MOOSCo: Multi-user Object-Oriented environments with Separation of Concerns

Project Proposal Submitted to FCT / MCT by INESC´s Software Engineering Group and LASIGE Team at Faculdade de Ciências

January 2000

1 Resumo

Os sistemas iteractivos multi-utilizador distribúidos são uma área aplicacional extremamente relevante. Aplicações como os ambientes virtuais, simulação distribuída, trabalho cooperativo suportado por computador (CSCW, MUDS), e ambientes multi-utilizador com objectos estão-se a tornar cada vez mais comuns.

Estas aplicações impoêm um conjunto de requisitos muito ricos e complexos do ponto de vista da análise, engenharia de software e suporte do sistema. Uma aproximação promissora para lidar com a complexidade destes sistemas é depender de arquitecturas configuraveis que são capazes de suportar a reutilização e composição de componentes.

O projecto propõe o estudo das dificuldades na aplicação de uma aproximação baseada em componentes de um forma vertical e integrada, da análise à implementação.

O projecto junta três equipas com experiência na utilização e construção de componentes configuráveis em diferentes níveis de abstracção: análise baseada em componentes de aplicações cooperativas (Electronic Meetings LASIGE team), molduras configuraveis para software distribúido (INESC Software Engineering team) e protocolos de comunicação configuráveis (Infrastructure Protocols LASIGE team).

O projecto irá definir uma arquitectura que será aplicada no contexto dos MOOS. Os ambientes MOO constituem um desafio para a teoria e prática dos sistemas distribuídos orientados a objectos devido aos seus requisitos únicos para a escalabilidade, adaptabilidade, usabilidade, alterações dinâmicas, domínios não funcionais a serem considerados e eficiência. A satisfação completa destes requisitos não é fácil e pode resultar em soluções conflictuosas e inconsistentes. Um desafio estimulante do projecto é o estudo de políticas e mecanismos de configuração e de composição consistentes em todos os níveis da arquitectura.

O trabalho para alcançar os objectivos do projecto está dividido em várias tarefas: (1) Identificar os modelos do utilizador que são relevantes para os MOOs de forma a classificar as variantes inerentes às aplicações MOO. (2) Desenhar abstracções de middelware passiveis de serem compostas e adaptadas de forma a suportar os modelos de utilizador e as suas variações, e permitir a construção de MOOs que são facilmente extensiveis e afinados. Implementar essas abstracções na forma de uma moldura de objectos. (3) Desenhar protocolos de infrastrrtura passiveis de serem compostos e adaptados, oferecendo diferentes qualidades de serviço de forma a serem utilizados pelas abstracções de middelware. (4) Integrar e validar os resultados das tarefas anteriores através da construção de um sistema MOO.

2 Summary

Distributed multi-user interactive systems are an extremely relevant application area. Applications such as virtual environments, distributed simulation, computer supported collaborative work (CSCW), multi-user games or dungeons (MUDs), and multi-user object-oriented environments (MOOs) are becoming increasingly pervasive.

These applications pose a very rich and complex set of requirements from the analysis, software engineering and system support point-of-view. A promising approach to tackle the complexity of these systems is to rely on configurable architectures that are able to support component re-utilization and composition.

The project proposes to study the difficulties in applying a component-based approach in a vertical and integrated manner, from analysis to implementation. The project brings together three teams with experience in using and building configurable components at different levels of abstraction: component based analysis of cooperative applications (Electronic Meetings LASIGE team), configurable distributed software frameworks (INESC Software Engineering team) and configurable communication protocols (Infrastructure Pro-

tocols LASIGE team).

The project will define an architecture that will be applied in the MOOs context. MOO environments constitute a challenge for object-oriented distributed systems theory and practice due to its unique requirements for scalability, adaptability, usability, dynamic changes, non-functional domains to be considered, and efficiency. The complete satisfaction of these requirements is not easy and may result in conflicting and inconsistent solutions. A challenging aspect of the project is to study policies and mechanisms to ensure that consistent configuration and composition choices are made at all levels of the architecture.

The work to reach the project goals is divided in several tasks: (1) Identify the user models that are relevant for MOOs in order to classify the variabilities inherent to the MOO applications. (2) Design composible and customizable middleware abstractions to support user models and their variabilities, and allow the construction of MOO application that are easily tuned and can evolve. Implement those abstractions as an object-oriented framework. (3) Design composible and customizable infrastructure protocols, offering different quality of service, to be used by the middleware abstractions. (4) Integrate and validade the results from the previous tasks in a concrete MOO system.

3 Project´s Description

3.1 Objectives

The project intends to design and implement an architecture to the support of multi-user object-oriented environments. The architecture is based on component composition and addresses three abstraction layers: user models, middleware abstractions, and infrastructure communication protocols. The architecture should, through component composition, integrate these layers ensuring traceability. Traceability allows to measure the impact that changes have through layers and specifies the requirements that each layer imposes on the layer bellow.

