ABEL-GUI: AN AGENT-BASED GRAPHICAL USER INTERFACE FOR E-LEARNING

Steffen Kernchen, Fritz Zbrog and Reiner R. Dumke
Department of Computer Science, Otto-von-Guericke-University of Magdeburg, Magdeburg, Germany

Keywords: e-Learning, Agent Technology, Graphical User Interface.

Abstract: In this paper we present an agent-based graphical user interface (GUI) for the domain of e-Learning: the ABEL-GUI. It is developed following a structured approach for the analysis, specification and design of agent-based graphical user interfaces. Therefore a logical goal/task hierarchy is designed and depicted by a society of agents. By this an implicit partition of the GUI takes shape and gets connected with the particular agents. The result takes benefits from the cooperation idea behind agent technology and focuses on specialization, personalization and interaction support.

1 INTRODUCTION

We think creating interfaces based on agent technology does not only follow the new paradigm of agent-based software development but leads to several advantages, too. O'Malley et al. list several guidelines whether an agent-oriented approach is useful (O'Malley and DeLoach, 2002). According to them agent-based graphical user interfaces can be necessary, because:

- GUI's may include complex/diverse types of interaction between components as well as to extern distributed heterogeneous resources. This flexibility of protocols can be easily supported by agent technology.
- Negotiation, cooperation and competition may occur among different entities.
- Some aspects of a GUI can have autonomous characteristics.
- A modification or expansion of the system can be anticipated. By using agents this flexibility is extended to modification or expansion during runtime of the system.

By choosing an agent-based approach additional aspects of flexibility can be applied for the domain of e-Learning. An increased support of adaptation for example leads to an improved adaptation to the user's needs. By choosing the development methodology presented in chapter 4 in combination with agent technology level-based as well as agent-based personalization of the functionality and the appearance of the GUI is possible. Basis of the adaptation are demands, properties, preferences, skills or interactions of the user that may occur or are explicitly modelled within a user model.

By means of agent technology, a framework development becomes possible. It can provide basic functions to fulfil minimal requirements. Additional, specialized agents can be added during runtime to extend the GUI. Thereby lightweight software provision and functional adaptation are distinguishing features of the system.

But why so much flexibility for e-Learning? Learning is one of the most important aspects of our culture. Its effective implementation and support is essential for human beings in all phases of their life. Life-long learning is the corresponding slogan. There are many changes in that long period of time. That refers to internal human being aspects as well as to external environmental ones. Human lifelong development requires highly flexible mechanisms for the adaptation of supporting systems. That applies to learning in a particular manner. External reasons are new scientific knowledge about didactics, psychol-

Kernchen S., Zbrog F. and R. Dumke R. (2007).
ABEL-GUI: AN AGENT-BASED GRAPHICAL USER INTERFACE FOR E-LEARNING.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Society, e-Business and e-Government / e-Learning*, pages 491-494
DOI: 10.5220/0001271504910494

ogy, interface design and new information about the topic to be learned. Versatile problems require adaptive and extendable solutions.

Following an agent-oriented methodology for designing graphical user interfaces we developed a framework of a front-end for a highly adaptive e-Learning system.

This paper is organized as follows. In section 2 we present basic key aspects that need to be considered in the context of our approach. The important factor of usability as well as the plug-in technology and the additional value contained in our approach are sketched out. The used methodology for the development of agent-based graphical user interfaces is presented in section 3. Key elements and novel features of the developed ABEL-GUI are part of section 4. Section 5 closes this paper with conclusions and future work.

2 FUNDAMENTALS

Following Shneiderman and Plaisant user interface design should be driven by user tasks and interface mechanisms (Shneiderman and Plaisant, 2005). The identified tasks must be real, complete and representative and the design that is developed following this task-centred approach must be validated to be appropriate. The other key element is the communication between user and system. The interface mechanisms must provide appropriate support for the user to solve his actual task.

There are three paradigms of user interface design described in (Tsou and Buttonfield, 1998). Traditional user interfaces operate within a centralized environment. They are not platform- and application independent. Client/Server user interfaces are clients remote connected with one server where data and program are located. They are usually platform independent like global user interfaces, which are not limited to connect to only one server. These interfaces can access multiple heterogeneous systems.

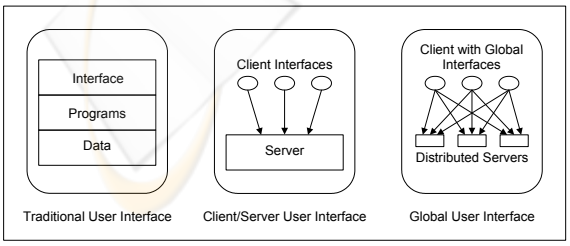

Figure 1: Three paradigms of user interface design (Tsou and Buttonfield, 1998).

Plug-in technology is an approach that targets the concept of mobile codes. Plug-ins are supplement software modules which are integrated in other software. They are only working in order with the application they are written for. Technologically they can be implemented in various ways: .NET, DCOM and Enterprise Java Beans to name a few.

