Identifying Autonomous Agents for Capitalizing Knowledge in R&D environments

Jean-Paul. A. BARTHÈS†, Hilton de AZEVEDO††

† CNRS UMR 6599 HEUDIASYC
Université de Technologie de Compiègne, France
barthes@utc.fr

†† CEFET
Curitiba, Parana, Brasil

CEUR-WS.org/Vol-14/ecai98om-barthes.pdf

Abstract:

Whilst we have a number of techniques for building multiagent systems, such techniques assume that the number and role of each agent in the system is known a priori. Unfortunately it is rarely the case in practice, in particular when trying to build multiagent knowledge management systems in R&D environments exhibiting a high rate of turnover. Thus, we need specific analysis and design guidelines. The paper presents a new method, SAAS, for identifying and selecting a collectively agreed upon set of autonomous agents necessary to implement a system for capitalizing collective knowledge. The method borrows concepts from another method proposed by Grundstein for identifying strategic knowledge in production processes. SAAS was used to determine the number of agents necessary for capitalizing the knowledge of a research group in the MEMOLAB project at UTC.

In 1994 we were developing the OSACA[1] environment as a set of tools for facilitating the construction of open systems of cognitive agents [Scalabrin et al 96]. The OSACA approach was to be tested in the design of a system for capitalizing the knowledge of a research laboratory, in the MEMOLAB project. The main objective of MEMOLAB was to record part of the knowledge used in a research laboratory for helping the researchers to do a better work, but also for helping the newcomers to be integrated faster, while avoiding to bother the old timers with too many questions. One of the main problems was thus to accommodate a population varying over time in a very dynamic environment, in particular as found in laboratories with a high turnover of Ph.D. students and a small number of permanent researchers. The main approach chosen in the MEMOLAB project was to delegate the management of parts of the knowledge to specific automatic agents, and to develop a system of such agents interacting with human people (through additional proxy agents). A priori and considering our previous experience in building multiagent systems, we thought that it would be relatively straightforward to set up agents as knowledge providers.

Our approach to the global problem of knowledge management followed the guidelines developed at the IIIA Institute, and summarized on Fig. 1 which displays one way of looking at the process of knowledge management. The process consists of four steps grouped around a central part referring to strategic or crucial knowledge: (i) identifying and localizing the relevant knowledge; (ii) formalizing it; (iii) distributing it (i.e., providing access to it); (iv) maintaining it. Transversal to this approach was the use of cognitive agents as a supporting technology.

---

[1] Open System for Asynchronous Cognitive Agents

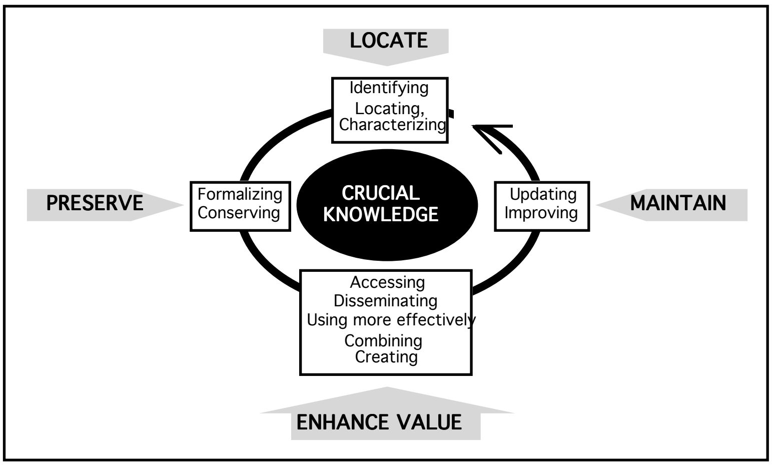


Fig 1. A view of the knowledge management process
(from [Grundstein & Barthès 96])

We soon discovered that if we wanted to improve the overall functioning of a research group including researchers, students, technical and clerical staff, it was not sufficient to focus on expert knowledge. Rather, it was found more fruitful to think in terms of *services* rather than of knowledge. Hence a shift towards designing agents providing services. A first try was attempted after informally asking several persons what kind of services they would like to have, but then it was rapidly found that some method was needed to provide the guidelines for analysis and design of the system we had in mind.

We turned towards the traditional methods found in knowledge engineering and used in the industrial groups or software houses we knew, (KOD [Vogel 88], MKSM [Ermine et al 96], REX [Malvache & Prieur 93]), methods in software engineering (for developing object systems, like Object Design [Coad & Yourdon 90, 91], and others), or approaches developed for enterprise modeling [Fox 88; Wiig 93, 94]. We did not try all possible methods proposed in the literature. We found that the methods we tried were not easy to use in our case. Indeed, briefly stated, such methods usually assume that services are well defined, and mainly optimize the resulting structure of concepts or classes. In our case, we did not know what services were required, nor how they would be used, and need first to determine such concept and to obtain the agreement from all the human actors. We then turned to methods for locating knowledge within the enterprise, and found a method proposed by Grundstein for identifying crucial knowledge [Grundstein 96].

