FAES: A CASE TOOL FOR INFORMATION ACQUISITION

Ana Paula Pinho Gilvaz*
Johnson Wax
Av. Comandante Guaranys 599
22275-610 - Rio de Janeiro, Brasil

Julio Cesar Sampaio do Prado Leite*
Departamento de Informática - PUC Rio
Marquês de São Vincente 225
22453-900 - Rio de Janeiro, Brasil
e:mail: julio@inf.puc-rio.br

Abstract

It has become a trend that new CASE based tools be more focused. FAES is such a case. It assists the software engineer in the interview process during an information systems' requirements phase. FAES inherited important concepts from the IS literature and embodies them in a conceptual model that drives the interview process. Special heuristics associated with the conceptual model were proposed and tested before inclusion on FAES. The goal of those heuristics is to provide an up front analysis of the information as the interview proceeds. Artificial Intelligence techniques were used in order to provide FAES with the intelligent assistance capability.

1. Introduction

Computer Aided Software Engineering has been helping the introduction of software engineering principles in several organizations [2] mainly by supporting the high end tasks of software development. Several commercially available tools implement several methods, ranging from structured methods to ones that are object oriented. These tools are fundamentally composed of graphical editors, repositories and syntax checking. Some of them provide automated assistance for code production, but none of them provide adequate support for the task of information elicitation.

Our work is aimed at providing support for the software engineer in eliciting information for corporate executive information requirements. We used well established IS techniques to build a conceptual model behind FAES. Using a conceptual model and some analysis heuristics we managed to provide the software engineer with an automated support for finding out important information in a given information system. As such the work we will describe here is focused on a particular example of computer automated support, namely the elicitation of information through interviews. Our tool, a prototype, was built with the purpose of supporting and evaluating our strategy. We strongly believe that its ideas as well as its architecture can seed the construction of a tool that could be integrated in a upperCase type of product.

Rich and Waters [9] have pointed out that CASE technology would need intelligent assistance in order to enhance the productivity as well as to support upstream activities. They also advocate the use of deep representation strategies to support CASE technology. FAES, a prototype tool, is an example of applying intelligent assistance to the task of information gathering. Although FAES uses a simple conceptual model, its shallowness is a positive factor in the tool's performance, since it is geared to the IS domain, and therefore more easily trusted by users.

As well noted by Loucopoulos and Karakostas [7], CASE tools have been concentrating on supporting the *population* of the repository and not on *constructing* the requirements' product itself, since they usually do not support the transition process from informal to formal. Our proposal is tackling this missing aspect of CASE technology.

FAES is the central part of an interview process that covers three basic interview questions: What to ask?, How to ask?, and Whom to ask? The process has automation support for the first two questions and relies on general guidelines for the third question. Following the process, we build a knowledge base that is organized according to the conceptual model and analyzed according to special heuristics. The conceptual model was built upon three well-know information system techniques: BSP (Business System Planning) [5], CSF (Critical Success Factors) [10] and E/M (End Means Analysis) [12] and follows the integration model proposed by [12]. The heuristics were developed for improving the knowledge base consistency and were improved by several test cases.

The FAES knowledge base is an important factor for providing an organized model of corporate information. The availability of intelligent assistance is a key point in

* This work is supported in part by CNPq grant n. 510845/93-2.

making FAES a tool for improving the productivity of a software engineer and the quality of the information kept in the model. We also have to stress that the tool provides support for the boring clerical tasks of asking, storing and organizing questions and answers.

The article is organized as follows. Section 2 provides a general description of the interview process. Section 3 describes the conceptual model, along with some of the heuristics used to analyze the model and the automation strategy. Section 4 gives details about the assistant. We conclude by comparing our approach with previous work, discussing its integration with existing CASE technology and lining up future research.

2. The Interview Process

We describe the general process using SADT [11], a graphical language well suited for general process SADT actigrams determine that the boxes contain verbs or verbal phrases and the arrows on the left of the box are *inputs,* the arrows on the right are *outputs*, the arrows on the top are *controls,* and the arrows on the bottom are *mechanisms.* Our objective is to give a general view of the interview process. Our viewpoint is the one from the authors of the process description.

