A COLLABORATIVE TOOL TO STUDY PATHS AND CYCLES IN DIGRAPHS[1]

Francisca Losavio, Oscar Ordaz, Jesús Yépez
Centro de Ingeniería de Software Y Sistemas ISYS,
Facultad de Ciencias, Universidad Central de Venezuela,
Apdo. 47567, Los Chaguáramos 1041-A, Caracas, Venezuela
{flosavio, oordaz, jyepez}@isys.ciens.ucv.ve

Luciano García
Departamento de Ciencia de la Computación
Universidad de La Habana
San Lázaro y L, La Habana, Cuba
luciano@matcom.uh.cu

Abstract

The main goal of this work is to propose a collaborative tool COL-AGORA (COLlAborative diGraph cOnjecture Research Assistant), for improving the graphist work in the conjecturing process. The usual practice in this domain is to formulate conjectures and try to prove, refute or leave them as open problems, on the basis of an existing knowledge. This knowledge is shared by the scientific community in graph theory and it is constituted by Boolean and integer invariants, relations between invariants, examples of graph or digraph families, theorems, conjectures, definitions and notations. Moreover, the knowledge, that must be easy to consult, is updated and augmented constantly by new results obtained by the research groups distributed all over the world. The available collaborative tools are in general limited to specialized text editors and e-mail facility. COL-AGORA will benefit from network technology and a client-server architecture, offering private knowledge repositories that can be easily updated and consulted by a specific research group. Moreover, clients may consult repositories of other scientific groups in the domain through a Web browser using remote networking facility, increasing the potentiality of conjecture evaluation on a specialized server. The reasoning system is based on intelligent agents, charged of the repository administration and the conjecture evaluation process.

Keywords: collaborative tool, reasoning system, conjecturing process, digraph conjecture, intelligent agent, distributed architecture, Web application.

INTRODUCTION

Our main problem is which computational system must be designed in order that research groups, teachers and students in graph and combinatorial theory can solve problems in these scientific domains in a natural and collaborative way. Actually the usual practice in this domain is to formulate conjectures and try to prove, refute or leave them as open problems, on the basis of an existing knowledge. This knowledge is shared by the scientific community in graph theory and it is constituted by theorems, Boolean and integer invariants, relations between invariants, examples of graph or digraph families, theorems, definitions and notations that are commonly used in combinatorial problem solving. *Invariants* are integer or Boolean values that are preserved under isomorphism. The number of nodes and the number of arcs are obvious integer invariants; they are related by $arcs \leq nodes\ (nodes\ -1)$. Some Boolean invariants are: *hamiltonian* and *traceable*; they are related by *hamiltonian* $\Rightarrow$ *traceable*. This

[1] This research has been supported by the Red GRACIAS, Economical European Community Alfa Program

knowledge, that must be easy to consult, is updated and augmented constantly by new results obtained by research groups. In practice, the knowledge is scattered in papers, books, scratches, drawings and the researchers must manipulate lots of non-structured and heterogeneous information.

An interesting attempt to structure and organize specialized knowledge on paths and cycles in digraphs has been made recently [5], and it is the kernel on which the AGORA computational tool [23] has been built. This tool assists the graphist in the reasoning process. The graphist reasoning is passionating, but at the same time it may easily conduce to a long and sterile work, if tools and methods are not available. The logical kernel of the tool is the reasoning system, based on terminological logic and the constraint logic programming paradigm [13], [17]. The main reasoning capabilities are performed by a subsystem composed fundamentally by a *knowledge base* constituted by selected examples of digraphs, relations between invariants and theorems in digraph theory, specifically on paths and circuits [5] and a *subsumption algorithm* [10], [11] for consistency, proving and refutation. An important additional feature of the system is to provide an explanation concerning the conjecture deduction process, through a powerful graphical user interface [20], [23]. However this system does not offers facility for sharing or updating the knowledge among different groups.

