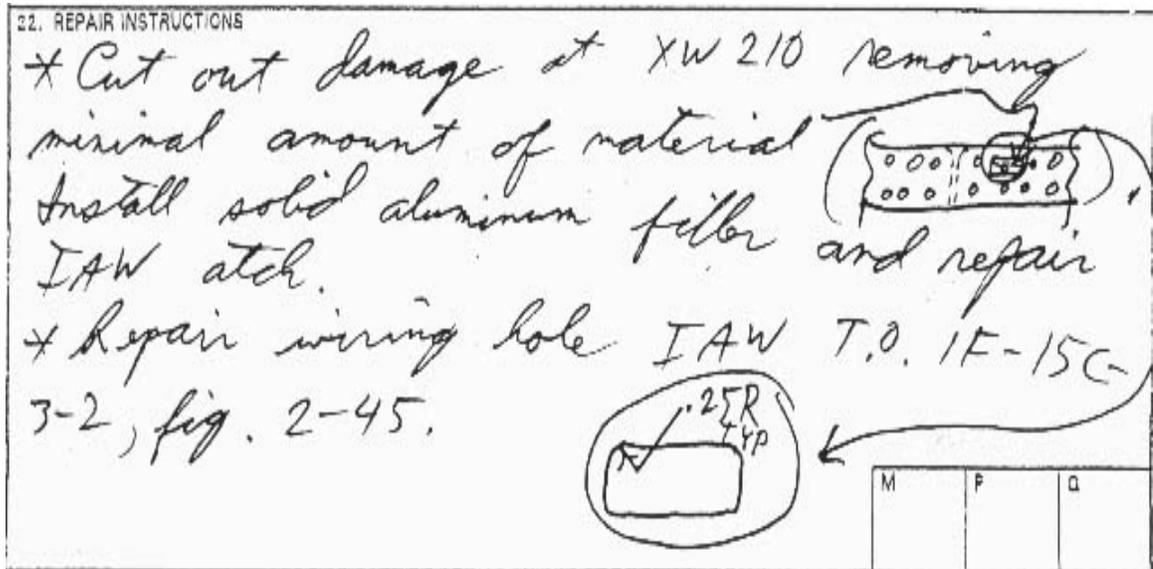


Figure 2: Part of a hand-written 202b form.

- The majority of the information, both historical repair information and technical manuals, is found in hard-copy, and part of it is hand-written.
- Mechanics and engineers spend precious time on:
 - Browsing manuals and searching for historical repair information.
 - Drawing graphical discrepancy and repair illustrations.
 - Mechanics spend time, idle, waiting for 202b forms to arrive from engineers in reply to their 202a forms.
- Old, valuable discrepancy and repair information is not used:
 - When stored in a remote location, historical information is inaccessible.
 - When stored locally, hard-copy information is difficult to browse through, especially when looking for keywords within the free text sections of the 202 forms. With regards to the information needs of mechanics, using paper technical manuals during inspection for

diagnosis is inefficient and at times impossible due to physical constraints of the inspection environment.²

- Hand-written information (as seen in Figure 2), both from historical forms and from the current 202 forms, may have a limited comprehensibility. This problem intensifies due to deterioration in the quality of such information when it is transmitted via fax machines or photo-copied.
- Historical forms are kept only for two years, then disposed.
- Time and effort are spent on paperwork and filing. This time should instead be used for diagnosis and repair.
- As a result of being held in paper format, the information in the technical manuals is not always updated in a consistent manner.

To summarize, the problem with which we deal consists of decision support in a physically distributed and dynamically changing environment, rich in multi-modal information, where users have diverse (and varying over time) information needs. This is the type of problem for which the RETSINA multi-agent system is most appropriate.

3 The RETSINA multi-agent infrastructure

RETSINA [6, 7, 8] (REusable Task-based System of Intelligent Networked Agents) is a multi-agent infrastructure that was developed for information gathering and integration from web-based sources and decision support tasks. It includes a distributed MAS organization, protocols for inter-agent interaction and collaboration, and a reusable set of software components for constructing agents. Each agent in RETSINA specializes in a specific class of tasks. When the agents execute tasks or plan for task execution, they organize themselves to avoid processing bottlenecks and form teams to deal with dynamic changes in information, tasks, number of agents and their capabilities.

²During aircraft maintenance, mechanics may need to enter narrow spaces where using a hard-copy manual is impracticable.

