Proceedings of the 29th Annual International
Conference of the IEEE EMBS
Cité Internationale, Lyon, France
August 23-26, 2007.

A Framework for Automated Evidence Gathering with Mobile Systems Using Bayesian Networks

A. B. José, T. M. G. de A. Barbosa Jr., I. G. Sene Jr., A. F. da Rocha, L. S. da S. Castro, F. A. de O. Nascimento, J. L. A. Carvalho and H. S. Carvalho

***Abstract*— This article presents a framework that allows the automated gathering of medical evidence, using a methodology based on Bayesian Networks (BN). For such purpose, system software and a programming methodology were developed. The proposed methodology is generic and can be applied to different contexts (application domains) were evidence gathering is executed in mobile devices.**

***Keywords*— Bayesian Networks (BN), symptomatological interrogatory, breadth-first search, Java technology.**

I. Introduction

The technological evolution of computing devices has made the inclusion of such devices into the user's personal space possible. This inclusion facilitates human health monitoring, as it allows the gathering of physiological and environmental information, at any time and in any place [1]-[4].

This technology leads to a great number of possible applications that may allow *in loco* evidence gathering. The collection of such evidences, in many of these applications, can be done by having the patient fill a specific questionnaire. This questionnaire should try to gather as many evidences on the subject as possible.

Bayesian decision theory can be used to generate a probabilistic model, as it allows the estimation of conditional probabilities associated with an *a priori* evidence. Bayesian Networks (BN) can be used to deal with problems that have some degree of uncertainty, such as in the diagnosis of diseases, the optimization of treatments, and the prescription of drugs [5], [6]. Bayesian networks can be modeled such that they can represent questionnaires about specific subjects, and have the advantage of saving probabilistic information supplied by a specialist in the field.

It would be valuable to have a framework that would allow information storage, and also clearly separate modeling tasks performed by the network from the implementation of the logic that controls the delivery of the questionnaire and the evidence gathering.

Software that automatically performs evidence gathering could be a great asset. In this kind of software, the user interface should not be complex [7], in order to limit the interaction time between patient and system, the patient should not be overwhelmed with too many questions, and the mobile system should have a small energy consumption.

The goal of this work is to implement a set of software mechanisms that form a framework capable of executing automated evidence gathering, allowing the use of BN in cell phones. For such purpose we developed a programming methodology capable of separating the modeling of the BN from the implementation of the system for evidence gathering.

In section II, we present the technique for evidence gathering that was used. Section III describes a way of modeling a BN for use in this framework. In section IV, we present the application prototype, and a description of the classes we used. In section V, we present the conclusions drawn from this work.

II. Collecting Evidences Though the Use of Graph Paths

The breadth-first search technique consists on visiting all the nodes in a branch until the end nodes are reached, and repeating this process for all branches [8]-[10]. The process is illustrated in Fig. 1.

In the original *breadth-first search* technique, a search key is generally well-known a priori. The goal of the system software discussed in this work is to allow the execution of the Bayesian Networks that represent the possible questionnaires in computational devices. Thus, there is a perception that the true search keys are reached only at the end of the set of questions, when one fills a questionnaire. Therefore, along the path graph, the user should be questioned about which of the next evidences are consistent with the actual situation.

If the questions are to be used in the questionnaire graphs, questions referring to the nodes that follow the

Manuscript received April 2, 2007.

Alexandre B. José , L. S. da S. Castro , A. F. da Rocha and F. A. de O. Nascimento are with the Department of Electrical Engineering, University of Brasília, Brasília, DF 70910-900 Brazil (phone: +55-61-33072328, e-mail: abjucg@gmail.com, castro@unb.br, adson@unb.br, assis@unb.br).

T. M. G. de A. Barbosa and I. G. Sene Jr. are with the Department of Computing, Catholic University of Goias, Goiania, GO 74605-110 Brazil (e-mail:: talles@ucg.br, iwens@ucg.br).

J. L. A. Carvalho is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: jcarvalh@usc.edu).

H. S. Carvalho is with the Medical School and with the Department of Electrical Engineering, University of Brasília, Brasília, DF 70910-900 Brazil (e-mail: carvalho@unb.br).

current question should be included. As an example, if question C in Fig. 1 was asked, it is necessary to ask the question referring to nodes E and G, because the original algorithm does not provide any kind of heuristics for choosing the most appropriate question, and depends on a choice made by the user.