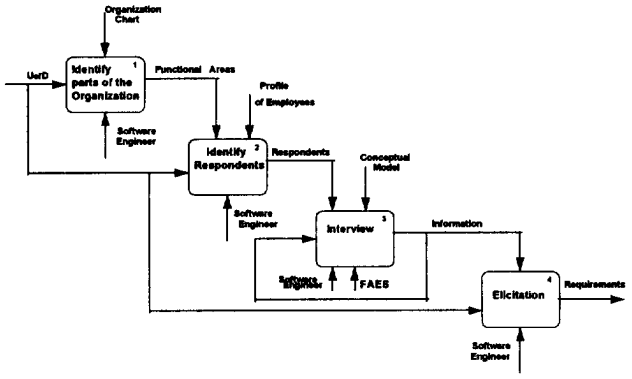


Fig. 1

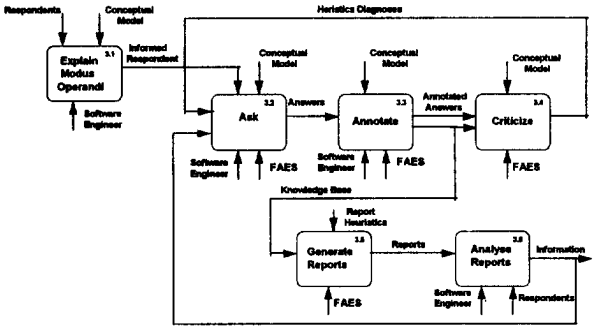


Fig 2

Figure 1 gives a view of the context where the interviews are conducted and how the respondents are determined. It is important to note that identifying the right persons to ask is not a trivial task. Another important highlight is that the information, gathered by the interview and filled in as the knowledge base, will be used as a source for a more detailed elicitation process. The *input* UofD means Universe of Discourse [6] and is defined as: "UofD is the overall context in which software will developed and operated. The Universe of Discourse includes all sources of information and all the people related to the software. It is the reality trimmed by the set of objectives established by those demanding the software".

Figure 2 details the main activities during interview. The first (3.1) and the last one (3.6) are activities at which the assistant (FAES) is not directly involved. The other activities are performed with the assistance of the tool. The interviews are conducted individually with selected persons (Figure 1). The software engineer asks the questions suggested by FAES and annotates the answers while trying to be as factual as possible, considering the respondents given answers and trying to be as clear as possible. The software engineer can also comment on the answers he is annotating. It is important to note that the tool offers two feedback mechanisms: one at the time the question is being annotated and the other as the interview ends. At the end of the interview a report is generated which mirrors the knowledge base and provides a diagnosis of the captured information.

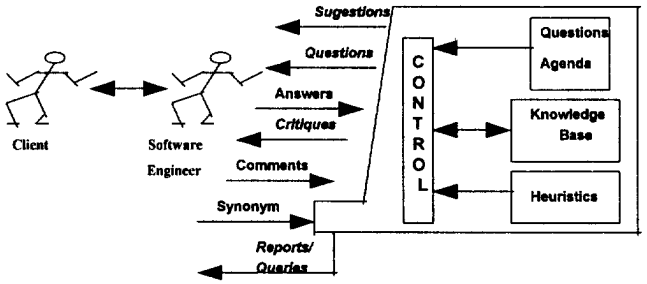


Fig. 3

The interview assistant (Figure 3) applies a basic set of questions that would fill in the conceptual model. The interview assistant has a set of heuristics based on the conceptual model and also on general common sense. These heuristics have been written to validate the answers, verify the existence of relationships between the answers and discover the need for more answers.

Figure 3 shows the general interaction of the assistant with the software engineer. The client (the respondent) does not have direct contact with FAES. The assistant

offers the questions, criticizes the answers based on heuristics and on the information kept in its knowledge base. The software engineer can post observations about the answers and can also associate terms as synonyms.

FAES has four basic components: control, questions, knowledge base and heuristics. Control deals with the interface, the order of questions and heuristics application. Heuristics are activated by a particular question or by the end of the interview. The questions are based on the conceptual model and contain the information necessary to instantiate the model. The knowledge base stores the answers, the diagnoses and the entries made by the software engineer (observations and synonyms).

3. The Basis for Interview Automation

The automation process is based on the conceptual model, on a general strategy and on heuristics. We detail each of these aspects.

3.1 Conceptual Model

The conceptual model is based on Wetherbe's work on executive information requirements [12]. According to Wetherbe, a common mistake made in determining information requirements is to ask the wrong question: "What information do you need from the new system ?" Although this is the obvious question, it is not always helpful to clients attempting to determine what information they need. In order to minimize this problem, Wetherbe proposes an approach to interviewing that uses indirect questions. The interviewing scheme is composed of types of questions from three methods/techniques defined in the table below.

Approach	Techinique	Developers
Specify problems and decisions	BSP's part dealing with executive interview	IBM
Specify critical factors	CSF	Rockart
Specify effectiveness criteria for outputs and efficiency criteria for output generation processes	E/M analysis	Wetherbe and Davis

Each one of these methods deals with a different aspect in the process of information gathering. The figure below illustrates their individual approaches.