Several other efforts have been made to build computational systems to aid graphists. From our perspective, the following systems deserve to be mentioned: the INGRID software system [3], [6] has been designed in order to bound all invariants once the ranges of selected invariants are specified. The system relies upon a collection of relations between invariants involving 36 graph invariants and a set of graphs. The GRAFFITI system [8] generates a conjecture on a negative way, i.e. by verifying that the system can not afford a graph as a counterexample to the conjecture and deciding after that whether the conjecture is or not interesting for being the object of further research. Later work on GRAFFITI [9] has been motivate by designing more powerful heuristics for telling whether the conjectures made by the system are of interest.

Our goal is to provide the basic elements required to design a new tool, COL-AGORA (COLlAborative diGraph cOnjecture Research Assistant) that can benefit from the recent technology on heterogeneous and distributed systems, facilitating collaborative work. We think that this feature is extremely important to model the "real world" of the graphist community practical work. The natural way to work for the researchers distributed all over the world, is actually collaborative, exchanging, consulting and updating results in graph theory. By now, the graphist community uses mostly TEX[2] for writing and register results and e-mail as the main collaborative tools. The main goals of computer-supported cooperative work [14], which includes workflow automation, GroupWare and collaborative work, is to increase productivity in an organization. We feel that a specific domain computational interoperable tool that will support this kind of practice could be a contribution to increase the research production in the field.

---

[2] TEX is a Trademark of the America Mathematical Society

On one hand our collaborative tool is inspired in the existing AGORA system [23] for modeling the graphist reasoning, taking advantage from its initial knowledge base. On the other hand, it will benefit from a distributed architecture on heterogeneous platforms, based on a multiagent model, which is described in this work.

Besides this introduction and the conclusion, this paper is structured in three main sections: the first one containing the tool initial basic knowledge, the second one discussing the experience obtained in the construction of the present AGORA tool, and finally the third section, devoted to present the new collaborative tool, COL-AGORA.

THE BASIC INITIAL KNOWLEDGE

The problem we want to solve, partially or totally or give assistance to its solution, is basically the generation and validation (consistency checking) of digraph families. These sets of graphs are defined in terms of conjunction of constraints establishing values of certain invariants and arithmetic relations between invariants. It is a non-decidable complex problem, from a logical point of view, that can de solved only in certain cases. The main subproblems involved are instantiation of a family, digraph random generation and generation and validation of digraph conjectures.

In graph theory, a *conjecture* is an implicational statement of the form $H \Rightarrow T$, whose antecedent called *(H)*ypothesis describes the conjunction of Boolean invariants and/or relation between integer invariants that a digraph family may fulfill in order to satisfy another conjunction of invariant relations described by the consequent, called *(T)*hesis. A very important terminological relation between set descriptions is the *subsumption relation*. We say that a set description *H is subsumed* by a set description *T*, if the family of digraphs which satisfies *H*, denoted by H^I is a subset of the family of digraphs which satisfies *T* denoted by T^I. Notice that to prove $H \Rightarrow T$, is equivalently to prove that H^I is subsumed by $T^{I.}$

The conjecturing process in graph theory means all the activities related to the study of a conjecture: solve or prove, reject or leave it open, on the basis of the existing knowledge (relations between invariants, reasoning strategies or methods, existing theorems, open conjectures, examples of digraphs families). Notice that the graphist must have at hand the definition of the terms used, such as hamiltonian and traceable. [5] Provides also these definitions and the computational tool must make them available to the researcher. The basic knowledge can be found in [5]. Most results are extracted from recent surveys, like [2], [26], [27] and it is constituted by the following information:

31 integer invariants. Among these we have:

`nodes`: number of nodes of a digraph
`arcs`: number of arcs of a digraph
`alpha0`: maximum size of a set of nodes inducing no arc

23 boolean invariants. For example:

`hamiltonian`: the value true of this variable means that the digraph contains a Hamiltonian circuit

`traceable`: the value true of this variable means that the digraph contains a Hamiltonian path

`symmetric`: the value true of this variable means that the digraph is symmetric

3 integer invariants for bipartite digraphs:

`bialpha0`: the maximum size of a balanced set of nodes without an induced arc

`bialpha1`: the maximum size of a balanced set of nodes without circuits

`bialpha2`: the maximum size of a balanced set of nodes without circuits of length 2

10 boolean invariants for bipartite digraphs, such as:

`bicomplete`: this variable is true if the digraph is complete bipartite

`X-s-circuit`: this variable is true if every vertex in `X` is on a circuit of length `s`