Grundstein's method provided a good basis for designing a new method better suited to our purposes (i.e., determining what services we could group inside which agent) which we named SAAS, and which we present in this paper. It must clearly be understood that SAAS is not intended to model the business process, e.g., as discussed in [Decker et al 1997], nor it is a method for knowledge acquisition. Its purpose is simply to partition useful knowledge into various domains, with as little overlapping as possible, to assess the usefulness of each domain so that we can assign a cognitive agent to each selected domain of interest. In other words, we are interested in determining how many agents we need to support the knowledge system, and which operations each agent will perform, which determines its interface protocole. Once we have the answers, the agent designer is free to structure and to implement knowledge within each agent in any way she feels adequate, probably using different methods.

Finally, SAAS is a bottom up approach which builds upon the users' work habits. On Fig 1, it can be positioned in the LOCATE and PRESERVE steps, with extensions to the design of interaction interfaces in the ENHANCE VALUE step (which will not be covered in this paper). SAAS has been developed for technical R&D environments subject to fast turnover. It is not clear whether it can be applied to other situations.

The paper is organized as follows: after reviewing Grundstein's method in Section 1, the paper introduces the SAAS method in Section 2, and illustrates its use in Section 3.

1. GRUNDSTEIN'S METHOD FOR IDENTIFYING CRUCIAL KNOWLEDGE

As many other methods used in industrial contexts, Grundstein's method assumes that an enterprise is a set of *production processes*, composed of *sequences*, in turn composed of *activities*.

An *activity* (Fig. 2) is a set of elementary effective tasks, homogeneous when considering their cost, behavior and performance. An activity is accomplished by one or more *actors*, it has an *objective* and produces something. An activity uses financial and technical *resources*, uses and produces *expertise*, is constrained by external factors (price, delay, quality, specifications...) and internal factors (activity's operability range). Finally, activities may present problems (standards, procedures, reasoning attached to the activity that results in differences between the expected and obtained results).

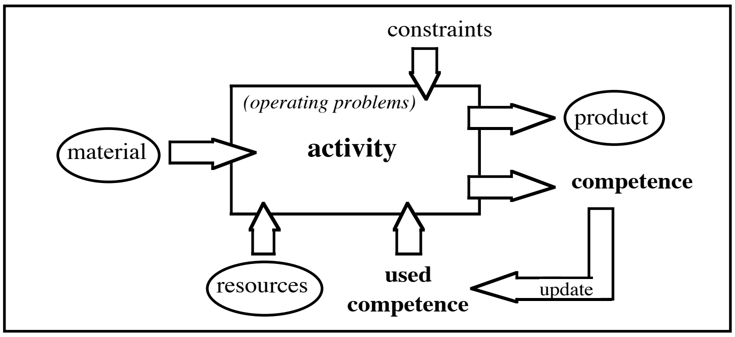


Fig. 2. Grundstein's model of activity

Some activities can be *critical* if they can disturb the process to which they belong, which could in turn disturb the production cycle.

The objective of Grundstein's method is to detect critical activities and to identify their associated *strategic knowledge* (necessary to achieve them). Strategic knowledge is identified by analyzing the actors and resources associated with a critical activity. Criteria like nature, access, cost, or duration of strategic knowledge, allow then to evaluate the interest of starting a capitalization process.

Grundstein's method includes four steps:

Step 1. Interviewing the actors of the process to be analyzed. Each interview yields a partial view of the process.

Step 2. Synthesizing a global viewpoint. Once all interviews are done, all actors are gathered to synthesize a global viewpoint. During this step conflicts must be solved by the actors.

Step 3. Analyzing the global process. For each activity in the process, one must list all problems and constraints that can impact each activity. Subsets of activities are then defined, after identifying:

- production cycles, composed of an homogeneous set of activities, working towards a common objective,
- critical activities, as defined previously,

- *crucial problems*, which can modify operating conditions of critical activities,
- *crucial knowledge*, which must be used to solve critical problems.

Step 4. Studying possible solutions. This step consists in comparing the various possibilities offered by different products when handling crucial knowledge. The cost of loosing some crucial knowledge and the cost of the production costs is compared with the cost of capitalizing the corresponding knowledge. One can thus assess the usefulness of the various commercial products.

The strong features of Grundstein's method are:

- representing a process exclusively by its actors' activities
- taking into account only activities actually performed by the actors
- making explicit the links among activities, actors and resources for a process.

We found that the overall approach centered on the concept of activity (which can be found in many other methods in particular for enterprise modeling), could be used for developing a technique suitable for analyzing and designing multiagent systems. Hence, the SAAS method (Service Analysis for Agent Systems).

2. THE SAAS METHOD

The SAAS method was developed for determining the services one has to provide to a population of different users. It was designed keeping in mind two sets of constraints: technical and human constraints.

