Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

EpiTalk
A Platform for Epiphyte Advisor Systems Dedicated to Both Individual and Collaborative Learning

Sylvain Giroux[1], Gilbert Paquette[1], François Pachet[2] and Jean Girard[1]

[1]LICEF/UER Science et Technologie, Télé-Université, 1001 Sherbrooke est, Montréal, P.Q., H2L 4L5 Canada, sgiroux@teluq.uquebec.ca

[2]LAFORIA, Université Paris VI, 4, place Jussieu, 75252 Paris Cedex 05, France, pachet@laforia.ibp.fr

Abstract. In distance learning environments, advisors can start off again a stalled learning process. EpiTalk is a platform enabling to develop advisors for individual and collaborative learning. EpiTalk advisors rely on agents, organizations and hierarchical graphs. Hierarchical graphs describe advisors. At run-time, agents provide autonomy to each advisor's components, while organizations link and co-ordinate agents. Since advisors are grafted onto existing arbitrary information systems, we say that they are *epiphyte*.

1 Introduction

To support distance learning, Télé-université supplies learners with information systems either needed for tasks, or managing on-going learning processes [1]. Besides there are human tutors [13]. But tutors are not *on-line*. So when a learner gets into trouble, accurate help may not be at hand. Advisor information systems can provide a just-in-time and ever available individualized first-line help. As communication technologies are integrated into modern distance learning environment [26], collaborative learning activities can take place at distance [3] [13] [2]. So advising isolated activities is not sufficient, distributed advice to groups must also be provided.

The EpiTalk project [27, 29] addresses the complex issue of advising individuals and groups using the existing home-made or commercial applications. Our solution is to consider advisors as *epiphytes*[1] information systems, that is as information systems growing onto other information systems, their *hosts*, without perturbing them whatsoever [8]. Exploiting this metaphor, we devised and implemented EpiTalk, a platform dedicated to epiphyte advisor information systems (EAIS). EpiTalk provides the means for describing EAIS: pattern/plan recognition algorithm based on information spied from host information systems [20], distributed student modeling [19] [15] [16], and advice contents and strategies [21]. At that time EpiTalk generates a multi-agent EAIS using descriptions of hosts, of content and task structures, and of pieces of advice [10]. Next EpiTalk grafts the EAIS onto learning environments [22]. Finally EpiTalk provides tools to monitor and debug EAIS [9].

This paper presents EpiTalk's main features and commitments. First we present our goals and constraints (§2-3). Then we show how EpiTalk enables to design, launch and monitor EAIS dedicated to individual (§4-7) and collaborative learning (§8). Next we sketch advice strategies (§9). Finally current works are outlined (§10).

[1]An epiphyte plant grows on the surfaces of other plants, its host, without damaging it. It lives on its own, but needs other plants. Ivy, and most orchids are typical examples.

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

2 A Generic Architecture for Advisor Systems

2.1 EAIS for Individualized and Collaborative Learning

Several applications can run concurrently on today's multi-window operating systems. Networks enable distributed collaborative work. Therefrom a learner may pursue as many goals as there are on-going individual and collaborative learning processes intertwined on its workstation. Thus advisors on distinct on-going processes must be distinct while still be able to cooperate, and prevent inconsistent pieces of advice. Moreover advisors must express advice that span across different levels of abstraction [21] and pieces of advice related to either an application —how to perform a task more effectively with the application? [14]—, a production —how to improve the design of a course [28]?—, a process —what is the next step to perform? [21] [23].

EpiTalk solutions to these issues bear on agents [4], organizational reflection [18], and knowledge graphs [25]. Agents are used to disentangle threads of concurrent learning processes. Each EAIS is an agent which focuses on one object —process, production, or application. Agents also bring concurrency between EAIS, and within an EAIS. Organizational reflection permits to easily shift viewpoints on agents and groups: "an agent can be viewed as an organization of agents" and vice-versa [5]. These shifts allow to treat uniformly both individual and collaborative learning. The same mechanisms for communication, cooperation and negotiation apply uniformly inside and outside EAIS. The knowledge graph expresses the designer's viewpoint on the object of advice. The part-of relations are used as a *template* to generate the multi-agent EAIS. Each node provides information used to generate and link EAIS agents, to graft the EAIS on the host, and to adapt the EAIS to the host evolution.