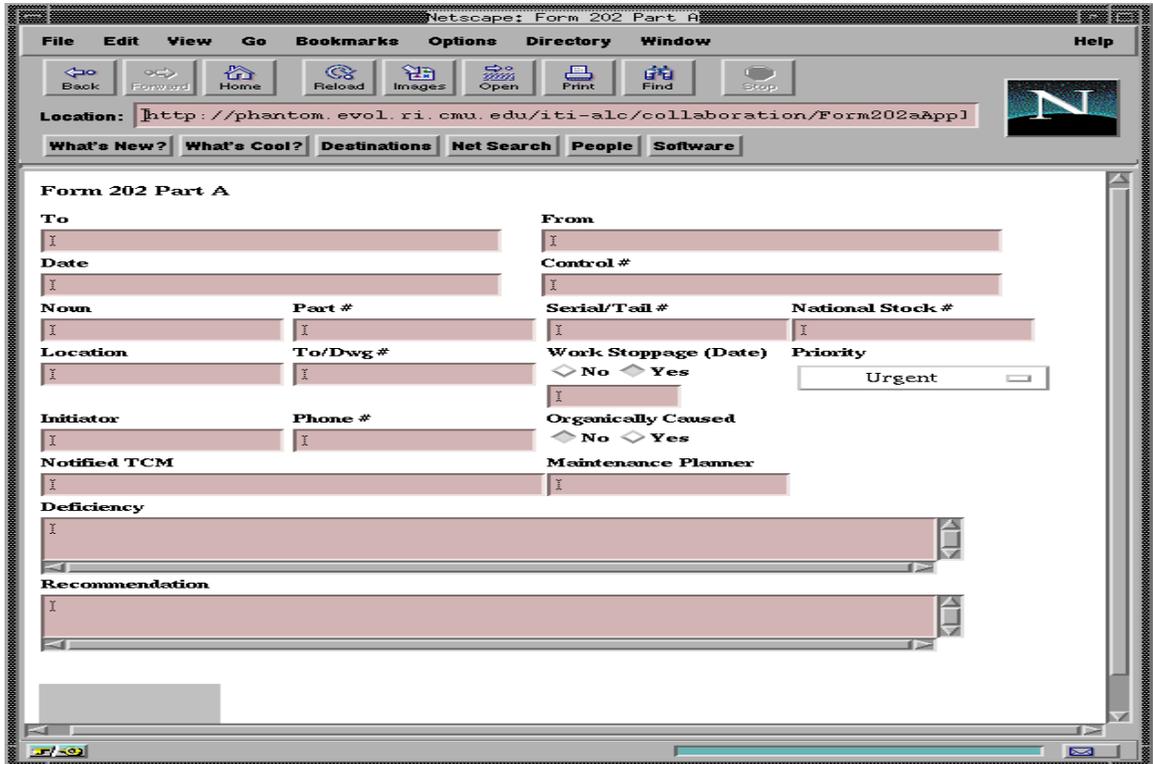


Figure 3: A HTML format 202a form.

In RETSINA, the agents are distributed and execute on different machines. Based on models of users, agents and tasks, the agents decide how to decompose tasks and whether to pass them to others, what information is needed at each decision point, and when to cooperate with other agents. The agents communicate with each other to delegate tasks, request or provide information, find information sources, filter or integrate information, and negotiate to resolve inconsistencies in information and task models. The RETSINA infrastructure consists (by convention) of three broad types of agents:³ *interface* agents, *task* agents and *information* agents. An important sub-type of information agents are middle agents [1]. This organization is depicted in Figure 4.

³Deviations from this strict categorization may occur. For instance, there may be a hybrid of two types, such as interface+task agent.

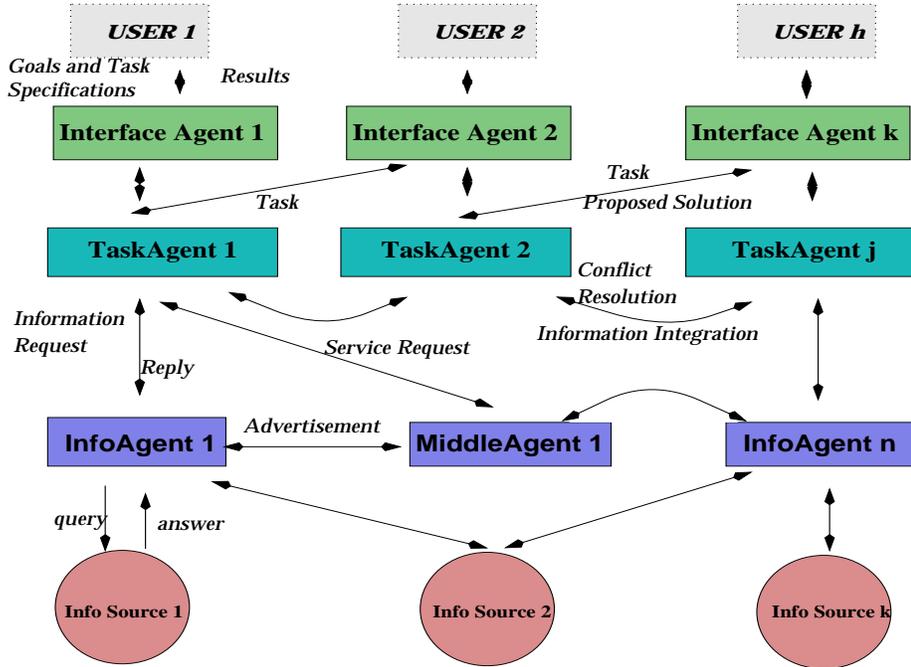


Figure 4: The RETSINA multi-agent infrastructure

Interface agents interact with users receiving their specifications and delivering results. They acquire, model and utilize user preferences. The main functions of an interface agent include: (1) collecting relevant information from the user to initiate a task, (2) presenting relevant intermediate and final results, (3) requesting additional information during task execution. The interface agents hide the underlying structural complexity of the agent system.