Both approaches differ in the flexibility of usage and adaptation. Autonomous, pro-active and cooperative characteristics of agent technology are aspects that are not part of plug-ins, but provide essential advantages for the usage in context of e-Learning systems.

3 DEVELOPING AGENT-BASED GRAPHICAL USER INTERFACES

For developing agent-based graphical user interfaces an iterative goal-directed methodology including design patterns and creative techniques is presented here. As visualized in figure 2 there are several design stages.

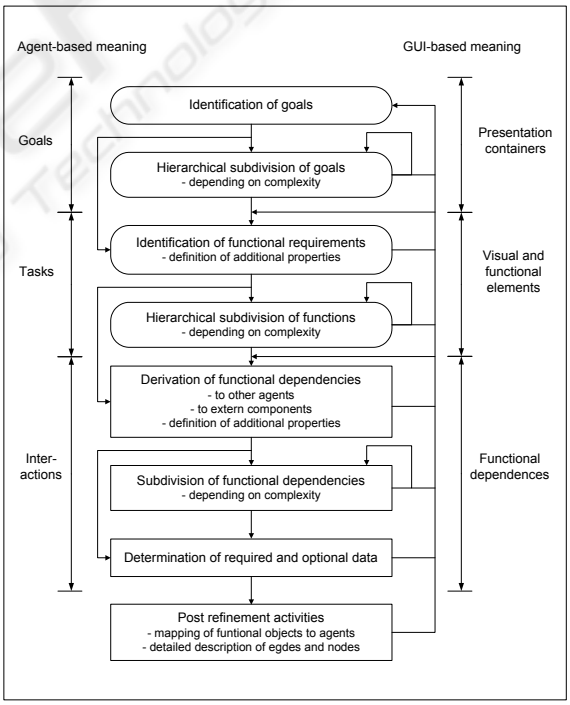

Figure 2: Stages of developing agent-based GUI's.

At the first stage the goals that should be solved are defined and refined. The starting point is the given problem in a special application domain. The goals and sub-goals have general character and form a tree. No special functions will be defined yet.

The special functionalities that are needed to reach the defined goals are identified at the second stage. If a function is too complex for an easy implementation its subdivision in smaller nodes is recommended. The complexity does not only refer to the functionality but also to the amount of dependencies to other nodes or external components.

A tree-shaped structure based on the subdivision of goals and tasks is defined so far. Now this structure is extended by the definition of additional internal and external connections. Next to the connections of the specified goal and task hierarchy - the edges of the tree - more internal functional dependencies may exist and need to be specified: a graph structure develops.

Up to this point the general structure of the multi-agent system and the graphical user interface is modelled. The next steps are the mapping of functional objects, respectively sets of those objects, to agents and the detailed description of edges and nodes. Complexity measures like the number of sub-functional objects or the number of edges can indicate an appropriate mapping. Sometimes aspects of the later implementation need to be taken into account, too. By recombining functionalities all sub-functional objects as well as all internal and external connections are compressed into one agent without disregarding the detailed functional design.

4 AGENT-BASED GUI FOR E-LEARNING

As already stated out flexibility, adaptation and personalization are key features of a GUI for e-Learning. Using the previously described methodology we developed a graphical user interface based on agent technology. By turning into account its heterogeneous and autonomous characteristics the intended goals can be achieved.

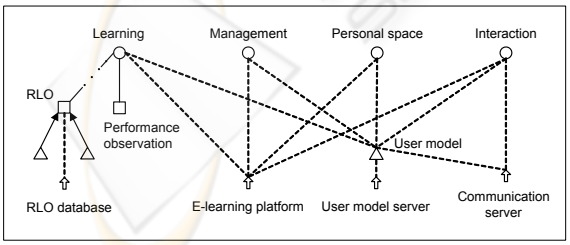

Figure 3: Core components of the ABEL-GUI and selected transitive connections to external distributed elements.

The framework benefits from the autonomy characteristic of agent technology in terms of improved modularity, adaptivity and flexibility. Proactivity and reactivity may directly result in adaptive changes of the component presentation. An example is the preference-based sortation of collaboratives in a chat list.

In this section we introduce main concepts of the architecture of the developed system from a student's point of view. Other roles are tutor, author and administrator and lead to a distinct usage of existing elements as well as to completely new features.

4.1 Learning Components

The first and obviously most important parts of the GUI to be developed are the learning components. That refers to all goals and functions that are directly coupled with the learning tasks.

Following the already described methodology we identified four direct sub-goals and one direct functionality, namely the course, course structure, domain visualization, support and actual announcements.

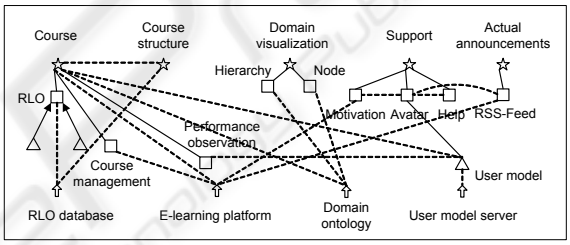

Figure 4: Graph presenting learning components, sub-components and functional edges.