2.2 Capitalizing on Existing Applications

Most advisor systems are designed and developed either as self-contained systems, or as components of top-down designed, integrated learning environments [33]. As a result, the design of an advisor is buried into the design of large systems, which makes transposing the advisor to different contexts a difficult task. Besides it precludes to build learning environment by integrating existing home-made or commercial applications. An advisor too closely coupled to a platform, an operating system or an application version must be thrown away whenever it changes, even if in most cases, advice relative to a task would not. Our approach to advisor systems has been to consider them not as parts of a self-contained system, but as independent extensions of applications, not necessarily designed to be parts of a learning environment. EpiTalk uses organizational reflection to graft the advisor on hosts. The hierarchical structure of knowledge graphs is then exploited to complete the separation of advisors from applications, providing a complete independence from a sufficiently abstract level.

2.3 Reusable Advisors

Advisors are one of the cornerstone of the virtual campus designed at Télé-université [30]. Since programming resources are limited, as many tasks as possible should be automated, and advisor designers must be autonomous. With respect to design, EpiTalk is based on graphs, a formalism pedagogues feel comfortable with. Using a palette of behaviors, the designer can assembled them with a simple graph based interface. When a brand-new behavior is designed, it is automatically integrated in the palette. At run-time, continuations [11] are used to linked the behaviors.

3 Advisor Systems as Epiphyte Systems

The very nature of advisor is to observe the behaviors of other systems and to reason upon observed actions as soon as they are performed. By analogy, we consider advisors as *epiphyte* information systems (Fig.1). EpiTalk provides a precise framework to design advisor systems in a way that reflects the characteristics of advice (viewpoints-dependency and hierarchical nature), while providing satisfactory solutions to our needs (§2). EpiTalk organizes the reasoning of the EAIS according to a hierarchical knowledge graph that models the object aimed at by the advice. EpiTalk also supplies models to represent the knowledge needed to build an EAIS: structure of the object, advice contents, advising strategies... At run-time, EpiTalk enables an EAIS to follow the evolution of its host. In the next sections we describe how EpiTalk allows advisor systems to be specified as epiphyte systems.

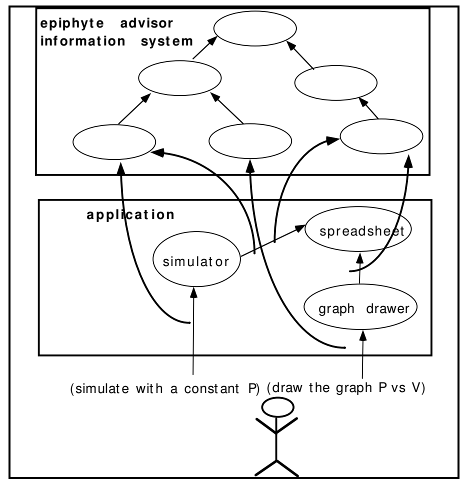


Fig. 1. An EAIS grafted onto a law induction system developed at Télé-université.

4 EpiTalk at One Glance

Within EpiTalk, description and run-time are distinct phases (Fig. 2). First the host and the EAIS are described. The host description is twofold: code (Smalltalk code, events...), and a description which gathers various information in one Smalltalk class. This information give the means to track the host evolution (user actions, new objects, new windows...) and provide the primitive actions an EAIS will have to interpret [20]. The EAIS descriptions are gathered in the knowledge graph: agents, communication network, and agent structure and reasoning process. Once descriptions are sufficiently precise, the run-time phase may proceed. The host is launched. A reflective description of the host is created and a causal link[2] is established. Then the multi-agent EAIS is generated and linked to its host through spies [22]. From then on, the EAIS agents receive spied information, analyze it and provide pieces of advice.