Task agents formulate plans and carry them out. They have knowledge of the task domain, and which other task agents or information agents are relevant to performing various parts of the task. In addition, task agents have strategies for resolving conflicts and fusing information retrieved by information agents. A task agent (1) receives user delegated task specifications from an interface agent, (2) interprets the specifications and extracts problem solving goals, (3) forms plans to satisfy these goals, (4) identifies information seeking subgoals that are present in its plans, (5) decomposes plans and cooperates with appropriate task agents or information agents for

plan execution, monitoring and results composition.

Information agents provide intelligent access to a heterogeneous collection of information sources. They have models of the information resources and strategies for source selection, information access, conflict resolution and information fusion. Information agents can actively monitor information sources.

Middle agents collect and provide information about the location, availability and capabilities of other agents, and possibly additional information about them. They may also serve as mediators, hiding the identities of either service requester agents or service provider agents, or both. Middle agents (e.g., matchmakers) provide RETSINA-based MAS with openness. That is, agents may leave and enter the system dynamically. When an agent appears, it advertises itself with a middle agent. When it leaves gracefully, it un-advertises. Agent disappearance as a result of agent or network failure is detected by middle agent via a pinging mechanism.

The RETSINA internal agent architecture is based on a multi-module, multi-thread design. It consists of two component types: functional units and data stores (see Figure 5). A RETSINA agent uses four data stores: The *objective database* is a dynamic data store. It stores the objectives of the agent of which it is a component. New objectives are inserted by the *communicator* (from outside sources) and by the planner (from inside sources) as planning may create new objectives. The *task database* is a dynamic data store. It stores tasks which were reduced to the lower level, i.e., to actions. These tasks may still be not ready for execution and wait in the task DB until the required conditions for their execution are set true. When this happens, the actions are considered enabled, and are scheduled for execution by the *scheduler*. The *task schema library* is a static data store that holds tasks schemas. These are used by the planner for task instantiation. The *task reduction library* is a static data store that holds reductions of tasks. These are used by the planner for task decomposition.

The functional modules, each of them an autonomous thread of control, use the data stores as follows:

The *communicator* receives and sends messages, parses incoming messages and creates objectives which are inserted to the objective DB. The *planner* performs instantiation and reduction of tasks. It takes the objectives of its agent, decomposes them to lower level tasks, and the tasks which are executable (referred to as actions) are passed forward, to be scheduled for

execution. The scheduling of enabled tasks is performed by the *scheduler*. It takes enabled actions from the task DB and puts them, scheduled, in the *schedule*. Activation of actions in the schedule is performed by the *execution monitor* thread. For each action on the schedule it creates a separate thread of control, and it monitors the activity of each of these working threads. Action threads may propagate outcomes to other modules.