Technical constraints

- crucial services must be recognized by all users.
- knowledge must be spread among agents with an optimal granularity. Indeed, if the knowledge is too complex, then the agent will be complex and difficult to implement and to manage, if the knowledge is too thin, then the communications will be complex and expensive.
- since agents will provide services, it is important that dependency among services become visible, which can help in case of failure.

Human constraints

- interviews must be short.
- viewpoints must be easily represented for a better identification of critical activities.
- the representation must accommodate several levels of details
- the used formalism must be simple and intuitive.
- the formalism must not introduce biases.

Considering the constraints we defined now the SAAS method.

The SAAS method

SAAS is defined as a series of eight steps:

Step 1. Gathering viewpoints
Step 2. Classifying activities and resources
Step 3. Validation by the group

Step 4. Description of services
Step 5. Writing scenarios
Step 6. Building mock up
Step 7. Identifying competencies
Step 8. Synthesizing competencies

Step 1 and 2 consist of gathering information and organizing it in tabular format (cards) or graphs; Step 3 and 4 allow to determine the needed services; Step 5 to Step 8 are required for validating the findings. We now detail the approach, focusing on the first four steps, since the remaining ones are more traditional in particular for designing man/machine interfaces in software engineering.

Step 1. Gathering viewpoints

During this step actors are interviewed so that they can give their viewpoint concerning processes. Three documents are produced during this phase: (i) a card describing the actor's activities, (ii) a card describing the used resources, and (iii) a graph describing how the actor connects the activities. In order to be able to compare the results from various interviews the representation is built using a limited number of primitive concepts. We use four concepts (Fig. 3): *activity, resource, actor*, and *product* (imported from Grundstein's approach), and five relationships: *using, has-actor, producing, followed-by, has-activity*.

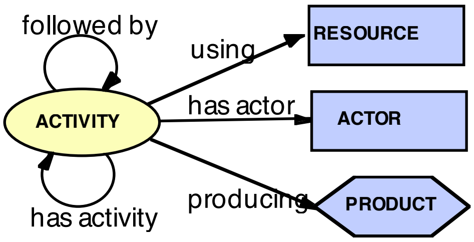


Fig. 3. Actigraph

The Graph, that we call an **Actigraph**, is an oriented graph in which nodes represent activities, and arcs represent the dependencies. It is built by the interviewer and is used to let the interviewed actor immediately visualize the interpretation of the interviewer in an intuitive fashion.

On the actigraph each activity can be detailed as shown on Fig. 4 until one reaches primary activities.

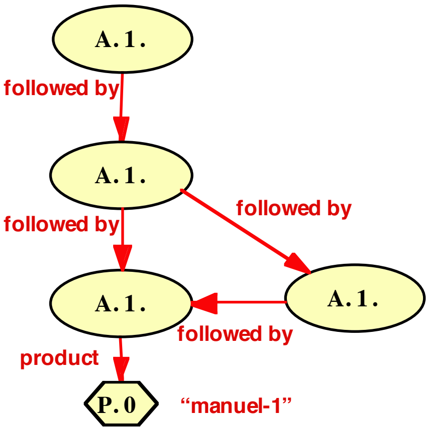


Fig. 4. Detailing a complex activity

Products are then identified for primary activities (Fig. 5).

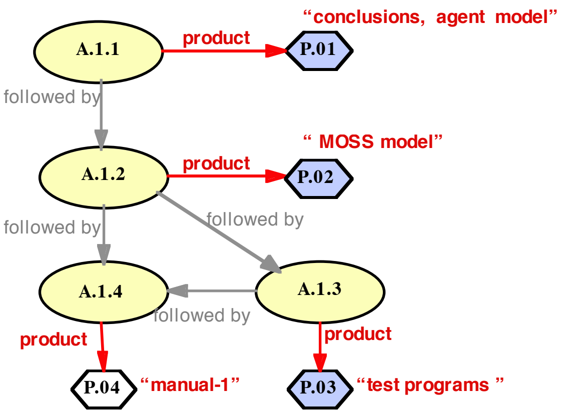


Fig. 5. Product attached to primary activities

Resources can then be displayed for each primary activity (Fig. 6).

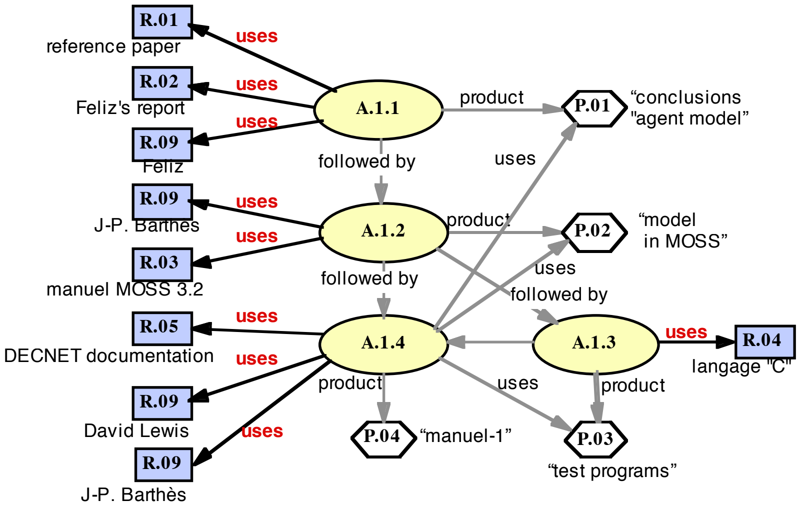


Fig. 6. Resources attached to primary activities

One then defines an Activity Card containing information about a particular activity, organized in tabular form. The information contains the objectives of the activity, the used resources, and remarks.