5 Describing an EAIS

A knowledge graph expresses the designer's point of view on the object the pieces of advice are on. There are as many knowledge graphs as there are points of view. A

[2] This link is causal because any modification in the host (for instance, a new window) is reflected in the description and conversely, any modification of the description is reflected in the host (for instance, deleting the object representing a window, close thewindow).

knowledge graph contains three kinds of nodes: non-terminal nodes, star nodes, and leaves. Star nodes are special non-terminal nodes; they express the notion of "0 to n" occurrences. For instance, the knowledge graph on courses designed with a knowledge-based didactic and generic workbench [28] (Fig. 3) specifies that a course is made of 0 to n objectives, and 0 to n learning units; a learning unit is made of...

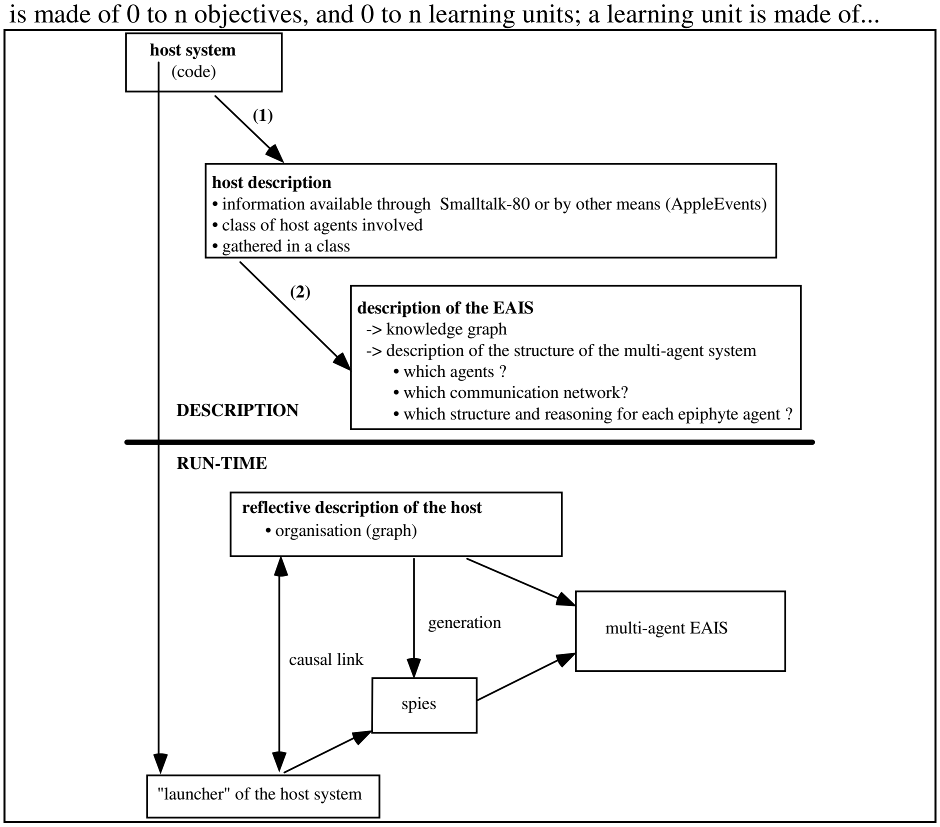
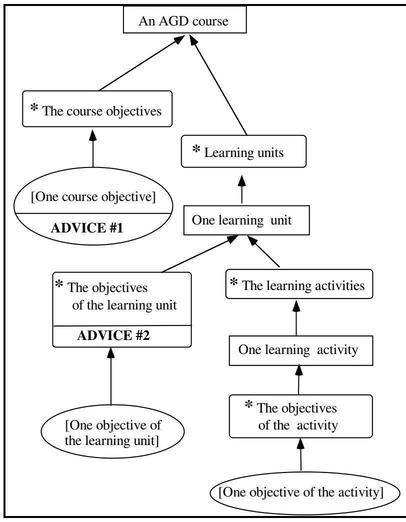


Fig. 2. EpiTalk uses descriptions to produce run-time EAIS.