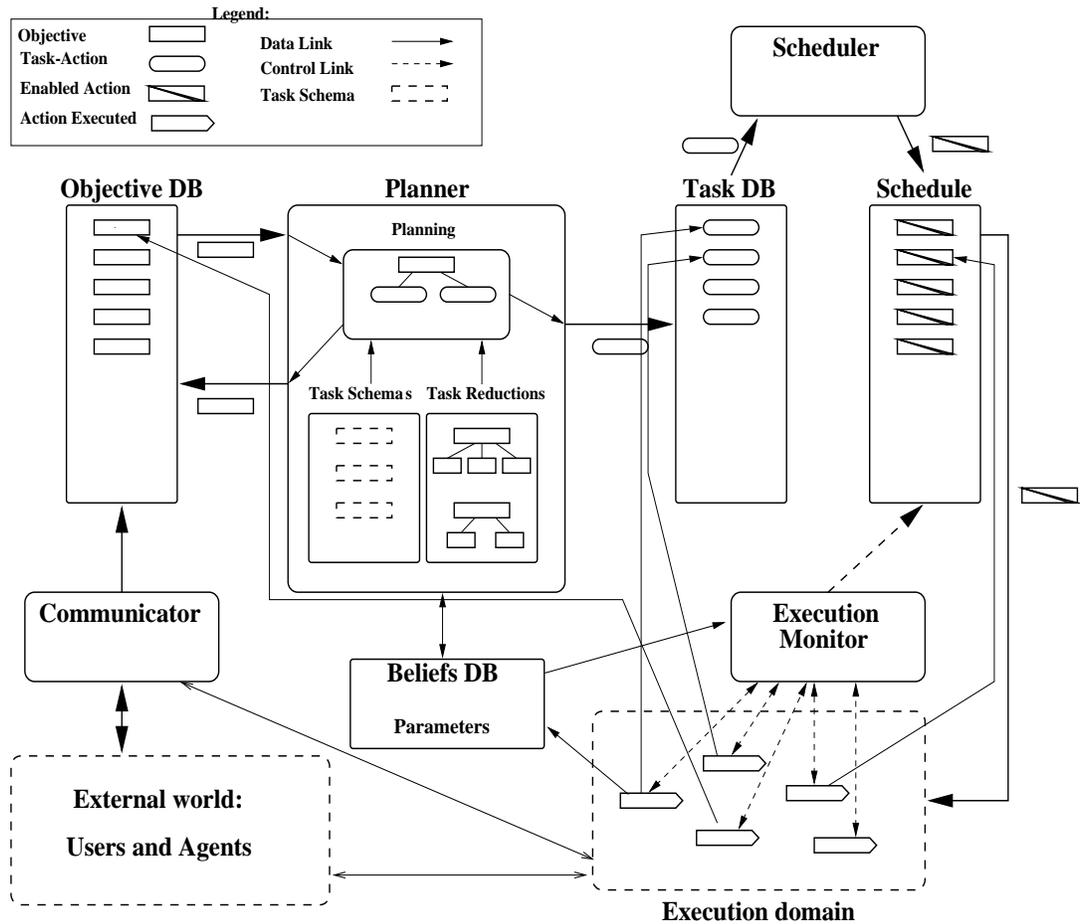


Figure 5: The RETSINA agent architecture.

The modularity of the RETSINA agent architecture and having no direct interfaces between its functional modules result in code re-usability (e.g., the RETSINA communicator is used for multiple agent types and for non-agent

applications that converse with agents). In addition, functional components can be easily replaced in a plug-in fashion.

4 The solution design

Given its properties, we found the RETSINA infrastructure appropriate to solve the problem considered. By developing using an agent architecture, we gain the following advantages:

- The RETSINA architecture can be used to wrap legacy software systems, by equipping them with a Communicator module. Thus the resulting system remains backwardly compatible with the older system, without tying future software development to an obsolete model. For instance, currently the Warner Robins AFB engineers are experimenting with entering some of the data into Access database format, as a temporary measure while waiting for the ITL-ALC system to become available. With our proposed design, separate Infoagents can easily be designed to accommodate both data sources; since the maintenance personnel only interact with Interface agents, they are shielded from internal data discontinuities.
- The information required by the maintenance engineers is likely to be distributed among several computers, possibly in different geographic locations. The RETSINA architecture provides built-in networking support useful for developing distributed systems, in the form of the Communicator, which handles low level socket operations, and the Agent Name Server/Matchmaker, which allows service requesters to locate service providers. Although currently we are focusing on handling the repair operations described in Form 202A, which are performed locally in Warner Robins AFB, additional agents can be added to the system to access collections of Form 00-107, immediate repair requests which can be filed from multiple locations. These agents would be located on computers at the local Air Force base performing the repair and would communicate to agents at the central F-15 repair location, Warner-Robins AFB.⁴

⁴Communications in the next release of RETSINA will be based on secure sockets to allow the transmission of sensitive military information over the Internet.

- The Warner-Robins Air Force Base is in a transitional phase of reorganizing their data and also training personnel. We hope to quickly prototype a group of agents to address the current situation and slowly add to the “agent population” as new information sources become available electronically. Since the interface agent is decoupled from the Infoagents, it is easily possible to replace older Infoagents without disturbing the users. Training the personnel to use the system is an important part of making it operational; by changing the interface as little as possible during development, we hope to facilitate the learning process.

Below, we describe our current implementation of an agent system designed to aid the aircraft maintenance engineers in their repair decisions, along with plans for future improvements.