The project proposes a innovative approach to the design and development of MOO environments. Due to its compositionality characteristics it is possible to use middleware abstractions and communication protocols tailored to the specific user models needed in each case. Moreover, the same MOO application can possess different user models and they can change at run-time. This novel approach allows a ”best buy” strategy according to user models.

3.2 Long Term Research Objectives

The main research area of the principal proponent is separation and composition of concerns. In that context an approach for the development of concurrent and distributed applications was previously defined. MOOSCo is an application of that approach to the MOO domain.

In terms of long term research this project will provide important feedback to the problem of separation of concerns approaches. Composing different concerns is hard and complex. So, an automatic support that simplifies final programmers from dealing with composition issues is envisaged. However, it is now clear to researchers that there are some semantics in the composition: the whole is more than the sum of the parts. This means that it is impossible to hide composition from the final programmer. The long term open problems we want to address are: (1) What can be hidden from final programmers and what should be explicit? (2) What kind of architectures can support compositions that consider the above properties?

3.3 State of the Art

Compositional approaches are attractive and becoming more and more fashionable. However, it is dangerous to have a naive approach to development with components. There are very hard open problems that current research results have identified but that are far from being solved. The main problems are related to the composition of non-orthogonal components, components which composition semantics is more than the sum of the parts. The current research approaches present some drawbacks : Aksit´s composition filters[Aksit94] do not address the problem of non-orthogonal composition; aspect-oriented programming[Kicsales97] allows non-orthogonal composition and delegates on automatic code generation the resolution of complexity associated with concern combination, but, it is not clear how non-orthogonal compositions can be handled automatically and with some generality; CodA[McAffer95] allows non-orthogonal composition, but, at a lower granularity than the component, which results on encapsulation break and, so, composition becomes very complex. [Silva99a] proposes a three-layered architecture with separation of concerns that deals with non-orthogonal composition. This framework was tested for some domain-specific components, as for instance object synchronization.

Existing systems, such as MASSIVE[Greenhalgh95] and DIVE[Hagsand96], fail, from a software engineering perspective, to provide a fully-fledged compositional, reusable and customizable approach to the design and implementation of MOO environments. As result of its monolithic structure these systems are restricted to a single user model and to a restricted set of middleware abstractions.

In this project we intend to apply a separation of concerns approach to the development of MOO environments in order to: provide a compositional approach to the construction of MOO environments; apply and test the three-layered architecture in this domain; generalize the architecture.

From the point of view of communications support, existing MOO systems are usually tied to a single quality of service. For instance MASSIVE only supports a single multicast communication protocol and DIVE requires mesages to be totally ordered. This represents a severe limitation since a single primitive is not adequate to support the wide range of different requirements imposed by MOOs. In recent years, there has been a significant progress in the development of group communication infrastructures. The latest systems offer a very impressive range of configuration facilities. For instance, Horus[Renesse95] allows communication stacks to be changed in runtime; BAST[Garbinato98] allows different protocols to be selected to implement the same services under different usage patterns; Coyote[Bhatti98] allows the same message to be processed by different protocols in parallel. Unfortunately, few of these new advances are currently used in the MOO design, maybe due to a phenomenon of interface mismatch that has not yet been clearly understood.

3.4 Other Projects

The INESC Software Engineering group has been developing DASCo (Development of Distributed Applications with Separation of Concerns) since 1995 (http://www.esw.inesc.pt/ ars/dasco.html). MOOSCo is an application of DASCo approach to the context of distributed multi-user interactive systems. DASCo is defined for constructing concurrent and distributed programs which separately handles, during the development process, functional and non-functional concerns. The approach defines solutions for each non-functional concern related to concurrency and distribution, such as object synchronization or configuration, and integrates those solutions as well as their composition in an tailorable incremental development process of concurrent and distributed programs. MOOSCo intends to apply DASCo approach to the MOO domain in order to define a development process and artifacts possessing the software engineering qualities DASCo aims at.

In terms of research the MOOSCo, and DASCo, projects address issues that are closely related to Multi-Dimensional Separation of Concerns (MDSC - http://www.research.ibm.com/hyperspace/). MDSC a is project that intends to aggregate several approaches of separation and composition of concerns. The project goals are to enable: (1) encapsulation of all kinds of concerns in a software system, simultaneously; (2) overlapping and interacting concerns; and (3) on-demand remodularization. MOOSCo contribution will address points 1 and 2.

The Infrastructure Protocols LASIGE team is currently involved in a project that aims to implement some of the fault-tolerant group communication algorithms designed by the team using the Ensemble platform. The project, called TOPCOM, started a few months before the writing of this proposal. This work complements the previous experience of the team in the area of configurable communication kernels. Given the timeframe of both projects, it is expected that results from TOPCOM could be incorporated in MOOSCo, in particular, the implementation of the light-weight group service described in[Rodrigues2000].