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)




```
Condition
  the skill is unspecified while the
  ability level according to Bloom
  taxonomy is greater than 3
Advice #1:
  It is better to specify all the
  requirements related to an objective.
  When the objective level is <n>, they
  are usually expressed as numbers, for
  instance, a deadline, an error rate...
-----------------------------------
  --

Condition :
  more than 30% of the objectives have a
  taxonomic level less than 4 and expert
  learners are expected
Advice #2:
   <x%> of the objectives of the block
<b%> have low taxonomic levels:
memorize, understand or apply. Since
expert learners are expected, you should
set higher level objectives.
```

Fig. 3. A knowledge graph for a production.

Pieces of advice are assigned to any nodes. Leaves correspond to local points of view. As we goes up in the graph, the perspective on the course structure enlarges. Specific pieces of advice are placed at the bottom of the graph (advice #1 deals with one single course objective) while general ones require a larger perspective, and appear upper in the hierarchy (advice #2 deals with all the objectives of the learning unit). Pieces of advice can be contextualized.

The knowledge graph hierarchical structure fixes the multi-agent EAIS structure to generate (§7). Each node is used to generate either statically (leaves and non-terminal nodes) or dynamically (star nodes and their subnodes) one epiphyte agent implemented as a concurrent continuation. An epiphyte agent is itself a multi-agent system described by a behavior graph. Behavior nodes are predefined: memory, advisor… They are used to generate concurrent continuation behavior agents. Edges in knowledge (behavior) graphs define the communication net of the multi-agent EAIS (epiphyte agents). Because continuations obey to a uniform protocol, epiphyte (behavior) agents can be easily mixed. Thanks to the hierarchical nature of knowledge graphs, plan recognition and advising are performed in a single walk through at run-time. Since in EAIS, information sift up from leaves to roots, primitive host actions to observe are specified only for leaves. The nature of information to observe depends on hosts and platforms, e.g. selectors and classes for hosts coded in Smalltalk-80. This information is used to generate, graft, and link spies to EAIS' leaf epiphyte agents.

6 EAIS Generic with Respect to Hosts

The hierarchical nature of knowledge graph provides the right handles to achieve EAIS

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

generic with respect to hosts, operating systems, and platforms. Such genericity shelters EAIS from rapid software evolution. The basic idea is to partition the knowledge graph into spies, trace analyzer and advisor layers (Fig. 4). The spies layer collects primitive events (e.g. □-C). The trace analyzer layer combines primitive events into higher level ones (e.g. replacements). The advisor layer processes higher level events to produce pieces of advice. The advisor layer depicted in Figure 4 was reused for two different word processors: Microsoft Word for Windows 3.1 and a homemade word processor implemented in Smalltalk on a Macintosh. For the latter host application, the trace analyzer is a multi-agent system (Fig. 4). For the former, it is implemented as a syntactic analyzer for DDE based on a grammar[3]. As EAIS are designed, libraries of high-level events and translators are defined and enriched.

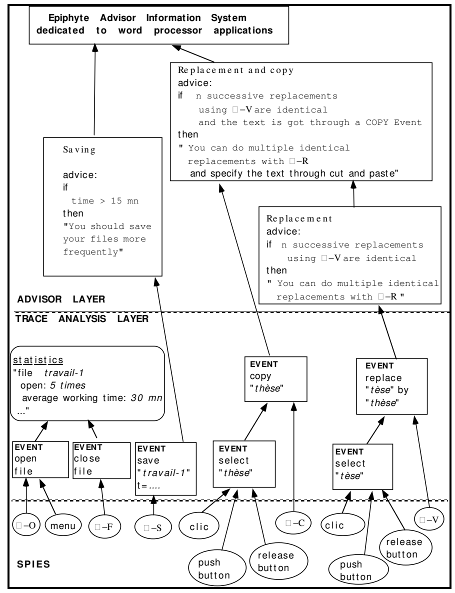


Fig. 4. A knowledge graph on word processing.