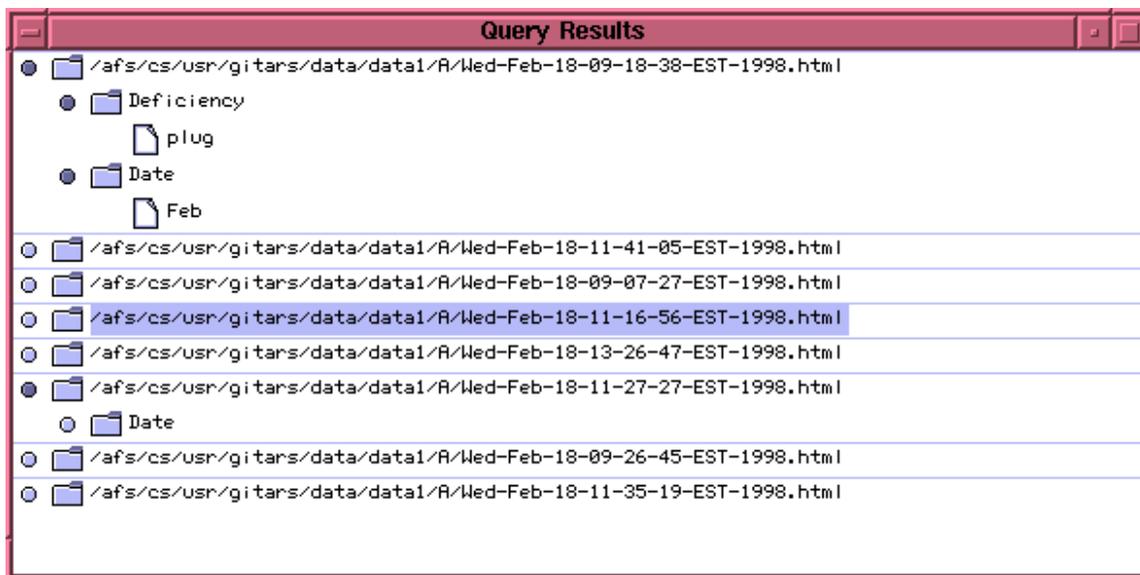


Figure 6: The result of the query, as presented by the form agent, is a list of matched forms. The matching fields and keywords can be viewed by clicking on a specific form entry.

4.1 Information sources

There are several sources of information relevant to the inspection and maintenance processes. Note that at this stage of system development some effort was already put into converting information into electronic formats, although most of the potential information remains unconverted. The current implementation of the system uses the following sources of information:

Form 202A: This form is used by the aircraft mechanic to request advice on specialized repairs not covered by the manuals (see below). Currently most of these forms are available only in paper form, although the ones produced by the ITL-ALC system are in HTML format. The collection of historical 202A forms is the only way that the maintenance people can track damage and repairs made to F-15 aircraft.

1F-15E-3-5: This is the standard manual used by the aircraft mechanics to perform frequently occurring procedures: “Typical Repairs, Repairs of Special Structure, and Sealing.” We were provided with a PDF version of this one manual, which is part of an entire series of F-15 manuals used by the engineers and mechanics.

1F-15E-4-1 : This document is the illustrated parts breakdown for the F-15E airframe. Using this document (available in PDF form), the mapping from part number to part name can be determined. The illustrated parts breakdown (IPB) includes complete schematics for the F-15E airframe.

Future agents could exploit the following information sources:

Form 202B: After the mechanic submits a special request with Form 202 A, the engineer responds to the request using Form 202B to describe the procedure that the mechanic must follow. The collection of past Form 202Bs constitute a valuable corpus of information, but unfortunately we only have a few electronic versions of this form converted into HTML.

Form 00-107: The Air Force has a slightly different form for emergency repairs performed at locations other than Warner-Robins AFB. This form is identical to the Form 202A in its function.

additional F15 manuals: The Air Force has an entire series of F15 manuals which are now only available in hardcopy. As these manuals become available electronically they can be easily incorporated into our system.

general aerospace information: Occasionally the engineers refer to standard aerospace handbooks. For our system, we have focused exclusively on information unique to the F-15 maintenance problem, since it seems likely that in the future general engineering textbooks could be accessed by interoperating with agents from a digital library collection (UMDL).

graphics and audio information: Mechanics using the ITL-ALC wearable computer system can append pictures of the aircraft discrepancy and a spoken voice clip describing the problem to the standard Form 202A. Although we believe that this information will be very useful to the maintenance engineers, we lacked sufficient information samples to develop a full agent. Our interface agent could easily be extended to display graphics or audio clips included with a 202A form.

human advice: Other maintenance personnel can provide useful advice for the engineers. Our agent system currently provides no formal support for conferencing, since almost all the F-15 personnel reside in the same geographic location.

4.2 Agent functionality

The history agent and the manuals agent are both, to a large extent, information agents. As such, they can receive requests for information and reply with the required information. Both of them receive the same type of requests—a single shot query. The content of the query includes one or more part-numbers and one or more fault descriptors for each fault in the request.

The history agent parses each historical form to which it has access, usually limited to local archives, searching for complete or partial matches. (We leave some flexibility to allow a search using an ontology to increase hits on less obvious cases). The forms which match the designated problem are inserted into the content of the reply message. In contrast, the manuals agent does not need to perform parsing. It only has to search, using indexes, through the manuals. Matches are inserted into the content of the reply message.

<input type="checkbox"/> To	<input type="text"/>
<input type="checkbox"/> From	<input type="text"/>
<input checked="" type="checkbox"/> Date	Feb
<input type="checkbox"/> ControlNo	<input type="text"/>
<input type="checkbox"/> Noun	<input type="text"/>
<input type="checkbox"/> PartNo	<input type="text"/>
<input type="checkbox"/> NationalStockNo	<input type="text"/>
<input type="checkbox"/> SerialTailNo	<input type="text"/>
<input type="checkbox"/> Location	<input type="text"/>
<input type="checkbox"/> ToDwgNo	<input type="text"/>
<input type="checkbox"/> WorkStoppage	<input type="text"/>
<input type="checkbox"/> Initiator	<input type="text"/>
<input type="checkbox"/> PhoneNo	<input type="text"/>
<input type="checkbox"/> NotifiedTCM	<input type="text"/>
<input type="checkbox"/> OrganicallyCaused	<input type="text"/>
<input type="checkbox"/> MaintenancePlanner	<input type="text"/>
<input type="checkbox"/> FileName	<input type="text"/>
<input type="checkbox"/> Html	<input type="text"/>
<input checked="" type="checkbox"/> Deficiency	plug
<input type="checkbox"/> Recommendation	<input type="text"/>
<input type="button" value="Submit search"/>	

Figure 7: A table for manual submission of requests for keyword searching.