ACT- *N*	*Activity Name*	
Description		
resources	**description**	**type/orig.**
RES- *1* … RES- *N*	*resource description* … *resource description*	= i … = i, e

Table 1 : Description of an activity

The Resource Card plays the same role for a resource and has a similar format.

The guidelines to construct the activity and resource cards are given in Table 2.

Activity	description	Resource		description
Mnemonic:	used to identify the activity	Name:		used to localize the resource
Activity name:	the complete name	Type:		people, material, software, procedure
Objective:	why it was done	Location:		where it can be found
Keywords:	used to index	Contact:	*(if exists)*	
Dependency:	activity that preceded		name	who supplied the resource
Upper activity:	activity that owns this activity, if it exists		location	division, laboratory, room, etc.
Supervisor:	name		telephone	
	telephone number		interaction	informal, appointment, team work
Production:	activity product description		cooperate. level	diffic., poor, medium, good, excel.
Short comings:	text describing the activity bad or good points		remarks	was it: positive, null, negative to the activity?

Table 2. Guideline to detail an activity

During this step a knowledge engineer interacts with the user, starting with very general questions and progressively refining the set of activities and their dependencies. The approach is summarized in Table 3.

Step 2. Classifying activities and resources

Contrary to Grundstein's method where all actors (specialists) are gathered to reach a consensus about a global view of the processes, we advise that this step be done by the knowledge engineer. The main objective of this step is to determine according to various users' profiles the crucial knowledge for a particular group. Criteria can be: frequency of utilization of an activity or a resource, quantity of resource used for an activity, quantity of activity used by a resource, quality of the product, relevance, robustness, etc. The first three criteria can be deduced from the interviews, the other ones would require additional questions, but also would require to increase the number of primitive concepts and relationships used to build the actigraphs. Thus, we limited ourselves to the first three criteria.

In Step 2, for each identified profile the knowledge engineer produces a report which is then sent to the members of the group. The members can validate by criticizing the propositions.

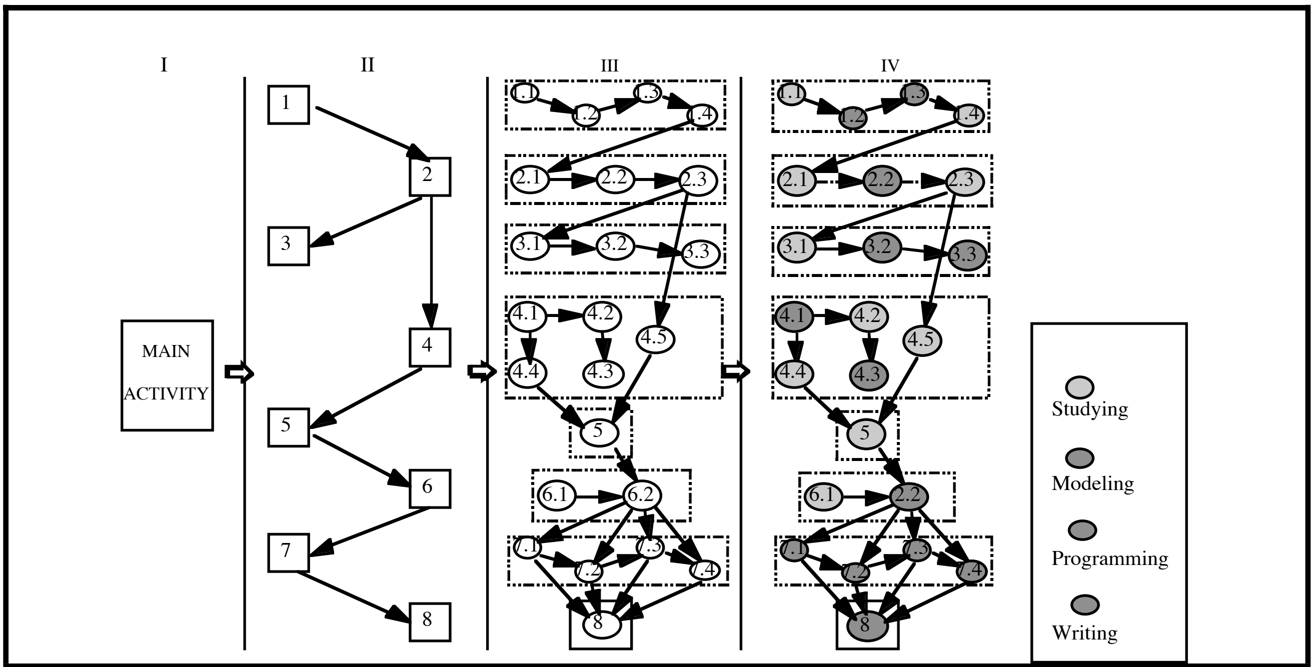
For a given user profile the report contains all actigraphs, all activity cards, a table synthesizing the ensemble of activities and indicating the amount of used resources (an example is given in Section 3), the activity profile of the given group, a table synthesizing the use of resources, a resource profile, and some conclusions regarding possible services.