7 EAIS: Generation, Graft, and Evolution

The generation of an EAIS from the knowledge graph is straightforward. When the host is launched, the EAIS is generated following a top-down traversal of the graph (Fig. 5a). First roots of the knowledge graph are put in the set of nodes to instantiate. Next for each node in the set, an epiphyte agent is created, then the epiphyte agent's internal behavior agents are instantiated (§5, finally the epiphyte agent is linked to its hierarchical superiors with respect to the `part-of` relations. The instantiation process is then repeated with the sons of the instantiated nodes that are not star nodes. Sons of star nodes are excluded to delay the instantiation of their subnodes since there is no existing instance of the concepts they model. For instance, the host editor (Fig. 5a) shows that the AGD host has been launched. Three EAIS agents `a1`, `a2`, `a3` have been generated. The creation of agents stops there because epiphyte agents `a2` and `a3` correspond to star nodes: there is no objective and learning unit yet defined. Once created, the brand-new EAIS is grafted onto its host and begins to observe its behavior. As the host evolves, the EAIS evolves accordingly: each time a new object is created within the host, relevant star nodes are instantiated, and a spy is installed on the new host object and linked to epiphyte leaf agents interested in the new host object. For instance, opening the `course objectives editor` and defining one course objective triggers the lazy instantiation of the star node `*The objectives of the course` (Fig. 3) and the epiphyte agent `a4` is created (Fig. **5**b).

[3] [Ritter and Koedinger, 95] proposes an approach similar to this solution.

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

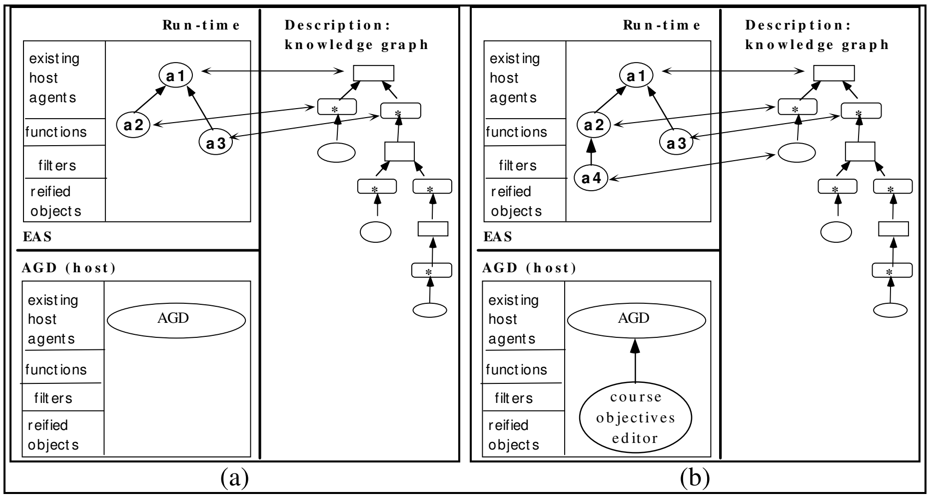


Fig. 5. The symbiotic evolution between the AGD host and an EAIS (cf Fig. 3).

8 EAIS for Collaborative Learning

First to investigate EpiTalk with respect to collaborative learning, we built up a simulation of a virtual campus based on Télé-université everyday practice, namely the learning and administrative procedures. A simulation involves administrative agents, tutor agents, learner agents and contract agents (Fig. 6a). Administrative agents register courses, learners and tutors. Tutor agents teach, give homeworks, and manage grades. Learner agents register to courses and do homework. They have a limited working capacity and limited knowledge. Contract agents manage homeworks according to the contract net protocol [32]. A homework is equivalent to a scenario. A graph describes the activities and the knowledge the learner must master to perform them. When an homework is issued, as many contract nets as it is necessary to give work to all registered learners are instantiated. Then learners divide work amongst themselves using the contract net protocol. A learner can bid on tasks he does not know how to achieve. In this case, he must learn how to do it. For instance, the simulation presented on Fig. 6 involves 4 learners (Etudiant), 2 tutors (Professeur), and 1 administrative agent (Registraire). Two homework contract net (ContratDevoir) were instantiated. Learner 4 is learning the knowledge required to perform its task.