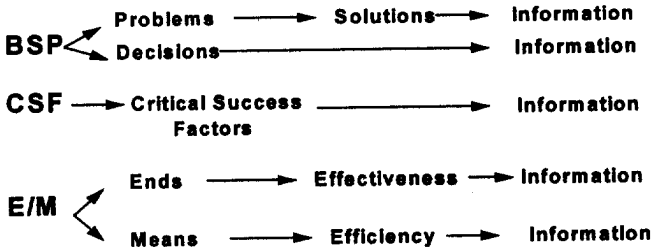


Fig. 4

Although the approaches depicted above are different, they share the same goal, which is to obtain relevant information about the context in which the software will operate, as a result, each one provides a different perspective. Putting them together as the basis for an interview process follows the idea of reframing [1], in which the same facts are re-communicated in a different frame. The objective is to enhance the chances of completeness and increasing redundancy by means of creating different frames for the clients (respondents).

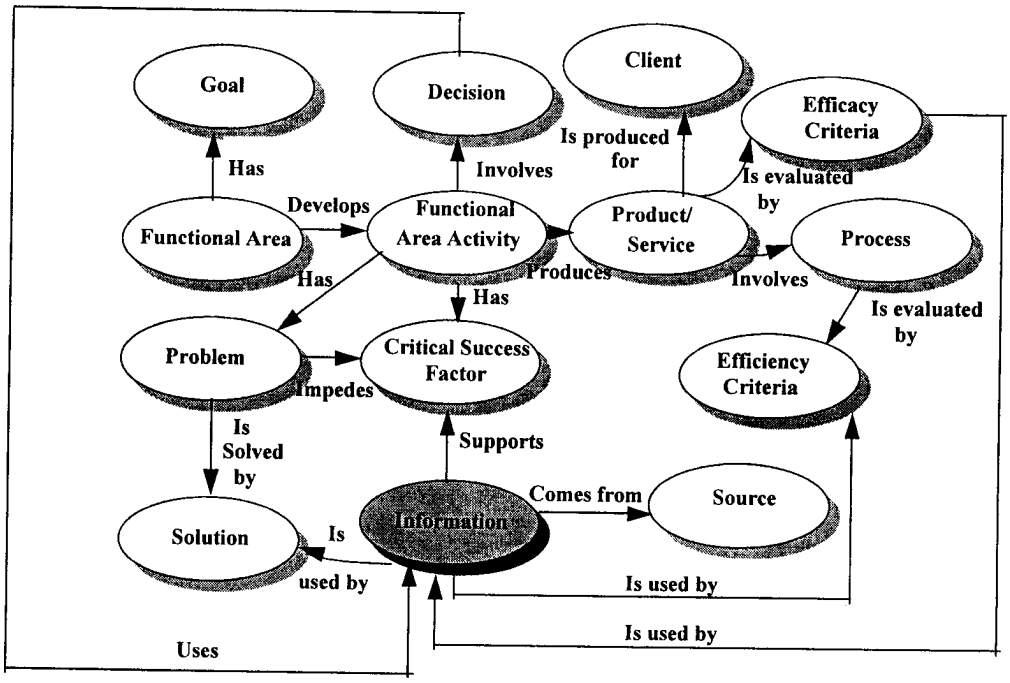


Fig. 5

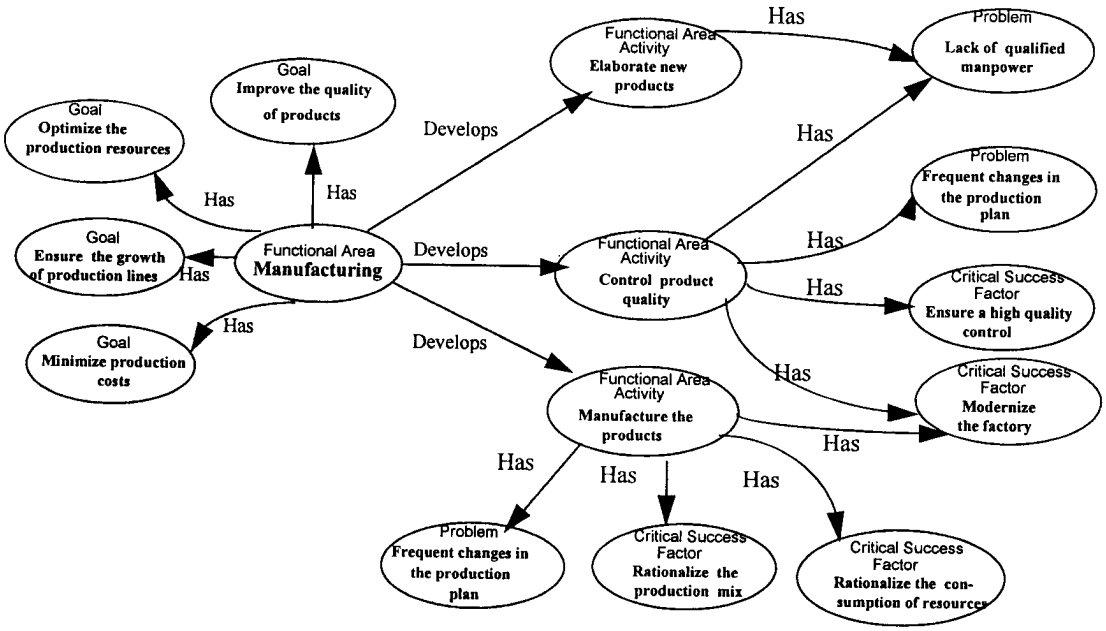


Fig. 6

Figure 5 shows the conceptual model we developed based on Wetherbe's approach. The conceptual model links the different types of questions and includes a lot of new information found necessary to support the interview. The model nodes represent the information to be defined

and the arcs, the relationships among the information. The nodes serve as
basis for the questions posed by the assistant. The arcs provide relationships that will help the analysis of the answers. The model has passed through several versions, as we tested its instantiation with a couple of case studies.

The questions used by these three techniques are structured around the idea of functional area. The concept of functional area determines the respondents, in which the managers of that functional area must be included. We can identify different parts of an organization when applying our process (see Figure 1). Gilvaz [4] provides more detail on functional area identification. Each functional area must have a set of objectives that will be achieved by activities. As a result , we decided to have activities as the main entity in our model [4].

Figure 6 shows a partial view of an instantiated model. It only shows two types of relation, with the more complex relationships being exemplified in Section 3.3. The above example was based on the manufacturing department of cleaning products from a chemical manufacturer, which served as the functional area. Below we show part of the interview that created Figure 6 (questions in boldface, answers indented). This interview was conducted with a middle manager from the cleaning products' division. It is important to observe that all our case studies were conducted in Portuguese, and here we provided a free context translation. So, some of the terms may not be the same as if the interviews were conducted with and or by English speaking individuals (domain vocabulary).

. Functional Area : Manufacturing.

. What are the goals of <Manufacturing> ?