Step 3. Validation by the group

The members of a group corresponding to a professional profile select some services, which are recorded in a Table Identifying Services.

Step 4. Description of services

Services selected by different groups are compared. If conflicts appear the groups have to get together. The knowledge engineer then produces the final Table of Services which contains a detailed description of each proposed service. This will be used later for positioning competencies within services. The Table of Services must be approved by all the groups.



process	production
1 - user declares her/his main activity;	graph I
2 - user details its main activity;	interview text, graph II
3 - the graph and the answers obtained are analyzed;	graph III
4 - graph II is presented and user is asked about more details;	interview text, graph II
5 - the user checks graph III, if agreement then step 6, else go to step 2, 3 or 4;	graph III
6 - classing the activities by domains. Graph III is colored;	graph IV

Table 3. The top-down approach used to identify a user's activities.

Step 5. Writing scenarios

Then, for each service in the Table of services, one has to write scenarios showing how the service will be used by the a particular user. The goal of the scenarios is to describe carefully and realistically the work environment. This must be done by the person who will be responsible for designing the agent supporting the service. Then, all scenarios are presented to the group of users corresponding to a particular profile. The group edits the scenarios, criticizing, adding, or removing parts.

Thus, steps 1 to 4 allow to determine which service will be associated with which agent, and, hence, determines the total number of agents. Steps 5 to 8 in contrast help to organize activities within each particular agent.

Step 6. Building mock up

Each part of a scenario leads to defining computer screens (showing the interactions with a particular service). A simulation is built to animate scenarios. The result is presented to selected members of a group for feedback.

Step 7. Identifying competencies

Each scenario contains references to competencies (actions needed in services) linked to the various services of the Table of Services. Competencies are identified, named, described, input and output parameters are recorded. Examples of use are noted. The needs of each competency are made explicit, which will be used to describe a competency provided by a given agent.

Step 8. Synthesizing competencies

In this phase all competencies referred to in the various scenarios supporting a given service are put together. Some competencies will be fused. The goal is to reduce the coding effort. At this stage external needs which cannot be found in other agents will have to be added locally. Thus the amount of programming effort for each agent can be estimated.

3. SAAS FOR MEMOLAB

The MEMOLAB project was conducted in the framework of our laboratory. In the laboratory, as we already mentioned, a large amount of knowledge is developed by Ph.D. students; not enough is written. After graduating, the students leave the group and a lot of knowledge is lost due to the small size of the permanent members. This may increase the time needed to integrate newcomers and consequently affect the laboratory overall efficiency. Our staff is reduced: 6 researchers and 18 Ph.D. students, having access to 4 part-time secretaries and 6 part-time engineers-technicians who work with all the other research teams in the department. Currently, the Computer Science Department has 37 full time researchers and 67 Ph.D. students. When a newcomer joins the group, it is important for her to create quickly an individual representation of the group's working domain and of the resources available within the laboratory. Thus, representing the collective knowledge of our research group and improving its accessibility to newcomers will reduce their integration time, as well as improve the overall reuse of knowledge.

Knowledge is stored on various media: videos, pictures, papers, articles, theses, human memory... The integration of newcomers is done by defining pointers for accessing such knowledge. Sometimes, one has to locate the physical documents (e.g., reports, articles, videos, books, manuals, software, seminars), sometimes, the access is more informal (e.g., chats in the coffee-room, sharing tips and tricks with newcomers). Transferring knowledge may use indirect non explicit channels (e.g., observing how someone operates a software package or equipment). Such examples illustrate the diversity of the forms of communication used by the group to continually reuse, update, create, and manage its knowledge. The goal of MEMOLAB was thus to develop a system of cognitive agents for managing the knowledge of the research group.

In an initial approach to identify MEMOLAB agents, we used the results of an inquiry involving 20 people who work in different research teams at UTC. They were requested to rank by order of importance a small list of services (that we thought would be useful for their work). They were also invited to extend the list of services if they wished. This inquiry resulted in a first set of agents that seemed to be representative of the group's needs. Nevertheless, we were not sure that the choices for the agents were the best. Thus, we decided to check our first choices by comparing it with those obtained as a result of applying the SAAS method.

The result of applying the SAAS method yielded a number of documents for each group of users (researchers, Ph.D. students, clerks, technical staff). We give some examples of the documents resulting from the work with the Ph.D. students. Interested readers can refer to [Azevedo & Barthès 96].