The form agent, upon request from the engineer (a button click), performs several actions. It first retrieves a list of keywords (of fault descriptors) from local file. Optionally, the engineer may also request for a search manually. This is performed using a field table, where the engineer can insert the requested keywords (a snapshot of this search request window is shown in Figure 7). Then, it parses the current form to find part-numbers and keywords. These are inserted into the content field of an outgoing request. The form agent then first sends (this functionality may be button controlled by user) a request to a manuals agent. If the results are null (or the user requests additional information) the form agent sends a request to a local history agent. If the results are not satisfying, it sends requests to remote history agents (in other service centers). Any non-empty result that arrives during this process is presented on a results window. A snapshot of this window is shown in Figure 6. Initially, the results of the query is a list of the addresses of the forms which were found relevant (according to keyword matches), ordered by relevancy. The user may click on items on this list to



Figure 8: The hand-held computer used by a mechanic.

find what fields and what keywords resulted in a match. In addition, the user may request to display a selected form. In this event, the form is displayed with the relevant keywords highlighted. Text and graphics should be arranged in a reasonable way (this is still in a development stage). After the automated consultation is finished, the form agent adds the current 202 forms to the historical database of these forms.

4.3 Data processing and flow

Using these three types of agents, we have built a multi-agent system in which each specialized type of agent may have several instances in the system. Below, we describe the processing and flow of information in the computerized system, as seen in Figure 9. The process begins with a mechanic (who is sometimes required to be certified as an inspector), inspecting the plane. The mechanic uses a wearable computer (a Fujitsu 1200 as in Figure 8) with a touch-screen, microphone and a digital camera. When a discrepancy is found, the mechanic fills in an electronic 202a form (Figure 3), and when necessary and practicable, adds voice notes and digital photographs. The