- Improve the quality of products.
- Optimize the production resources.
- Ensure the growth of production lines.
- Minimize production costs.

. What are the activities of <Manufacturing >?

- Control product quality.
- Manufacture the products.
- Elaborate new products.

. What are the problems associated with <Control product quality>?

- Lack of qualified manpower.
- Frequent changes in the production plan.

. What are the problems associated with <Elaborate new products>?

- Lack of qualified manpower

. What are the problems associate with <Manufacture the products> ?

- Frequent changes in the production plan.

. What are the critical success factors associated with <Control product quality> ?

- Ensure a high quality control.
- Modernize the factory.

. What are the critical success factors associated with <Manufacture the products> ?

- Rationalize production mix.
- Rationalize resource consumption.
- Modernize the factory.

It is important to stress that the questions presented above (in boldface) are provided to the software engineer by the tool. As such, we see that the clerical task of not forgetting a question is taken care of by the software. It is also important to stress the spreading of information that happens in the process; for example: see the <manufacturing> and <control product quality> chain depicted above. It is also important to stress that it was not our objective to validate the conceptual model. Despite the fact that we built on previous work, we could not find in our case studies any evidence that would undermine Wetherbe's approach. That is to say, that although our case studies with FAES showed that the model fitted in well with its purpose, we did not conduct extensive studies with the objective of validating Wetherbe's model.

3.2 Automation Strategy

The questions that are used to fill in the model use the concept of information chaining, that is each question is composed of a fixed part and a variable part (< >). The variable part is an answer given to another question already answered, thus making a chaining process, since each of the answers of a given question will produce a different question for one of the fixed patterns. Figure 7 illustrates the possible chaining for the instantiation questions. The strategy uses four question types: instantiation questions, relation questions, inconsistency questions and investigation questions. The instantiation questions are driven by the basic conceptual model, the other types of questions are driven by the heuristics (Section 3.3). Following we present all the 36 questions

that FAES suggests the software engineer to ask.