119 relations between invariants. There are two types of relations between invariants in called (absolute) invariant relations which are sets of basic constraints and (conditional) invariant relations which are conditional constraints. The following are examples of these relations. The numbers in brackets means the bibliographic reference number of the paper where the relation is put into evidence:

R_{67} `1 alpha0 alpha1 alpha2.`

R_{1} `hamiltonian` ⇒ `1-connected.`

R_{61} `k-connected and k 1` ⇒ `(k-1)connected.`

R_{55} `k-connected` ⇒ `mindegree 2k.`

R_{56} `k-connected` ⇒ `mindepositive k.`

R_{64} `k-connected` ⇒ `minimum k.`

R_{37} `minimum nodes/2` ⇒ `1-connected [69].`

R_{71} `symmetric` ⇒ `alpha0=alpha1=alpha2 [51].`

R_{58} `k-connected` ⇒ `nodes k+1.`

R_{63} `woodall nodes` ⇒ `1-connected.`

R_{84} `meyniel 2woodall.`

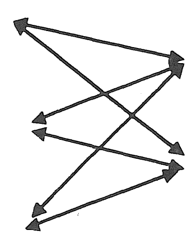
64 example of digraph families descriptions; each example is specified as a list of invariants values that the digraph or digraph family must satisfy. For example, we have:

D10 Integer invariants

```
meyniel = 8
mindegree = 4
minimum = 2
nodes = 5
woodall = 4
```

Boolean invariants

```
hamiltonian = false
1-connected = true
```



Comment: This is an instance of the family of symmetric bipartite complete graphs, with stable components of sizes differing by 1. They show that Theorem 55 [10,1.1.2], 56 [10,1.1.3], and 57 [10,1.1.4] are the best possible, since they are not Hamiltonian and `mindegree = nodes - 1, meyniel = 2nodes - 2 and woodall = nodes - 1.`

Notice that the number of the theorems listed in brackets, where the example is best possible is shown. Moreover, the bibliographical reference number of the paper, and the particular section where the example is used is also shown within the brackets. A semicolon separates different reference numbers.

136 Theorems and open conjectures, which are represented as conditional constraints; each theorem is specified as $H \Rightarrow T$, where *H and T* are descriptions of digraph families. *T* can also be a disjunction of digraph family's descriptions. Some examples of theorems and conjectures are:

```
56. Theorem [10,1.1.3;69]:
woodall   nodes ⇒ hamiltonian.
Best result: see D11 and D10

61. Theorem [47;73, 1.1.12]:
k-connected and (alpha2 a)  and k 2^a(a+2)! ⇒ hamiltonian.

62. Conjecture [10,1.4.5;45]:
antisymmetric and r-diregular and (r 3) and (nodes 4r+1) ⇒
hamiltonian
```

THE AGORA EXPERIENCE

The AGORA system [23] (see Figure 1) is constituted by two main components, the *AGORA Interface System* (AUI) and the *AGORA Reasoning System* (ARS). AUI captures the user inputs and adequately deliver them to ARS. AUI is constituted by four main modules: - *Conjecture Editor*, - *Conjecture Browser* (CB), - *Knowledge Base Browser* (KBB) and - *Help System*. (HS) The interface layer between both systems is constituted by two temporary text files, one for storing the conjecture in Prolog form, and the other for storing the results of the ARS evaluation.

The *Conjecture Editor* (CE) is responsible for the AGORA main functionalities. Its graphical interface allows the user to enter a conjecture, expressed in *LEC (Conjecture Specification Language)* [19]; this expression is submitted to a semantic and syntactic analysis before being translated into an internal form, which is understood and evaluated by ARS. For example the LEC expression:

woodall ≥ nodes ==> hamiltonian-connected.

Represent a conjecture written in LEC. A text file, produced as a result of the translation of the conjecture in a PROLOG internal form, is read by ARS to evaluate the conjecture, under the responsibility of the *Deductive Mechanism*. The results of the evaluation are stored by the

Explanation module in another text file. This file is used by the Conjecture Editor for showing these results to the user. Both temporary files maintain the independence of the reasoning system from the graphical user interface.