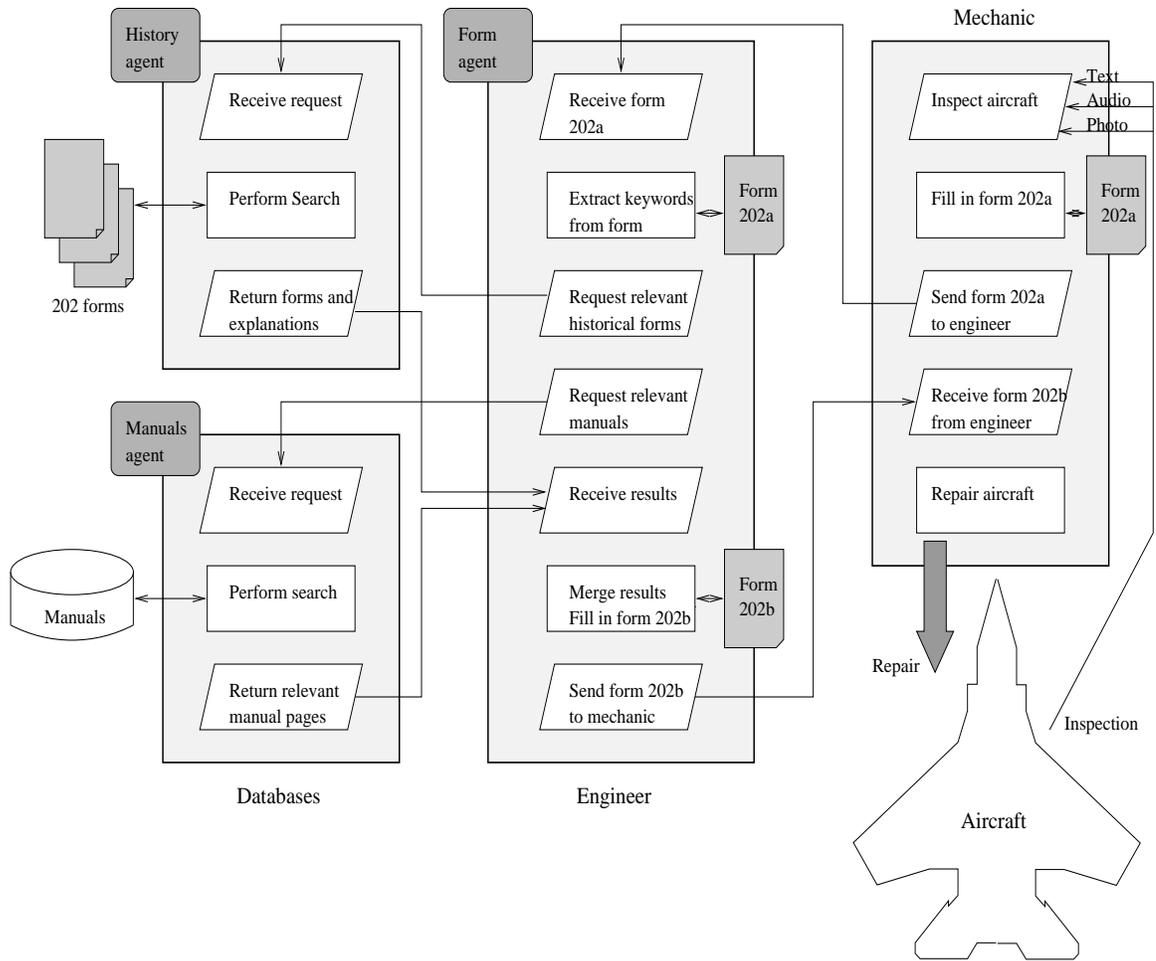


Figure 9: Data processing and flow in the computerized system.

202a form with the attachments is sent to an engineer. At this point, the mechanic waits for a reply from the engineer.

The engineer, with the support of a form agent on his/her workstation, extracts keywords from the 202a form. Using these keywords, the form agent automatically requests for relevant historical forms from history agents and for relevant manual pages from a manuals agent. These requests may also be activated, controlled and edited by the user (engineer). At this point, the form agent waits for the requested information to arrive, in reply to its requests.

History and manuals agents are located on central computer networks of service centers, on which the archival information they need to access is located as well. Upon receiving a request for information, history agents perform a search on the historical 202 forms archive, and conduct a relevancy analysis. They reply with a list of relevant forms, the reason for their selection and the level of relevancy. A manuals agent performs a simple search in an indexed manuals database and replies with the results of this search.

Upon receiving replies from history and manuals agents, the form agent merges results and displays them to the engineer. Using this information the engineer can decide upon the appropriate repair procedure, fill in an electronic 202b form, attach to it graphical description grabbed from manuals and historical forms (or draw new ones, if necessary), and send it to the mechanics wearable computer.

The information flow and processing end when the mechanic receives and opens the 202b form on the wearable computer. The details in the 202b form and the approval of a repair procedure allow the mechanic to execute the actual repair.

4.4 Multi-agent organization

The description of information flow and processing provides only a partial view of the system developed. At least as important is the way in which the agents are organized to provide the required processing and flow. Therefore, it is necessary to provide a description of the multi-agent organization. Such presentation also provides an insight into the distribution, reuse and scalability of the system. A graphical illustration of the multi-agent organization is presented in Figure 10. As depicted there, multiple mechanics use each a wearable computer in the inspection process to compose a 202a form. These