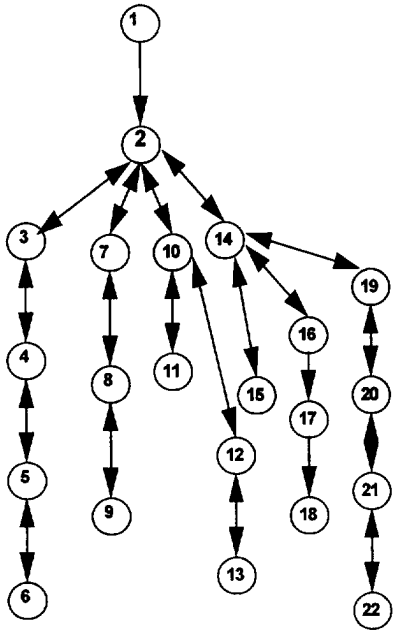


Fig. 7

Instantiation questions

Instantiation questions are the basic questions used to fill in the model. Note that each variable part (< >) creates a chain between the answer and the next questions (Figure 7).

Functional Area:

1. What area the goals of < functional area >?
2. What area the activities of < functional area > ?
3. What are the problems of < activity > ?
4. What area the best solutions for < problem > ?
5. What information support < solution > ?
6. Who provides < information > ?
7. What are the decisions related to < activity > ?
8. What information support < decision > ?
9. Who provides < information > ?
10. What are the critical success factors of < activity > ?
11. What problems impede <critical success factor > ?
12. What information support <critical success factor >?
13. Who provides < information > ?
14. What are the products/services of < activity > ?
15. Who are the clients of < products/services > ?
16. What efficacy criteria evaluate <products/services>?
17. What information is used to evaluate < efficacy criteria > ?
18. Who provides < information > ?
19. What are the key processes used to generate or provide <products/services> ?
20. What are the efficiency criteria to evaluate < process > ?
21. What information is used to evaluate < efficiency criteria > ?
22. Who provides < information > ?

Relation Questions

These are yes/no questions derived from relation heuristics. Below we list the possible questions, which will be activated based on the heuristics pre-conditions

23. < Critical Factor > is critical for < Goal > ?
24. < Solution > improves the decision < Decision > ?
25. < Critical Factor > interferes in < Decision > ?
26. < Goal > may be evaluated by < Efficacy Criteria>?
27. < Goal > may be evaluated by <EfficiencyCriteria>?
28. < Solution > contributes to < Critical Factor > ?
29. < Information > support < Critical Factor > ?
30. < Information > support < Decision > ?

Consistency Questions

These questions check possible inconsistencies.

31. Client answered before that < Information > was provided by < Source > ! Do you confirm?

Investigation Questions

These questions are related to the validation heuristics, and work basically as a final checklist.

32. Is there any other goal (< list of goals >)?
33. Is there any other critical factor (< list of critical factors>)?
34. Is there any other information that supports < Solution > (< list of information >)?
35. Is there any other information that supports < Critical Factor > (< list of information >)?
36. Is there any other efficacy criteria (<list of efficacy criteria>)?

3.3 The Heuristics

The heuristics were developed by taking the model as the basis, using common sense and the feedback of the case studies. The actions recommended by the heuristics should be confirmed before any changes are made to the model, so a heuristic generates a question that aims to validating its conclusion. We classified the heuristics in: relationship

heuristics, completeness heuristics, validation heuristics and consistency heuristics.

Relation Heuristics - should discover the relations that are not previously defined in the model.

Example

Given a goal and a critical factor, if there is a term—named key-term—which is common for both the goal and the critical factor, then the critical factor may be related to the goal.

Question:

< Critical factor> is critical to the < Goal> ?

< Ensure a high quality control> is critical for <optimize product quality>?

In this case if the question was confirmed, a relation between goal and critical factor would be instantiated, as indicated by the double line in the graph above.

It is important to stress that the procedure used to check for key-terms does not only consider a perfect match. We use a partial match scheme, that considers a match if at least 60% of the letters from the shorter term are present in the longer term.

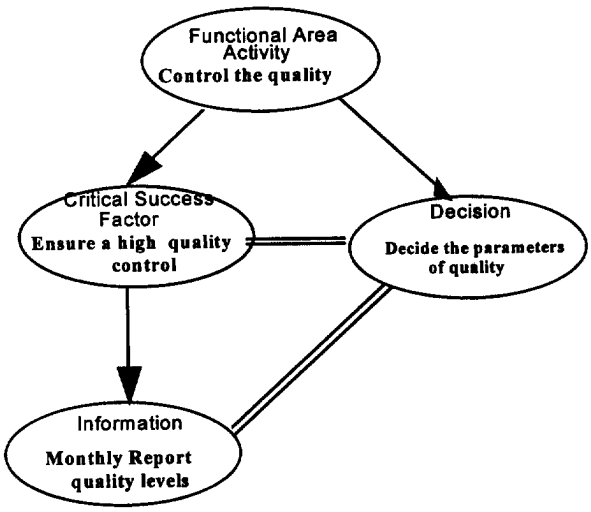
Completeness Heuristics - happen whenever a relation heuristic is confirmed and has an objective of relating model entities with information already available for other entity.