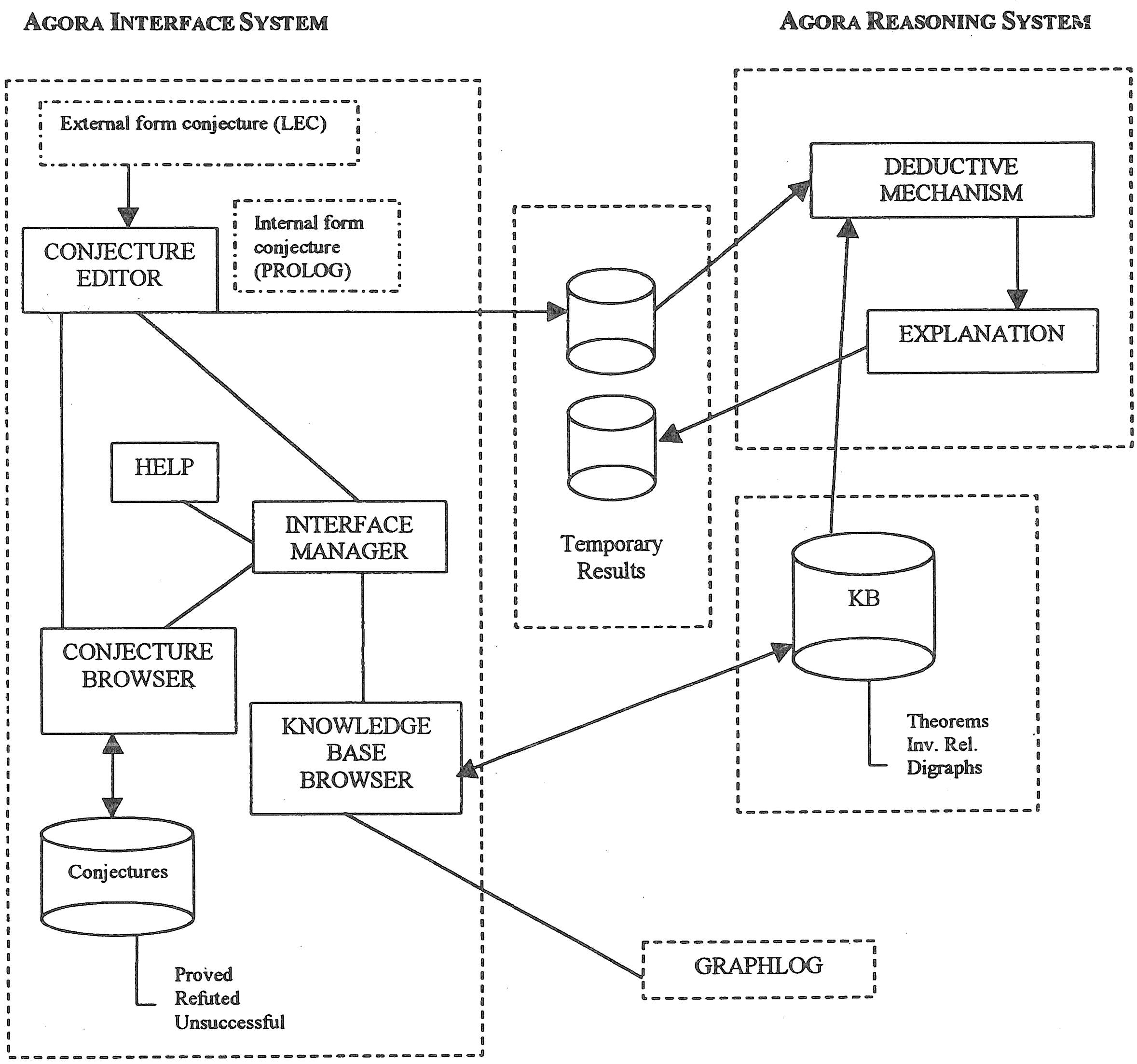


Figure 1. Architecture of AGORA

The CB is used to store and show the persistent information on conjectures and the results of the evaluation. Information such as date of evaluation, results and explanation, is registered for each conjecture. The browser can handle *Proved, Refuted* and *Failed Conjectures* through a powerful graphical interface