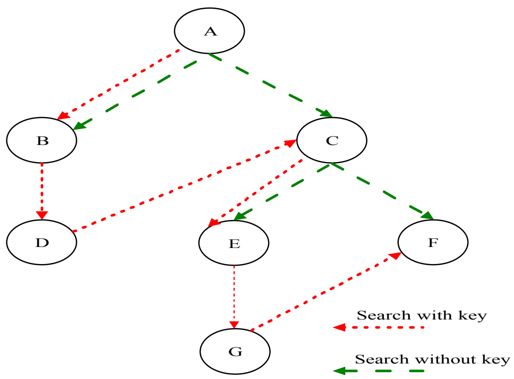

Fig. 1. Breadth-first search.

The computational complexity of the Breadth-first search algorithm discussed above is O(n), as the number of iterations is restricted to the number of nodes in the path, and there aren't cycles in the graph correspondent to this path.

III. Tools for Bayesian Network Modeling

Microsoft Belief Network (MBN) is a Windows software for creating, manipulating and evaluating BN. This software graphically represents the network with a diagram, where variables are presented as ellipses composing the vertices of the graph and conditional dependences are the connections (edges) between variables [11].

MBN stores the graphical structure in a ".dsc" file, which is a text file in a proprietary format. As a contribution of this work, we implemented a Java translator that is discussed in section II.

A. Questionnaire Modeling

The questions that compose the questionnaire must be hierarchy organized in a top-down approach, with general (macro) and detailed (micro) views of the problem.

This hierarchy can be observed in Fig. 2, which shows the topic structure of a Symptomatological interrogatory [12, 13]. Underlined items correspond to evidences that may be checked by the patient.

3. Head
- 3.1. Headache? Yes No
  - 3.1.1. Started: suddenly gradually
  - 3.1.2. Intensity: mild moderate intense very intense
  - 3.1.3. Pain: burning in bursts pulsatile continuous ...

Fig. 2. A Segment of the symptomatological interrogatory.

This questionnaire can be converted into a decision graph. Thus, the questions are represented by the vertices, while the evidence (probability value) are inserted into each graph vertex (node). The questionnaire generates only one antecessor for each node. This means that child nodes are assigned to a single ancestor node, child's parent node. This guarantees a single possible path to a leaf node, the final question.

IV. Framework Implementation

The classes implemented in this work are shown in Fig. 3. Note that one can create its own path through the graph, instead of using an object from the "BSearch" class. Thus, a researcher is able to implement a path that more adequately fits his study. The programmer should be familiarized with the interfaces of the "Nodo" and "GAD" classes, in order to be able to take advantage of their functionalities.

Therefore, there are two possible approaches for using

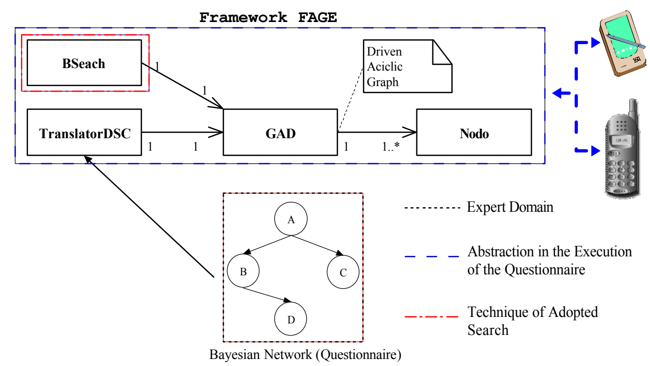

Fig. 3. Framework FAGE.

these classes. In the first approach, the network that represents the questionnaire is described in ".dsc" format, and is imported directly by the classes. In the other approach, a different path is used, and it is implemented in a different class, allowing a new control over the questionnaire execution logic.

A. The DSC Translator

We designed a simplified translator, called TranslatorDSC, which can be ported for executing BN in environments with limited computational resources, such as cell phones and PDAs. An overview of the TranslatorDSC functioning is showed in the sequence diagram presented in Fig. 4.

When a new object of the "TranslatorDSC" class is instanced, it instances an object of the "GAD" class (described in section IV, item C). After the ".dsc" file reading, the "TranslatorDSC" object instances an object of the "Node" class. This object is associated with the "GAD" object previously created. When the evidences of a node are found, these evidences and their associated probability values become part of the node characteristics.