Then a knowledge graph models collaborative work in the simulated campus (Fig. **6**b). 0 to n learners evolve in a virtual classroom. Tutors and teams are involved in teaching and support. A team is composed of contract net agents. At run -time, the EAIS is able to identify and model groups and interactions according to membership, team structure, leadership, division of the tasks among learners. The EAIS is also able to use these models to give pieces of advice to a member or to the group on balancing a team workload, directing a learner towards appropriate teaching resources (teammates, other learners in the classroom, tutors.

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

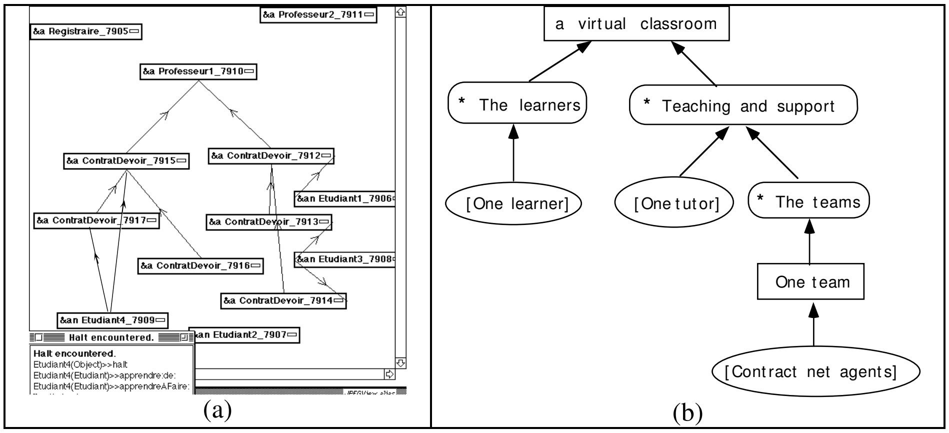


Fig. 6. A simulation run and a knowledge graph on collaborative learning.

9 Advice Strategy

EpiTalk supplies various means to provide pieces of advice. The designer sets a global advice strategy, but can also define local advice strategies on a per piece of advice basis. A piece of advice can be given an unlimited number of time or a limited one, only on request or at the full initiative of the advisor, using a dedicated passive window or a pop-up window, to specific individuals or to the whole group.

10 Current and Future works

A lot of work is still in progress. We are exploring the integration of multiple perspectives on a problem solving process, cooperation among several EAIS having different focuses, modeling erratic processes, detecting and modeling expertisation [17], exploiting new kinds of relations in knowledge graphs (e.g. precedence relation) in the production of generic advice module [23], teleconference analysis [12] in EAIS for collaborative learning, the articulation between individual and collaborative EAIS in a distributed Web-oriented virtual colloquium, the management of the physical distribution of EAIS epiphyte agents (for instance, local knowledge can be ascribed to leaves and thus the corresponding agents will be resident on the learners' workstation).