The KBB allows the *AGORA Knowledge Base* (AKB) editing, located in ARS. Through this browser, specialized information on theorems, invariant relations and digraphs can be stored or retrieved. Records are used to represent AKB data. New AKB records can be added and existing ones can be modified or deleted. An AKB record contains the following main fields: *identifier, references, comments, text*. The identifier is used to distinguish each record. References are constituted by bibliographical information on theorems and conjectures, and pictures of special digraphs, which are accessed through hyperlinks by the *GRAPHLOG* external module [7][3]. Comments are textual complementary information that can be introduced by the user, such as a date or 'best result'. The text field contains data on the theorem, invariant relation or digraph example. This information is written in LEC and, in case of modification, it is syntactically and semantically verified before being translated to the PROLOG internal form understood by ARS.

The HS is a standard on-line help tool that can be accessed from any of the AUI modules.

Finally, the *Interface Manager* controls the main modules of the AGORA Interface System

ARS is constituted by three modules:

1. The AKB [5] is used to structure and organize the information used by the AKB, which is partitioned into three interacting submodules Invariant Relations, Examples and Theorems. It contains:

2. The deductive mechanism based on the Subsumption Algorithm [11].
It constitutes the ARS inference machine and accomplishes the refutation and proof processes of the conjecture to be evaluated. For evaluation purpose, this mechanism uses also the information present in AKB.

3. The explanation subsystem.
Once a conjecture is proved or refuted, explanations are given to the user on the evaluation process followed by ARS, justifying logically the solution obtained. The main feature of this module is to use the information produced by the deductive mechanism while proving or refuting a conjecture, in order to construct the explanation. This information is stored in a temporary text file and is retrieved by the CE for displaying the explanation to the specialist.

THE COLLABORATIVE AGORA

One of the major drawbacks of the AGORA architecture is that it is monouser. The present technological advances of fast remote communication will be used in order to facilitate the collaborative work. COL-AGORA will present a distributed architecture, with a client-server organization that will be running on heterogeneous platforms, thus favoring the different

[3] This subsytem, part of the GReAt environment [22], can run also as a stand-alone system and implements a hypermedia library and a digraph album, on graph theory specialized knowledge.

research groups all over the world. In order to guarantee platform independence, the Java language [18] will be used to implement COL-AGORA.

We requires that the different users of the system can work interactively, independently or in cooperation, making use of local or remote resources, distributed on the network, cooperating in the problem solution submitted to the system. So, a multiagent system is proposed, composed by human and artificial agents. From the computational point of view, artificial intelligent agents (AIA) are introduced for assisting and cooperating with human agents in certain specialized tasks.
The graphical user interface (GUI) will be based on HTML Web pages. Java applets will be retrieved from Web servers for satisfying the client's requests and are responsible for the interaction. In this way, the system is independent form the platform and can be widely distributed among the research groups working with different platforms.

Requirements for a Distributed Architecture

The architecture for (COL-AGORA) is shown in Figure 2.