Activity cards (Step 1)

Table 4 is an example of an activity/resource table. The nature of a resource can be diverse and is indicated by a symbol: **4** physical (documents), **:** (equipment), = software (commercial, public, internal software), or **J**□human (experts). The origin for the resource is indicated by a letter **i** (internal to the laboratory) **e** (external, i.e., coming from outside the laboratory). Resources with few users are indicated by a white background, resources shared by several users are indicated by a gray shaded background.

Ph.D. Students Activities		ACT- 1 **Studying**	
The objective is to bring the technical knowledge of a Ph.D. student to be the state of the art. It is normally done by: reading articles, theses, reports, books; attending classes, workshops, congresses; discussing with specialists; visiting industries…			
resource		**resource description**	**nat./orig.**
RES- 1	programs	pieces of code written in a programming language	= **i**
RES- 2	bibliographic references	papers from journals, magazines, proceedings, readings; reports; manuals; books, manuscripts, Ph.D. thesis, programs	4 **i, e**
RES- 3	software package	commercial and public software package developed outside the laboratory e.g.: ftp; NETSCAPE; EMACS; WORD; PowerPoint, Hugo	= **i, e**
RES- 4	expert advise	get from someone that has more experience in the context of work	J **i, e**
RES- 5	team work	working with someone else	J
RES- 6	equipment	SUN workstation, Macintosh, NEXT, laser printer, fax, telephone, photocopy machine, minitel, BUTC terminal...	: **i**

Table 4. Studying activity

Synthesis tables (Step 2)

Table 5 combines the synthesis of activities, the activity profile, the synthesis of resources, and the resource profile. Such tables are produced by gathering all information acquired during Step 1. Together with the activities and resources, a quantitative information is produced and is also displayed as histograms. The bottom left histogram show the percentage of activities which use a particular resource. The bottom right histogram shows the percentage of resources used by aparticular activity.

Resources	(%)	**Users**	**Activity**	**Resources (%)**
RES-01 programs	**78**	= **i**	ACT-01 studying	**50**
RES-02 bibliographic references	**89**	4 **i, e**	ACT-02 developing	**80**
RES-03 software packages	**67**	= **i, e**	ACT-03 programming	**80**
RES-04 expert advise	**44**	J **i, e**	ACT-04 writing	**60**
RES-05 team work	**55**	J	ACT-05 preparing exercise classes	**70**
RES-06 equipment	**89**	: **i**	ACT-06 preparing laboratory classes	**70**
RES-07 programming environm	**44**	= **i, e**	ACT-07 presenting papers	**50**
RES-08 house made tools	**44**	= **i**	ACT-08 copying documents	**20**
RES-09 personal expertise	**55**	J i	ACT-09 phoning	**10**
RES-10 administrative documents	**0**			

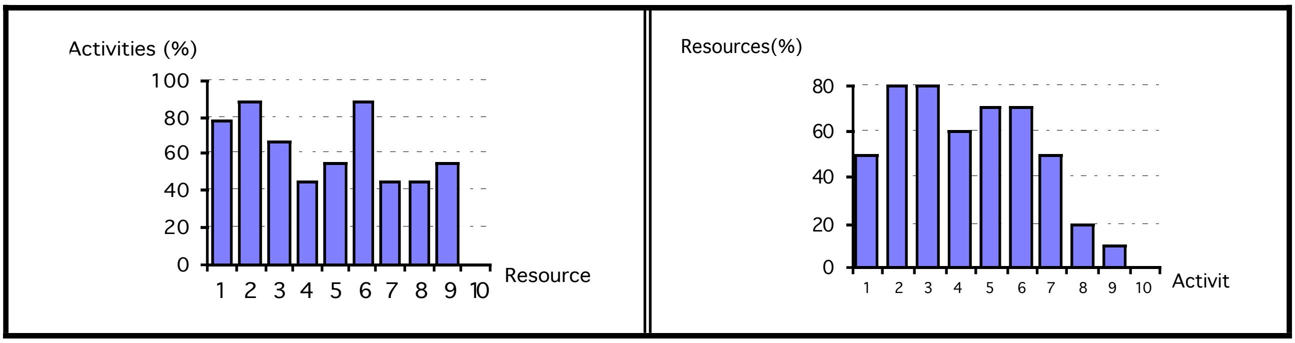


Table 5. Summary of the Ph.D. students interviews

Table 6 contains proposals for possible services for the specific population of Ph.D. students. The crucial activities are then selected by applying a threshold on the corresponding histogram.

service	**description**
document research	A list of bibliographic references is prepared in accordance with the context of work (e.g., technical report, activity report, paper, document destination).
formatting text	Bibliographic lists of references are processed in order to be included in documents in accordance with the destination formats
tips & tricks	Tips and tricks about the resources that can be used to achieve a given activity are proposed
dictionary	The keywords used in the group are checked in order to let their meaning be known to all the members. Informing interest groups about new keywords or definitions.
writing	Text editor with facilities like: format models (reports, papers, letters); reference adding; keyword parser, drawing ...