11 Conclusion

Télé-université supplies learners with information systems that either help them to manage their on-going learning processes or that enable them to perform their tasks. Furthermore distance learning asks for resources capable to start off again a stalled learning process. That is where advisors take up their duties. In distance education, advisors have to address both individualized and collaborative learning. In this paper, we have shown how EpiTalk helps to design, generate, launch, graft and monitor advisor information systems for both individual and collaborative learning processes. Advisors produced by EpiTalk rely on agents, organizational reflection and hierarchical knowledge graphs. Hierarchical graphs help to design and structure descriptions of an advisor. Using a hierarchical knowledge graph enabled to combine

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

at run-time plan recognition and the production of advice into a single walk-through. At run-time, agents provide autonomy, an essential feature, to each component of an advisor, and organizations link and co-ordinate agents. So EpiTalk addresses the design and description of advisors as well as their generation and unfolding at run-time. We have also shown how advisors could be designed in such a way to become platform, operating systems and applications independent. Since advisors were grafted onto existing arbitrary applications, we say that they are *epiphyte* by analogy with botany.

12 References

1. Bergeron, G., Paquette, G., Bourdeau, J.: *HyperGuide: An interactive student guide for a virtual classroom*, ED-MEDIA-93, Orlando, USA, June 23-26, 1993.
2. Christiansen, E.,Dirckinck-Holmfeld,L.:*Making Distance Learning Collaborative*, in3.
3. *CSCL '95, Computer Support for Collaborative Learning*, October 17-20, 1995, Bloomington, IN, USA, Lawrence Erlbaum Associates, Inc.
4. Gasser, L.: *Social Conceptions of Knowledge and Action: DAI Foundations and Open Systems Semantics*, Artificial Intelligence, 1991, vol. 47, no 1-3, pp. 107-138.
5. Giroux, S., Senteni, A., Lapalme, G.: *Adaptation in Open Systems*, First International Conference on Intelligent and Cooperative Information Systems (ICICIS), IEEE Computer Society Press, May 11-14, 1993, Rotterdam, Hollande, pp. 114-123.
6. Giroux, S.: *Agents et systèmes, une nécessaire unité*, Ph. D. thesis in computer science, DIRO, Université de Montréal, August 1993, Publication #883, 246 p.
7. Giroux, S.: *Open Reflective Agents*, in Intelligent Agents II, M. Wooldridge, J.-P. Müller and M. Tambe, eds, Springer-Verlag, LNAI-1037, 1996.
8. Giroux, S., Pachet, F., Paquette, G.: *Des systèmes d'information multi-agents épiphytes*, Deuxièmes Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents, Voiron, France, May 9-11, 1994, pp. 211-222.
9. Giroux, S., Pachet, F., Desbiens, J.: *Debugging Multi-Agent Systems: a Distributed Approach to Events Collection and Analysis*, CWDAI'94, Canadian Workshop on Distributed Artificial Intelligence, Banff, Canada, May 16, 1994.
10. Giroux, S., Pachet, F., Paquette, G., Girard, J.: *Des systèmes conseillers épiphytes*, Revue d'intelligence artificielle, Afcet/Hermès, vol. 9, no 2, 1995, pp. 165-190.
11. Hewitt, C.: *Viewing control structures as patterns of passing messages*, Artificial Intelligence, 1977, vol. 8, pp. 323-364.
12. Henri, F., Ricciardi-Rigault, C.: *Collaborative Learning and Computer Conferencing*. NATO Advanced Research Workshop, Grenoble, Sept. 1993. In Liao, T.T. ed. Advanced Educational Technology: Research Issues and Future Potential, Springer-Verlag, 1994.
13. Hotte, R.: *Encadrement assisté par ordinateur et formation à distance*, Revue de l'éducation à distance, vol. VIII, no 2, Fall 1993, pp. 37-53.
14. Karsenty, S., Pachet, F.: *Un mécanisme hiérarchique de répétition et de prédiction de tâches*, in Conférence IHM'95, 7e Journées sur l'Interaction Homme-Machine, Toulouse, France, October, 11-13, 1995.
15. Leman, S., Giroux, S., Marcenac, P.: *A Generic Distributed Method for Cognitive Modelling*, 7th Australian Joint Conference on Artificial Intelligence, Nov. 94.
16. Leman, S., Giroux, S., Marcenac, P., *A Multi-Agent Approach to ModellingStudent REAISoning Process*, AI-ED 95, Aug. 16-19, 95, Washington DC, USA, pp. 258-265.
17. Leman, S., Marcenac, P., Giroux, S.: *When the student surpasses the master*, 3rd International Conference on Computer Aided Engineering Education, CAEE'95, Bratislava, Slovakia, September 13-15, 1995, pp. 22-27.
18. Maes, P.: *Concepts and Experiments in Computational Reflection*, OOPSLA '87 Proceedings, Orlando, Florida, October 4-8, 1987, pp. 147-155.