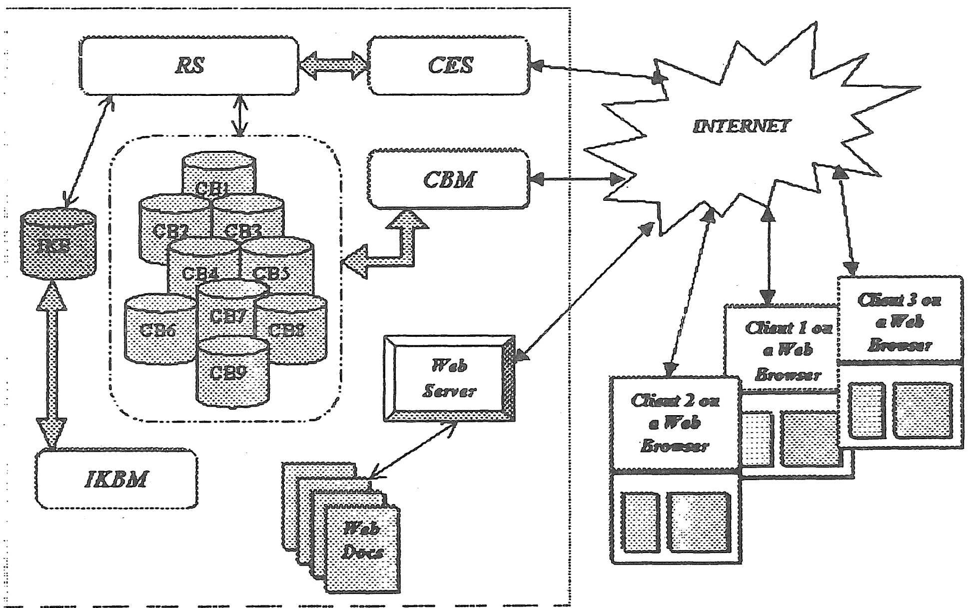


Figure 2. COL-AGORA architecture

A user, working on a client workstation on a local network, must connect to a browser and to the AGORA Web site through an Internet access. The first documents that are retrieved are HTML pages, which are interpreted, generating new requests to the Web server for retrieving the applets providing the system human-interaction (represented by small gray squares on the client). The AGORA GUI must be interoperable, intuitive and easy to understand for specialists in the domain, i.e. well adapted to the graphist problem solving tasks, facilitating also the administration of the user "private" knowledge base or conjecture box (CB).

The applets will be used to initiate a session with the Conjecture Evaluation Server (CES) or with the users Conjecture Box Manager (CBM), which behaves according to a distributed object management schema. CES receive a conjecture request from a remote user and submits it to the Reasoning Subsystem (RS), which consults the Initial Knowledge Base (IKB) and the remote users CBs. Moreover, CBM is used also to administrate the CBs. It is obvious that the object manager must provide synchronization capabilities. It may happen that a user, for example, is updating its CB while others are evaluating conjectures on the same CB. The IKB manager (IKBM) is a Java application running on the COL-AGORA Web site. It allows a special user, the administrator, to maintain i.e. add, delete or change theorems, Boolean and integer invariants, invariants relations and digraph examples. No external users can access to IKB. Research groups have the right to modify only their private CBs, but they are not allowed to modify IKB. Nevertheless, IKB can be enriched by new knowledge. A researcher can send a message to the IKB administrator with the required modification, by e-mail for example; the administrator must submit the request to a specialist board, which is the only authority that can authorize the modification, after a careful study. After having received authorization, the administrator updates IKB. When IKB is being updated, a message is sent to inform the users that COL-AGORA cannot evaluate conjectures using IKB, and that only the private CBs can be used.

Strategies should be supported to possibly reduce the time of the evaluation. A client should be able to select between evaluate a conjecture using only the IKB, or a combination with some or all the CBs, including its own. In this case, a mechanism should be provided to make visible the other CBs to the evaluator. Notice that in a collaborative environment, a group may require that the visibility of the CB will be restricted only to the group.

The agent based reasoning subsystem

A multiagent approach will be used to model the reasoning system. An agent is an entity capable of perceiving (by sensors) and take actions (by efectors) towards the obtention of a predefined goal [26]. An agent behavior is based on its own experience and on the integrated knowledge used in constructing the agent for the environment in which it will operate. A system is considered autonomous in the measure that its behavior is defined by its own experience. An agent program is a function implementing the agents mapping from perceptions into actions. The system architecture allows the program to capture the perceptions obtained by sensors, execute and provide the selected actions to the efectors in the measures that they are being generated. Figure 3 shows the agent based reasoning subsystem. A client formulates a request for a conjecture evaluation. For each new request, different instances of the AIA using the IKB and/or the CBs are generated. Figure 3 shows an example with one request, where four agents are generated, a coordinator and three subordinate agents. One of the subordinate agents is charged to work with the IKB, the other two are using the users private CBs. The coordinator gets the results obtained by its subordinates, organizes and enroutes them to CES, which returns them to the client, to satisfy the requests. CES creates also the AIA instances and distributes the tasks. A broker or a mediator architectural pattern [4] can model the architecture for CES, for example.