Example:

Given a critical factor and a decision and also considering that a previous heuristic found a relation between them, then the information that supports the critical factor may also be relevant to the decision.

Question

< Information > supports < Decision > ?



<Monthly report of quality levels > supports <decide quality parameters>?

In this case the critical factor <ensure a high quality control> and the decision <Decide the quality parameters> were related by the confirmation of a relation heuristic that happened during the interview. As a consequence, the completeness heuristic was activated.

Validation Heuristics - Checks if relations heuristics were not activated, this fact may show that the client did not properly answer a question or gave an incomplete answer.

Example:

If there is a goal that is not related to a critical factor, verify if there are other critical factors.

Question:

Is there any other critical factor (< list of critical factors >)?

FAES uses a standard production system scheme for dealing with the above heuristics. Once a given node in the conceptual model is filled in by one of the questions from the automation strategy, the control mechanism activates the production memory to check if a rule will fire given the state of the knowledge base. These types of rules fire during the interview process. Other types will only be fired once the interview has ended (validation heuristics).

4. FAES

FAES was built with the main idea of assisting and not replacing the software engineer during the interview process. Its main characteristics are:

a) guide the software engineer during the interview, providing a set of questions,
b) providing the facilities of a notebook,
c) playing the role of an assistant by analyzing answers and proposing new questions,
d) providing access to the knowledge base by reports or queries,
e) making possible the storage of observations,
f) using a dictionary which enhances the analysis power of the tool.

FAES was developed using an object oriented language, ENFIN, and a database tool, SQLBase. ENFIN is a Windows compatible software and as result has the advantage of easily interfacing with other software.

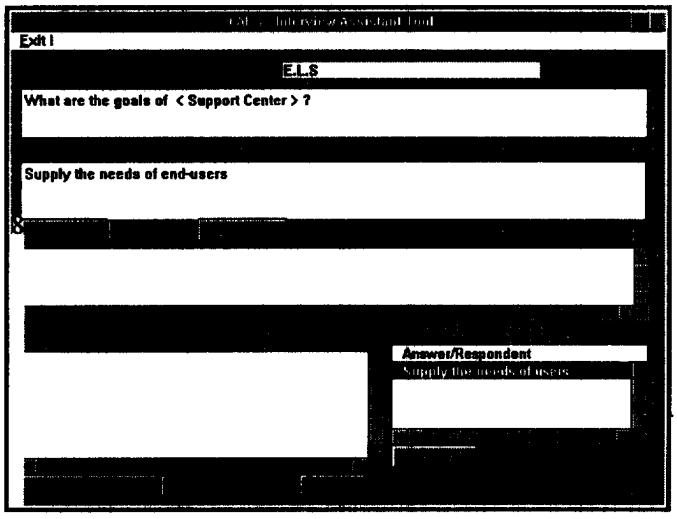


Fig. 8

Figure 8 shows a FAES window. On its top it poses a question to be asked to the client. The software engineer will use the <Answer> frame to type in the answer. A frame labeled <Preview Answers> shows answers previously given by other respondents for this same question and this same functional area (not yet fully explored, see Conclusion). The OBS bottom makes it possible to add comments to the answer. The Synonym bottom makes it possible to associate chosen terms in the <Answer> with other terms. In the case of multi-word terms, the tool will group them by adding hyphens between the words. The <Questions triggered by heuristics> frame shows all the questions generated by the heuristics. In order to answer a question posed by a heuristic, a special window is activated. (Figure 9).

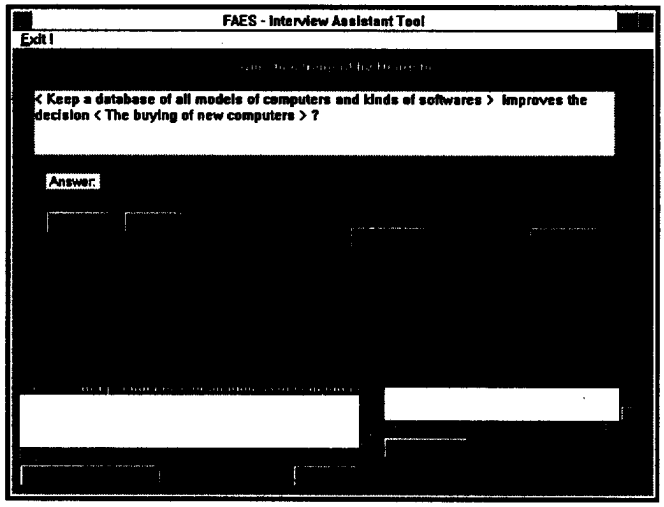


Fig. 9

Figure 9 shows a window activated by a relation question that requires a yes/no answer. It is noteworthy that the button <About...> provides a justification of why that heuristic has been activated. The justification is provided by the window showed in Figure 10.