Giroux et al. A Platform for Epiphyte Advisor Systems Dedicated to both individual and Collaborative Learning. Proc. of ITS'96, Montréal, Springer-Verlag Lecture Notes in Computer Science, 1086, pp. 363-371 (1996)

19. Marcenac, P., Giroux, S., Leman, S.: *A Multi-Agent Approach to Student Modelling*, EW-ED'94: Third East-West Conference on Computer Technologies in Education, 19th-23rd Sept. 1994, Yalta, Crimea, Ukraine, pp. 148-153.
20. Pachet, F., Giroux, S.: *Building plan recognition systems on arbitrary applications: the spying technique*, IJCAI-95 Workshop on The Next Generation of Plan Recognition Systems: Challenges for and Insight from Related Areas of AI, Aug. 20, 95, Canada.
21. Pachet, F., Giroux, S., Paquette, G.: *Pluggable Advisors as Epiphyte Systems*, CALISCE '94, August 31, September 1-2, 1994, Paris, France, pp. 167-174.
22. Pachet, F., Wolinski, F., Giroux, S.: *Spying as an object-oriented programming paradigm*, TOOLS Europe '95, France, March 6-9, 1995, Prentice-Hall, pp. 109-118
23. Pachet, F., Djamen, J.-Y., Frasson, C., Kaltenbach, M.: *Un mécanisme de production de conseils pertinents exploitant les relations de décomposition et précédence dans un arbre de tâches*, to appear in Sciences et Techniques Educatives, Hermès.
24. Paquette, G.: *Metaknowledge in the LOUTI development system*. Proc. of CSCSI-92, Canadian Society for Computational Study of Intelligence, Vancouver, Canada, May 92
25. Paquette, G.: *La modélisation par objets typés: une méthode de représentation pour les systèmes d'apprentissage et d'aide a la tâch*e. to appear in Sciences et techniques éducatives, Hermès.
26. Paquette, G., Bergeron, G., Bourdeau, J.: *The Virtual Classroom revisited*, Conference TeleTeaching'93, Trondheim, Norway, August 1993.
27. Paquette, G., Pachet, F., Giroux, S.: *ÉpiTalk, un outil générique pour la construction de systèmes conseillers*, Sciences et Techniques Educatives, Hermès, vol. 1, no 3, 1994.
28. Paquette, G., et al., *Design of a Knowledge-based Didactic and Generic Workbench*, CALISCE '94, August 31, September 1-2 1994, Paris, France, pp. 303-311.
29. Paquette, G., Pachet, F., Giroux, S., Girard, J.: *EpiTalk, a generic tool for the development of advisor systems*, to appear in J. of AI in Education.
30. Paquette, G., Ricciardi-Rigault, C., Bourdeau, J., Paquin, C., Liégeois, S.: *Modeling a Virtual Campus Environment for Interactive Distance Learning*, ED-Media International Conference, Graatz, Austria, June 1995.
31. Ritter, S., Koedinger, K. R.: *Towards lightweight tutoring agents*, AI-ED 95, August 16-19, 1995, Washington DC, USA, pp. 91-98.
32. Smith, R. G.: *The Contract Net Protocol: High-Level Communication and Control in a Disptributed Problem Solver*, IEEE Trans. on Computers, vol. C-29, no 12, 1980.
33. Wilson, B.G., Jonassen, D.H.: *Automated instructional systems design: A review of prototype systems.* J. of AI in Education, vol. 2, no 2, 1990, pp. 17-30.