The Graphical User Interface

COL-AGORA will run on the clients as an Internet application, through a browser, displaying Web pages; applets will be used within a Web page to answer the queries. The GUI architecture will be also based on a multiagent approach. The agents will be interface agents that are "simple" AIA. Since the Web servers are accessed directly while executing the applets, the GUI architecture cannot be modeled with the PAC (Presentation – Abstraction - Control) [1] multiagent model approach, as was the case for the existing AGORA. PAC interface agents communicate only through their controls; so, a PAC interface agent could not access directly the Web server, in case of an applet execution [4]. We propose that the user interaction should be modeled instead MVC (Model-View-Controller) [12] interface agents with no communication constraints. The resulting architecture will be more coupled, but it will gain in efficiency, benefiting from the direct communication schema with the Web server. An MVC agent could model each COL-AGORA subsystem. For example, for the Conjecture Evaluation Subsystem we could have: -View-controller: the components displaying the conjecture representation in LEC. - Model: the internal representation of the conjecture.

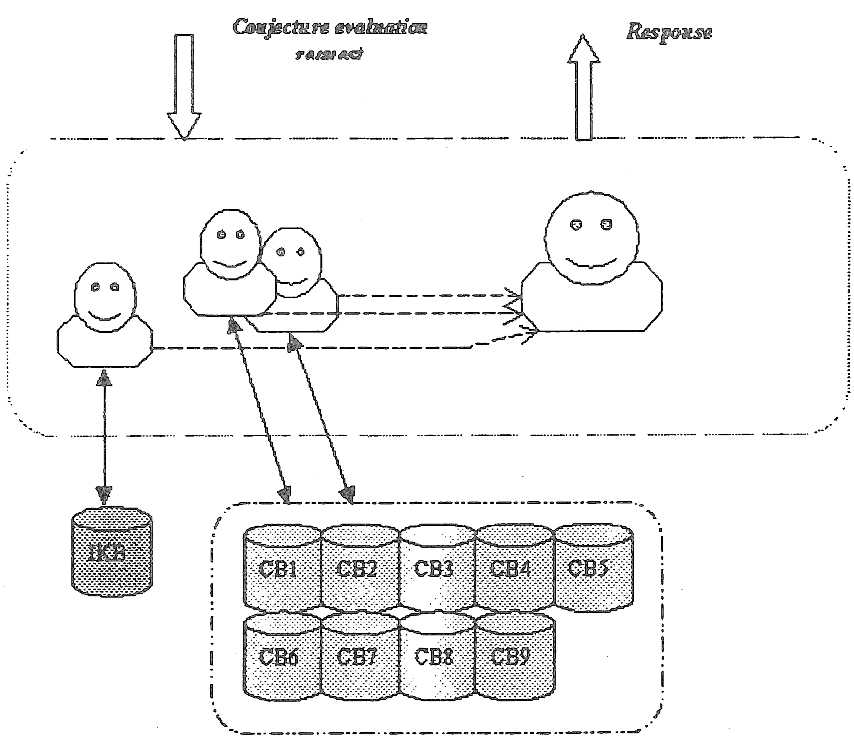


Figure 3. Example of a conjecture evaluation

CONCLUSION

In this paper we have presented an architecture for the distributed COL-AGORA collaborative tool, based on the monouser AGORA experience, to give support to the work of the scientific community in graph theory, located all over the world. COL-AGORA increases the AGORA facility, providing a distributed application with an easy to use and learn user interface conceived as a Web application. The benefits are multiple: a total independence from the platform, since any user can reach COL-AGORA from its Web browser, capable of applets execution. Moreover, the conjecture evaluation process is extended, since it uses the private CBs and it is not limited to IKB.

On the other hand, the AIA's involved in RS must be precised, studying the tasks complexity that have to be distributed among different agents. Another aspect that must be taken into account is the reuse of the AGORA RS, using a Prolog version Java compatible, as for example BinProlog[4].