B. Node Class

The Node class is responsible for generating the data structures and methods for creating and maintaining nodes in memory. A graphical description is shown in Fig. 5.
The most useful method for the "TranslatorDSC" class is "receiveSon". The other methods can be used for the programmer in order to increase system functionality. For example, the "objectInstanced.listSons( )" call returns an object "ChoiceGroup", which may be immediately associated to a form for display on the device screen. This allows the choice of an evidence contained in the node. Similarly, the method "printEvidences" can be used to display evidences.

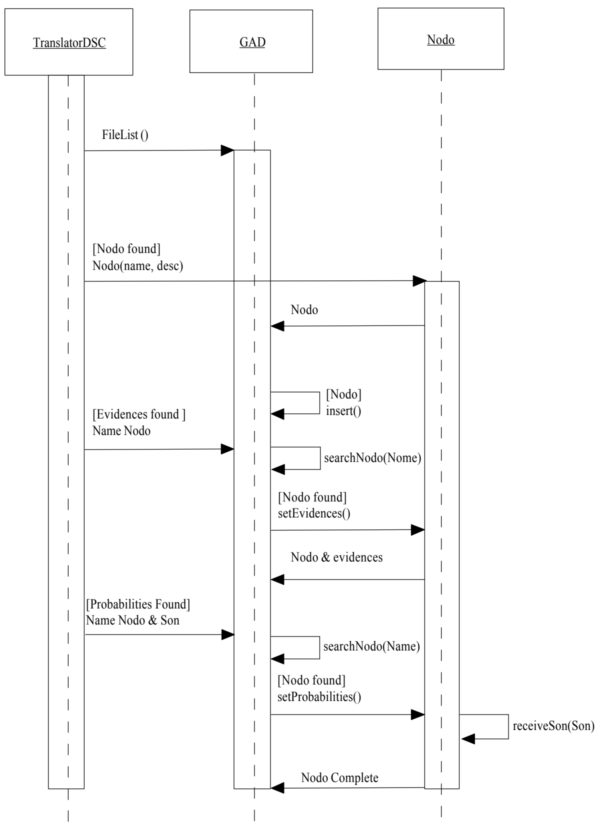

Fig. 4. Sequence Diagram of TranslatorDSC.

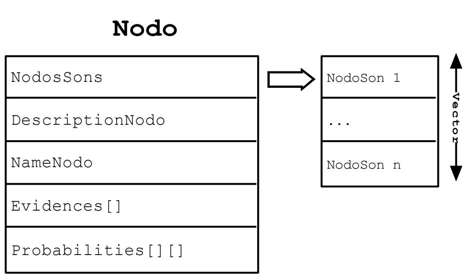

Fig. 5. Data Structure of the Node class.

C. The GAD Class

The GAD class is responsible for maintaining and creating the data structure that represents a Directed Acyclic Graph (DAG). The GAD class stores references to objects of the "Node" type, and creates a collection of nodes that compose the BN. This class supports the execution of TranslatorDSC. To reach such objective, the "Insert" and "searchNode" methods allow nodes referring to a specific GAD to be inserted, and also searched, in order to guarantee the correct localization of pointers (object references) during algorithm execution.

D. The BSearch Class

The class that implements the search in the graph is defined such that it can operate on BNs that represent a set of questions and answers on a certain subject. In this work, the BN used as an example represents the set of questions referring to patient health, which is contained in the symptomatological interrogatory. The BSearch class has only one method, called Search. It is responsible for starting the questionnaire. BSearch class assumes that an instance of the "TranslatorDSC" already exists.

Figure 6 illustrates, as an example, that this system software can be used in other application domains. This example presents a brief questionnaire about hardware issues in a personal computer.

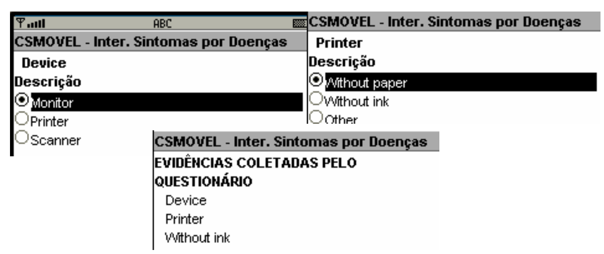

Fig. 6. Using FAGE in a different application domain.

E. The Prototype

Prototypes of a desktop and a mobile device versions were developed (Fig. 7). The use of the symptomatological interrogatory as a BN that represents a questionnaire confirms the usefulness of the software artifacts previously described in [14].

Fig. 7. Collected evidences.