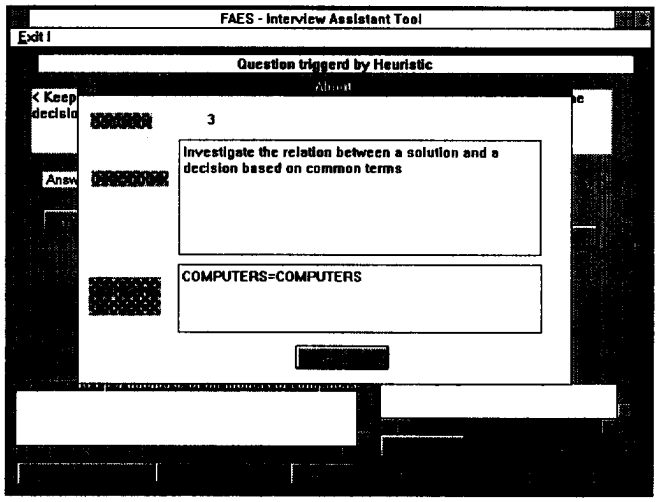


Fig. 10

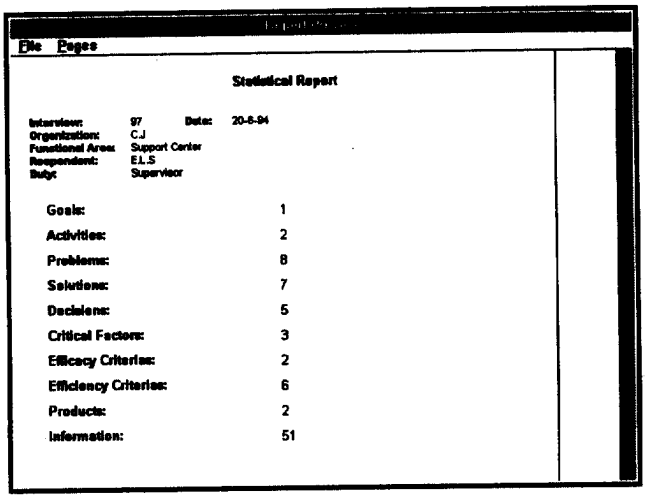


Fig. 11

At the end of an interview, FAES generates reports that not only provide feedback of the interview process, but also provides statistics about the process (Figure 11). The table below shows other statistics, in this case, related to the heuristics used during the case study.

Relation heuristics activated -- 25
Completeness heuristics activated -- 25
Consistency heuristics activated -- 0
Validation heuristics activated -- 3
Relationships instantiated by relation heuristics -- 19
Relationships instantiated by completeness heuristics -- 20
Information acquired by validation heuristics confirmation -- 0

Figure 11 and the Table above show a particular set of data related to one of the case studies conducted with FAES. In this case, they picture an interview conducted with a member of Johnson's Wax information support center. Figure 11 shows the number of nodes' types elicited during the interview. We can see that for this case one goal, two activities and three critical success factors were found. It was not a surprise that the node information was the one that held more answers, since the main objective of FAES is to find information. The Table shows the heuristics that were activated in this case study in order to generate 51 instances of the information node. It is also important to observe that in this exercise, departing from the basic 36 questions, FAES had suggested 137 questions to the software engineer due to the information chaining concept.

5. Conclusion

We have presented an interview process in which an automated assistant plays the main role. The assistant helps software engineers in conducting interviews with clients. The interviews' target is the discovery of information pertaining to the Universe of Discourse in which the software will operate. The assistant uses a general model which was based upon previous work on information systems [12]. The assistant uses general heuristics to improve the quality of the knowledge base which will store the information gathered in the process. The questions posed by the assistant are linked together by the concept of information chaining. FAES is an example of an upstream focused CASE tool, in this case supporting the elicitation of information for corporate information systems.

Our results so far are encouraging. The prototype is operational and we have used it in different case studies. We believe that the main achievement of this research was the one of automating the interview process and the productivity gains obtained from using the tool. We can claim this, based on the data collected during our case studies. Prior to building FAES, we used a prototyping approach in order to tune the model, the questions and heuristics. This prototype effort consisted of applying manually the strategy described in the paper. We have conducted several manual interviews following the strategy, and as a result the time and the volume of paper generated quickly made us believe in the impracticability of performing the task without automated support. After building and using FAES it was clear to us the importance of intelligent assistance as well as the clerical support provided by FAES.