4.2 Management Components

The adaptive functionality of this system is based on technical as well as organizational management. That refers to the management of agents, courses, user model and to logging mechanisms.

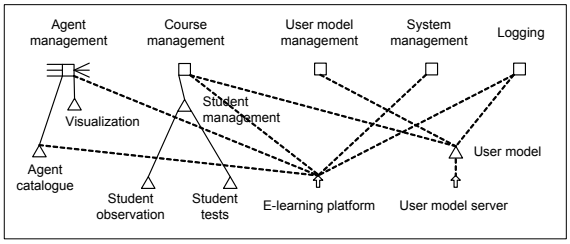

Figure 5: Graph presenting management components, chosen sub-components and functional edges.

4.3 Personal Components

Personal parts that might be implemented in an e-Learning GUI are e.g. a personal homepage, a course scheduler, personal space for storing of files and information, individualized search mechanisms and a kind

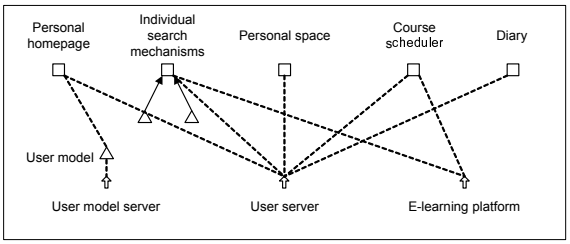

Figure 6: Graph presenting personal components, sub-components and functional edges.

of diary. A user server might be a technological base for these elements.

4.4 Interaction Components

Human interaction is essential for learning and learning-related motivation (e.g.: (Walther, 1992), (Dimitrova et al., 2003)). Our approach offers various core synchronous and asynchronous communication supporting elements and the easy usage, update and extension of this set. Examples are a wiki, chats, forums, user search component, etc.

All interaction components imply functional connections to the user model, the user server and the communication server. Preferences and logfiles need to be stored for every communication tool for personalization and adaptation.

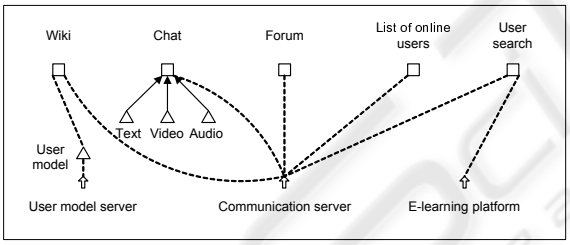

Figure 7: Graph presenting interaction components, chosen sub-components and functional edges.

4.5 Implementation Remarks

This GUI is intended to be implemented as a framework. That is an important aspect for the provision of the flexibility, the necessity and advantages of which was proven in the introduction. Therefore minimal required elements as well as necessary infrastructural aspects must be defined.

At this point we argue that the core elements shown in figure 4, their direct sub-elements and all elements, that are directly or transitive connected by required edges, are non-optional parts of the framework. This is visualised in the particular figures of the other subsections of this section.

Infrastructural aspects that must be respected are the identification of agents, docking mechanisms, interaction interfaces, the communication itself and warranty of role-based access and modification issues.

Interaction is one of the most important features of agent technology. Most of their advantages are based on communication. In the context of interfaces external and internal communication needs to be optimized to guarantee the flow of usage. Those are contrary goals that must be carefully balanced and that directly refer to the implementation technology.

Performing post-refinement activities is a stage of the used methodology that is between refinement of goals/functionalities and the implementation. It pools sub-functions belonging together into one agent.

5 CONCLUSIONS AND FURTHER WORK

In this paper we presented main concepts of the ABEL-GUI, a graphical user interface for the domain of e-Learning, which was developed following a graph-based approach for developing agent-based GUI's. High flexibility, organized interaction mechanisms as well as adaptation support are key benefits of this technique. To examine the usefulness of our approach we modelled the presented agent-based interface for the e-Learning domain as a framework. An integral part of our future research is its implementation using existing web technologies by extending them with agent characteristics.

REFERENCES

Dimitrova, M., Sadler, C., Hatzipanagos, S., and Murphy, A. (2003). Addressing learner diversity by promoting flexibility in e-learning environments. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA03)*.

O'Malley, S. A. and DeLoach, S. A. (2002). *Determing When to Use an Agent-Oriented Software Engineering Paradigm*, pages 188–205. Springer.

Shneiderman, B. and Plaisant, C. (2005). *Designing the User Interface – Strategies for Effective Human-Computer Interaction*. Pearson Education, 4th edition.

Tsou, M.-H. and Buttonfield, B. (1998). An Agent-Based, Global User Interface for Distributed Geographic Information Services. In *Proceedings of the 7th International Symposium on Spatial Data Handling*, pages 602–612.

Walther, J. B. (1992). Interpersonal effects in computer-mediated interaction: A relational perspective. *Communication Research*, pages 52–90.