The algorithms are capable of supplying a list of all the evidences that were collected during the execution of the symptomatological interrogatory (Fig. 7). This allows, for example, a set of collected evidences (answers) to be sent to the patient's physician, or to an external system that handles this information.

V. CONCLUSION

The defined classes allow a structured approach for the construction of a computing system that can deliver a questionnaire on a certain subject. Using the proposed framework, it is possible to separate the semantics from the questionnaires, by using BNs in the implementation of the execution of the questions. Thus, it is possible to use the proposed framework as a tool for the construction of this kind of software.

As a result, a much smaller number of code lines are needed. If a conventional method, such as the method based on selection using IF..THEN..ELSE, was used, there would be a nesting of commands that would lead to an equal number of questions. In contrast, the "BSearch" class, has only thirteen selection commands.

For the user of mobile systems, the algorithm guarantees that it is not necessary to have a complete vision of the universe of questions on the subject under evaluation. This reduces the evaluation time for the multiple options.

As future work, we propose the use of a new technique based on graph path, using heuristic methods for the selection of a more optimized set of questions, which can accelerate the response time perceived by the system's users.

TranslatorDSC should be modified in such a way that it becomes a generic translator, which is capable of working with BNs that possess more than a parent node for each child node. This will allow the use of BNs that do not describe only the models of structured questionnaires in a hierarchic way.

Also, we plan to add support for file formats other than ".dsc", such as the XML (eXtensible Markup Language) format.

The project is currently registered in SourceForge.net, and it is called F.A.G.E (Framework for Automatic Gathering of Evidences in Mobile Devices). The software is available under the terms of the GPL/GNU license [15].

REFERENCES

[1] M. Billinghurst, Wearable Appliances; "The Future of Wearable Computing", *Human Interface Technology Laboratory*, University of Washington, 2002.

[2] H. Carvalho, A. Murphy, and C. Coelho, "A General Data Fusion Architecture". *Proceedings of the 6th International Conference on Information Fusion (Fusion 2003),* June 2003.

[3] R. Devaul, M. Sung, J. Gips, and A. Pentland "Sandy". "MIThril 2003: Applications and Architecture", *Media Laboratory*, Massachusetts Institute of Technology, 2003.

[4] S. Mann, "Wearable Computing: Toward Humanistic Intellingence", *IEEE Intelligent Systems*, University of Toronto, 2001.

[5] J. Pearl, and S. Russel, "Bayesian Networks", Technical Report, R-277, *Handbook of Brain Theory and Neural Networks*, MIT Press, November 2000.

[6] A. L. Carneiro, and W. T. Silva, "Introduction to Bayesian Networks" (in portuguese), Research Report, Department of Computer Science, University of Brasilia - 09/99.

[7] M. Dunlop, and S. Brewster, "The Challenge of Mobile Devices for Human Computer Interaction". Available: www.cis.strath.ac.uk/~mdd/mobilehci01/

[8] S. J. Russell, and P. Norvig, *Artificial Intelligence A Modern Approach*, Ed. Prentice Hall, Englewood Cliffs, New Jersey.

[9] T. Cormem, C. Leiserson, R. Rivest, and C. Stein, *Algoritmos - Teoria e Prática*, (Translation of 2nd American Ed.),2002, 936pgs

[10] A. L. V. Forbellone, and H. F. Eberspächer, *Lógica de Programação - A Construção de Algoritmos e Estruturas de Dados*, Ed. Makron Books, 1998.

[11] Microsoft Research, http://research.microsoft.com/adapt/MSBNx/, November 2004.

[12] C. C. Porto, *Medical Semiology* (in portuguese), Editora Guanabara Koogan, Third edition, 2001.

[13] M. G. A. T. Barbosa, Jr. G. I. Sene, S. S. L. Castro, J. P. H. Branisso, C. É. Figueredo, S. H. Carvalho, F. A. Rocha, and A. O. F. Nascimento, "Personal Móbile System for Monitoring Human Health: Algorithm for Intelligent Capture Symptoms" (in portugues), *Brazilian Conference on Heath Informatics*, 2004.

[14] A. B. José, M. G. A. T. Barbosa, Jr. G. I. Sene, A. F. Rocha, S. H. Carvalho, and A. O. F. Nascimento, "Implementation of Systematica Symptom Review Using Mobile Systems" (im Portuguese), *Workshop on Medical Informatics*, ISBN 85-7430-515-4, 2005.

[15] Framework for Automatic Gathering of Evidences in Mobile Devices, http://sourceforge.net/projects/fage/, April 2005.