Table 6. Ph.D. students candidate services

Validation by the group (Step 3)

The services are validated by the group of Ph.D. students.

Description of services (Step 4)

Steps 1 to 3 are repeated for all the other groups: researchers, technical staff, clerical staff. Services selected by the different groups are included into a global document (Table 7), which must be approved by all groups.

The final choice

From the synthetic results, we selected a subset of services that we believed could help new Ph.D. students. Thus, we decided to implement the following agents:

- Dictionary - to manage the concepts used in the laboratory (i.e., ontology)
- Tips-&-Tricks - to inform about equipment, software, procedures, as well as personal remarks about the working environment
- Notebook - to preserve information about the laboratory (people, adm. data...)
- Bibliographic - to manage a bibliographic reference knowledge base and

- Writing - to guide users in writing repetitive documents

	Ph.D. students	**Researchers**	**Secretaries**	**Techn. Support**	$\frac{Serv.}{\sum Serv.}$ %
formatting text	1	1	1		**75**
writing	1	1	1	1	**100**
project monitoring		1	1	1	**75**
admin. data-base			1	1	**50**
tips & tricks	1	1	1	1	**100**
document research	1	1			**50**
dictionary	1	1			**50**

Table 7. Ph.D. Synthesis of the candidate services

Detail of the agent contents (steps 4 to 8)

The remaining four steps—writing scenarios, building mock-up, identifying competencies at each step of the scenarios—are quite straightforward and are also found in approaches for designing man/machine interfaces. They will not be detailed in this paper.

Results

Not very surprisingly the selected agents were similar to the ones proposed by an informal polling of researchers. The main benefit of the SAAS approach resided with the amount of detailed information collected in the accumulated cards, which facilitated the construction of the different scenarios, hence the definition of the interaction protocols, and allowed to give a priority to the deployment of the particular agents.

The resulting system however still waits to be fully deployed, mainly because of the complexity of the underlying OSACA platform, and to the amount of work needed to encapsulate interactive legacy systems.

4. CONCLUSION AND DEVELOPMENTS

In this section we would like to indicate why we found that other methods were not suitable, to give some conclusions about the use of SAAS in the MEMOLAB project, and to indicate some possible extensions of the work.

4.1. Other approaches

We mentioned in the introduction our trials with other methods developed in other domains.

Methods for developing knowledge-based systems

In all the methods we have tried for developing knowledge-based systems we found that

- methods are focused to small restricted domains of knowledge
- usually the choice of the domain is already known
- most often, the knowledge handled by such methods is quite deep and is owned by experts

- the quality of the dialog between the expert and the knowledge engineer is essential.

Within the MEMOLAB project, we were interested in modeling expert knowledge, but also in more mundane knowledge used everyday and quite important for the proper functioning of the research team that we call **usage knowledge**. In addition, we were looking for an acquisition technique which would be independent from the technical background of the interviewed users (researchers, technical or clerical staff, students), and we wanted to minimize the interviewing time. Thus, the methods we tried were not satisfactory in this respect.

Methods used in the context of knowledge management for enterprises

Here again, a good part of the proposed techniques deal with how to organize known and rather well structured pieces of information. Analyses are centered on information flows, which is not the case of the MEMOLAB project.

Methods used for software engineering

Software engineering methods are used for structuring information (e.g., MERISE, SADT, SA, SA-RT) or for developing object systems (BOOCH, OOA and OOD, Shaler-Mellor, OMT, or now UML). We used such methods in the past for structuring object-oriented databases and applications, but found that because of the level of granularity at which they operate, they are not really suitable for determining services that one could wrap up within agents.

A method for agent systems

Larvet (94) proposes a specific method, MASOA, for analyzing complex 'agent-oriented' systems. The method includes 10 steps for representing knowledge with agents. It aims at determining for each agent the best way to structure its knowledge, but assumes that the type of knowledge that each agent will own is known.

General remarks

Most of the methods we have tried look alike in the sense that they analyze processes, the constitutive parts and related actions of which are known beforehand. According to the domain or the objectives of the analysis, one will try to reproduce some functionality of the process (software engineering), the information flow (enterprise modeling), remember intermediate results (knowledge management). Such methods are well adapted to stable processes, where the method criteria guide the analysis for each process to be observed. We do not think that they can be easily applied to our problems concerning multiagent systems for the following reasons:

- most methods start from a known process, in our case one must establish what the process really is.
- most existing methods take into account constitutive elements of a process as an ordered optimized set of modules allowing to execute a task. Formalisms use concepts like material flow, control flow, or task, for describing processes and deriving conclusions about their optimal configuration. Very few method integrate human actions. In our case however, we can find different procedures for doing the same action. Furthermore, the *usage knowledge* is constantly changing in time.

4.2. Comments about the use of the SAAS method.

From our limited experience in applying the SAAS method we can make some remarks about the interviewing steps and about the findings.

About the interview method

- Representing each user's activities in a graph was easy. The interviews never lasted more than two hours.