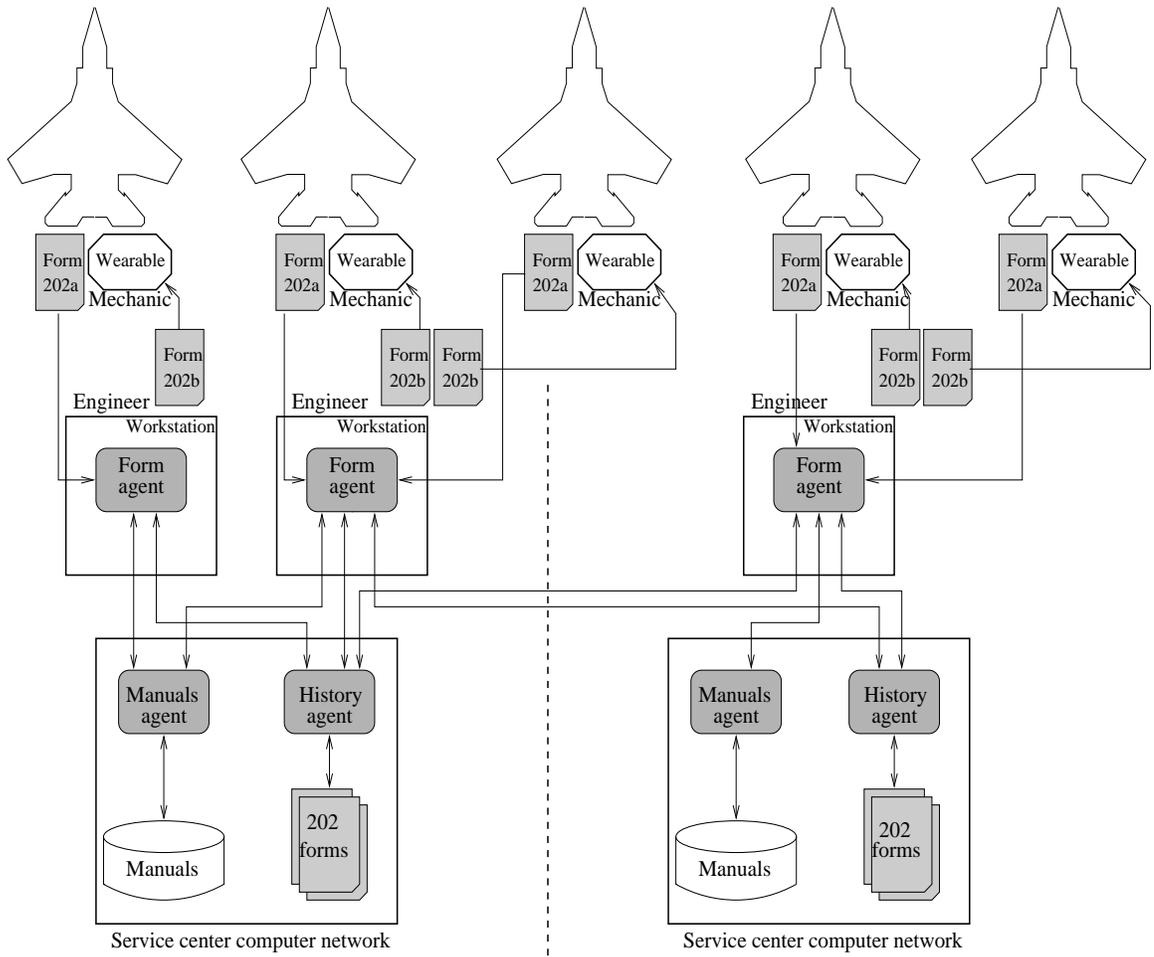


Figure 10: The multi-agent system organization.

forms are sent to form agents. There may be multiple form agents and each form agent may handle several 202a forms. Each form agent may request information relevant to the forms it handles from multiple history agents. This is necessary since historical archives of 202 forms may be distributed over multiple service centers. Manual agents and history agent may be requested for information by multiple form agents. In a closed system agents know about one another in advance. Since agents may go down or new agents appear, there is need for open system organization. The RETSINA multi-agent infrastructure allows for such an open system organization through the presence and interaction with middle agents [1].

5 Discussion

The application of a RETSINA multi-agent infrastructure to the aircraft maintenance problem and its implementation provide the following advantages.

- Automatic location and retrieval of relevant information. This information is necessary to provide advice regarding to the potential risk discrepancies pose and support decisions upon the appropriate repair methods.
- Historical repair data is utilized, thus information about discrepancy re-occurrence can be re-used. In addition, long term learning of typical problems of the aircraft can be performed.
- The access to manuals via an electronic, indexed database increases the efficiency, accuracy and completeness of information retrieval from the indexes, thus reduces the probability of mistaken decision for repairs.
- As a result of electronic retrieval, fusion and transmission of information, the average repair time is reduced.

In conclusion, the increased speed and quality of information gathering, filtering and merging for repair decision results in timely and qualitative aircraft maintenance. In addition, since the underlying architecture of the system is an open multi-agent system, it is possible to re-use it for similar domains (and there are myriad distributed repair domains which can benefit from such

a system). The openness of the system facilitates the automatic addition of new agents; as new resources⁵ become available, agents can be created to exploit them, without modifying the existing system. This openness also allows for dynamic appearance and disappearance of agents and information sources without hampering the overall system performance, since RETSINA agents know how to find alternative agents when their initial choice is not available. We believe that multi-agent systems, and in particular the RETSINA infrastructure, provide a good solution to this type of decision support problem, as demonstrated by our prototype of agent-aided aircraft maintenance system.

6 Acknowledgement

This work could not have been performed without the cooperation and support of Dan Siewiorek, Dick Martin and Jane Siegel from the Wearable Computing project at Carnegie Mellon university and the Warner Robins Air Force Base, Georgia. Funding for this work has been provided by the ONR grant #N00014-96-1-1222.

References

- [1] K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceeding of IJCAI-97*, pages 578–583, Nagoya, Japan, 1997.
- [2] Michael N. Huhns and Munindar P. Singh, editors. *Readings in Agents*. Morgan Kaufmann, San Francisco, 1998.
- [3] N. Jennings and M. Wooldridge (editors). *Agent technology*. Springer, 1998.
- [4] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous agents and multi-agent systems*, 1(1):7–38, 1998.
- [5] H. Nwana and N. Azarmi (editors). *Lecture notes in artificial intelligence vol. 1198: Software agents and soft computing*. Springer, 1997.

⁵Examples of such resources include: information sources; methods for processing information; user preference learning.

- [6] K. Sycara, K. Decker, A. Pannu, and M. Williamson. Designing behaviors for information agents. In *Proceeding of Agents-97*, pages 404–412, Los Angeles, 1997.
- [7] K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng. Distributed intelligent agents. *IEEE Expert – Intelligent Systems and Their Applications*, 11(6):36–45, 1996.
- [8] K. Sycara and D. Zeng. Coordination of multiple intelligent software agents. *International Journal of Intelligent and Cooperative Information Systems*, 5(2 & 3):181–211, 1996.