REFERENCES

[1] L. Bass, J. Coutaz . Developing Software for the User Interface, Addison-Wesley Publishing Company, Reading, Mass., 1991
[2] Bermond J-C, Thomassen C. "Cycles in digraphs - a survey" J. Graph Theory 5 (1981) 1-43
[3] R.C. Brigham, R. D. Dutton. A Compilation of Relations between Graph Invariants. Networks, 15 (1985) 73-107.
[4] Correa A., Gálvez C., Losavio F. Multiagent Model Based Architecture for Internet Application Submitted to ISAS'99.
[5] C. Delorme, O. Ordaz, D. Quiroz. Tools for Studying Path and Cycles in Digraphs. Networks 31 125-148, 1998.
[6] R. D. Dutton, R. C. Brigham. INGRID: A Software Tool for Extremal, Graph Theory Research. In Congressus Numerantium, 39 (1983) 337-352.
[7] L. Escalona. GRAPHLOG: Una Enciclopedia para el Ambiente GReAt. Tesis de Licenciatura, Escuela de Computación, Universidad Central de Venezuela, Octubre 1996.
[8] S. Fajtlowicz. On Conjectures of Graffiti. II. In Congressus Numerantium 60 (1987)187-197.
[9] S. Fajtlowicz. On Conjectures of Graffiti, IV. In Congressus Numerantium 70 (1990) 231-240.
[10] L. Garcia. A CLP-Approach to Graph Conjecture Problems, Technical Report, AIG-Universidad de la Habana, 1995.
[11] L. Garcia, E. Lastres, F. Losavio, O. Ordaz. A reasoning system for digraph conjecture solving. Proceedings of the International Conference on Information Systems Analysis and Synthesis ISAS'96. Orlando, USA 1996.
[12] A. Goldberg. Smalltalk-80. The Interactive Programming Environment, Addison-Wesley 1984.
[13] J. Heinsohn, D. Kudenko, B. Nebel, H.-J. Profitlich. An empirical analysis of terminological representation systems. In Artificial Intelligence, 68 (1994) 367-397.
[14] Graham I. "Migrating to ObjectTechnology", Addison Wesley, 1995
[15] Jackson B., Ordaz O. On some conjectures on Hamiltonicity in digraphs Acta Cientiifica Venezolana 38 (1987) 629-632
[16] Jackson B., Ordaz O. Chvatal-Erdos conditions for paths and cycles in graphs and digraphs. A survey. Discrete Nath. 84 (1990) 241-254
[17] J. Jaffar, M. J. Maher. Costraint Logic Programming: A Survey. In Journal of Logic Programming
[18] Javasoft.com, "Java Platform Documentation", http://www.javasoft.com/docs/index.html

[4] BinProlog is a registered trademark of BinNet Corp.

[19] E. Lastres. Lenguaje de Especificación de Conjeturas en AGORA, Technical Report, AIG-Universidad de la Habana, October 1996.
[20] F. Losavio, F. Marchena, A. Matteo. Specifications of GUI Frameworks, In Proceedings of the XXIII Conferencia Latinoamericana de Informática Panel'97. 10-13 Noviembre, Valparaiso, Chile, (1997) 791-802.
[21] F. Losavio, A. Matteo, J. Yépez. MWW: Un envoltorio para los Widgets de Motif, XLVII Convención de ASOVAC, Valencia, Venezuela (1997) 232-233.
[22] F. Losavio, A. Matteo, O. Ordaz, O. Meza, W. Gontier. Object-oriented approach and PAC model: Design of the GReAT(Graph Researcher Assistant) environment. In Proceedings of the XX Conferencia Latinoamerican de Informatica, Panel'94. Mexico (1994) 433-443.
[23] F. Losavio, O. Ordaz, J. Yépez, Quiroz D., García L., Padrón O. A Tool for Conjecturing in Graph Theory, In Proceedings of the XXIV Conferencia Latinoamerican de Informatica, CLEI'98, Quito (1998) 583-594.
[24] A. Márquez, S. Contreras, O. Ordaz, F. Losavio, W. Gontier. A knowledege-based assistant for formulating digraph conjectures. In Proceedings of the IV Congreso Iberoamericano de Inteligencia Artificial IBERAMIA'94, Mac Graw Hill ed. Caracas. Venezuela (1994) 30-43.
[25] J. Outerhout. Tcl/TK. Tcl: Summary of Tcl language syntax. TK: The TK Toolkit, Overview and Porting Guide. 1995.
[26] Russel S., Norvig P. Inteligencia Artificial. Un Enfoque Moderno, Prentice Hall.
[27] Zhang K.M., Song Z.M. Cycles in digraphs a survey - Preprint.