As it stands, FAES is sitting on its own. Proper integration with existing products and environments is future work. We believe that a strong point in favor of FAES integration with upperCase is the usage of FAES repository as an initial data dictionary for a given application. In the FAES repository (Figure 5) the entity **information** will provide a first list for the data dictionary entries. More than that, the repository/conceptual model allows for future automatic analysis, in addition to the existing interview analysis heuristics. We also believe that the fact of using SQLBase will make it easier for the integration with other tools/environments based on the Windows platform.

Overall the results achieved by FAES's conceptual model are the following:

a) usage of a general information system framework,
b) usage of domain language,
c) organization of information gathered during the interview,
d) easy manipulation of gathered information.

Reubenstein [8] and Drake [3] also dealt with interview automation. Reubenstein has developed a general assistant to gather information in the process of knowledge acquisition, but his strategy is based on a previous encoded knowledge base, which will serve as an oracle for the acquisition of requirements. Reubenstein's assistant user is the software engineer who will have the help of an intelligent tool to guide and criticize the requirements formation. Drake proposed an assistant to guide the client in answering questions anchored on a general model geared towards input/output. Our work is similar to Reubenstein's in terms of serving the software engineer and not the client, it is similar to Drake's in the usage of a general model.

Unlike Reubenstein we do not use previously encoded domain knowledge. Our knowledge base is populated as we proceed through the interview process. Our approach is also shallower than that used by Reubenstein, since we limited the range of gathered information by using a fixed frame (the conceptual model) and general heuristics, but we can use the conceptual model in a range of different domains for computer based organization support. With respect to Drake's, we believe that by using a conceptual model geared to information systems concepts, we can have a better performance, and we do not put on the clients shoulder the responsibility of using a tool.

As pointed out, future work should look into integration, but besides that, other internal aspects have to be worked out. First there is a need to work with the interface aspects of the tool, improving its ease of use. Second FAES needs to be used in many other case studies. Third we would like to explore other analysis heuristics upon the repository and we would also like to explore the possibility of viewpoint analysis [6] upon the repository by comparing the previous answers of different respondents.

References

[1] Bostron R. P. *Successful Application of Communication Techniques to Improve the Systems Development Process* Information & Management 16 ,1989 pp. 279 - 295

[2] *Chikofsky, E. Introduction in Computer-Aided Software Engineering (CASE) 2nd Edition IEEE Computer Society Press, pp 1, 1993*

[3] Drake J. M., Xie W. W., Tsai W. T. *A Case Study of Requirement Analysis where End Users take an Active Role* Proceedings of the 15th International Conference on Software Engineering, IEEE Computer Society Press, 1993

[4] Gilvaz, Ana Paula P.and Leite, Julio Cesar, S. P. *FAES - Um Assistente para Entrevistas*, Proceedings of the VIII Simpósio Brasileiro de Engenharia de Software, Sociedade Brasileira de Computação, Out. 1994

[5] *Information Systems Planning Guide, Application Manual*, GE20-0527-3, Third Edition, IBM Corporation (July 1981)

[6] Leite, Julio Cesar S P. and Freeman, P.A. *Requirements Validation Through Viewpoint Resolution,* IEEE Trans. on Sof. Eng., Vol. 17, No. 12, Dec. 1991, pp-1253--1269.

[7] Loucopoulos, Pericles and Karakostas, Vassilios *System Requirements Engineering,* McGraw Hill International Series in Software Engineering, 1995.

[8] Reubenstein Howard B., Waters Richard C. *The Requirements Apprentice: Automated Assistance for Requirements Acquisition*
IEEE Transactions on Software Engineering, 1991, vol 17, (3)

[9] Rich, Charles, Waters Richard C. Knowledge Intensive Software Engineering Tools, IEEE Transactions on Knowledge and Data Engineering, 1992, vol 4, (5), pp. 424 - 430

[10] Rockart, J. F. *Chief Executives Define Their Own Data Needs*
Harvard Business Review, 1979, (March-April), pp. 81 - 93

[11] Ross, D.T. Structured Analysis (SA): A Language for Communicating Ideas, IEEE TRans. on Software Eng., vol 3, no 1 (Jan 1977) pp. 16 - 34

[12] Wetherbe James C. *Executive Information Requirements: Getting It Right*
MIS Research Center Carlson School of Management University of Minnesota , MIS Quarterly, March 1991 pp. 50 - 65