- An activity must produce only one kind of result (models, programs, articles, reports, seminars). In the case where more than one product was identified, the activity was divided into sub-activities. Activities with two or more products lead to graphs that are difficult to understand.
- We must familiarize people that have to be interviewed with the meaning of concepts like activity, sub-activity resource and product, and in particular with the fact that an activity may have several sub-activities but should lead to a single product .
- In our case a consensus was reached quite easily. However the groups which we considered are rather small, and it is not clear whether this could scale up with larger group without straining the knowledge engineer.
- We were surprised that activities dealing with 'analysis' and 'evaluation' were never mentioned by most of the persons we interviewed. We assume that it is explained by the fact that such activities are not explicitly represented in the working behavior. They are carried out immediately after running a program or testing a prototype and are used as a feedback when executing activities like: developing; programming; or form the basis of writing activities.

Limits of the SAAS method

The SAAS method is not intended to produce a model of the enterprise, but simply to determine a possible number of services that can be implemented as independent cognitive agents. The SAAS methods is exclusively descriptive and works bottom up. Existing services (i.e., working habits) can be easily identified. However, it is impossible to detect the group's needs for *new* services. The reason is that only the *performed* activities are the basis for the SAAS method. By their nature, services that are needed but do not yet exist are *never mentioned* during the interviews.

4.3. Possible Developments

Although the role of the SAAS method ends with the creation of agents, as a side-effect it produces actigraphs. Such graphs are models of the activity of *a particular user*. Thus, an idea would be to use an idealized actigraph as a background reference for following the work of a user automatically (since in our case users essentially use computer terminals). Furthermore, observing and comparing the activity nets, leads to think that they contain interesting information about the users' working strategies. Such strategies could be modeled and re-used for helping newcomers as an on-line help when they encounter problems in their everyday work.

ACKNOWLEDGMENTS

We are grateful to Mr. Grundstein who spent some time explaining his method to us, and to all the members of the laboratory who took time to answer willingly our questionnaires.

REFERENCES

de Azevedo H., Scalabrin E.E., Barthès J-P. 1995. Interaction Homme/Machine dans le projet MEMOLAB, pour la capitalisation des Connaissances d'un groupe de recherches. In Actes des 3èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, Chambéry-Saint-Bandolph. 15-17 Mars. France.

Coad P., E. Yourdon. 1990. Object-Oriented Analysis. Prentice Hall.

Coad P., E. Yourdon. 1991. Object-Oriented Design. Prentice Hall.

Decker S., Daniel M., Erdmann M., Studer R. 1997. An Enterprise Reference Scheme for Integrating Model Based Knowledge Engineering and Enterprise Modelling. In Knowledge Acquisition, Modeling and Management. 10th European Workshop, EKAW'97. Enric Plaza, Richard Benjamins (Eds). Lecture Notes in Artificial Intelligence. Springer. 81-96.

Ermine J.-L., Chaillot M., Bigeon P., Charreton B., Malavielle B. 1996. MKSM, a Method for Knowledge Kanagement. In Proceedings of the ISMICK'96 - Management of Industrial and Corporate Knowledge. Schreinemakers (Ed). Egon. pp. 288-302.

Fox, M.S. 1988. An Organizationanl View of Distributed systems. In Readings in Artificial Intelligence. Bond et Gasser (Eds). Morgan Kaufman. 140-150.

Grundstein M. 1996. CORPUS - An Approach to Capitalize Company Knowledge. In Proceedings of AIEM-4 - The fourth International Workshop: Artificial Intelligence in Economics and Management. Tel-Aviv. January 8-10.

Grundstein M., Barthès, J-P. A.. 1996. An industrial View of the Process of Capitalizing Knowledge. In Knowledge Management - Organization, Competence, and Methodology. Jos F. Schreinemakers (Ed). Advances in Knowledge Management, Vol 1. Ergon. 258-264.

Larvet P. 1994. Analyse des systèmes : de l'approche fonctionnelle à l'approche objet. Collection Informatique Intelligence Artificielle. InterEditions, Paris.

Malvache P. and Prieur P. 1993. Mastering Corporate Experience with the REX Method. In Proceedings of ISMICK'93 - Management of Industrial and Corporate Knowledge, IIIA. Compiègne, 27-28 October. 33-41.

Scalabrin E. E., Vandenberghe L., de Azevedo H., Barthès J-P. A.. 1996. A Generic Model of Cognitive Agent to Develop Open Systems. In Advances in Artificial Intelligence. Series Lecture Notes in Artificial Intelligence. D.L. Borges, C.A.A. Kaestner (Eds). Springer. 61-70.

Vogel C. 1988. Génie Cognitif. Masson. Paris.

Wiig, K.M. 1993. Knowledge Management Foundations: thinking about thinking - How People and Organizations Create, Represent and Use Knowledge. Schema Press, Arlington Texas.

Wiig, K.M. 1994. Knowledge Management: The Central Management Focus for Intelligent-Acting Organizations. Schema Press, Arlington Texas.