The Systems Analysis Team at LASIGE has been experimenting participatory software development approaches in the context of of a project designated Decision and Creativity in Electronic and Natural Groups, financed by PRAXIS. The project main goal was to set up an electronic meeting room at the ISCTE university. To date, this room supported more than 100 meetings. The

MOOSCo project will allow to combine and consolidate several approaches that have beed loosely experimented (e.g. colaborative requirements elicitation) into an articulated analysis framework.

3.5 Diffusion of Results

Dissemination activities will be based on the production of research papers and on the free distribution of software. We believe that 2 workshop and 9 conference papers can be produced based on the work described in the current proposal. It is also expected to produce 2 journal paper summarizing the results of the project as a whole.

The INESC´s Software Engineering Group maintains a Internet site dedicated to the MOOSCo project (http://www.esw.inesc.pt/ ars/moosco.html), where the documents, software and results of the project will be also made available for public distribution.

Finally, since the project foresees the involvement of Master's and PhD´s students, the production of 2 MsC thesis and 1 PhD thesis in the framework of the project is expected.

3.6 Ethic, Social and Environment Impacts

There are no negative impact of any kind.

4 Work Plan

4.1 User Models Task

The Electronic Meetings team at LASIGE participating in this project is dedicated to research and develop systems and competences in the general field of Systems Analysis, considering in particular complex socio-technical systems, with multiple users, where requirements are unknown and problems deal with ethical issues.

Socio-technical systems are closely tied to the situated activities, social constructions and communication exchanged by their specific communities of users. In such context, it is extremely difficult to develop generic, or abstract, analysis and design solutions, which may not be recognizable outside a community of practice. That is the fundamental goal pursued by the team.

This project is particularly suitable to the general objectives described above for the reason that two privileged conditions are met. First, MOOs make up a rich socio-technical system, based on information sharing, collaboration and long-term community development. Second, the project scope, which makes up a complete vertical approach to systems development, provides a good setting where to extend, experiment and validate several key analysis methods and tools, which the team has been developing for some time.

This task will specifically contribute with a collection of analysis / design components that specify the situated functionality of MOOSCo.

The construction of these components will be rooted on the following approach:

1. System models must be specified in collaboration with users. We have been extensively using Group Support Systems to accomplish this level of user involvement.

2. System models must articulate the concepts of genre and pattern. A genre is a socially recognisable type of communicative action, which is habitually enacted by members of a community to accomplish particular social purposes (e.g. be accepted by MOO's members). A pattern is a kind of reusable solution to design problems (e.g. entering or finding who is in a MOO). Some particular characteristics of this approach should be stressed and confronted with other participative methodologies to software development. The most important is that there is much more technological support to the analysis process. The second is that technological support is more oriented towards increasing user participation (rather than modelling). To this respect, empirical findings have proved the positive impact of Group Support Systems, measured by increased quality, commitment and satisfaction with outcomes.

4.2 Middleware Abstractions Task

In the MOOs context there is not a unique and best solution. Solutions should be contextual. For instance, the complete satisfaction of MOO requirements for consistency, adaptability, scalability, and efficiency, is not easy and may result in conflicting and inconsistent solutions. In addition, domain-specific requirements may consider different levels of consistency and even their change at runtime.

Due to these requirements MOOs design and development will profit from an approach that allows the customization of contextual solutions by the tuning and composition of predefined reusable components. These reusable components should support user models variability and impose on infrastructure protocols the quality of service that they rely on to provide their services.

The INESC Software Engineering team has developed expertise on the development of concurrent and distributed applications, DASCo, using a "divide and conquer" strategy that allows the most appropriate solution for each concern to be customized by the programmer to take into account the specific needs of the program being built.

This task will define middleware abstractions and implement object-oriented frameworks for MOOs. The abstractions and frameworks will have the qualities defined by DASCo such that the development of MOOs can profit from a customizable, compositional and reusable approach. We will apply separation of concerns to two aspects of MOO design: domain abstractions; and distributed architectures. For instance, domain abstractions variabilities for scope and ownership will result from the customization of abstractions for space partitioning and object synchronization. Concerning distributed architectures, separation of concerns will allow incremental introduction of distribution, according to the number and location of users. For instance, if a virtual meeting is going to occur and most of the participants are located in the same place, efficiency would be improved if the class room is located in a nearby server, and that way, there is also a scalability improvement.

4.3 Infrastructure Protocols Task

MOOs are quite challenging applications from the communications point of view.

To start with, MOOS are multi-user applications, thus point-to-point primitives are not the most adequate to support them. Instead multi-point or group communication services should be used. Even a centralized version of a MOO can benefit from multi-point communication to disseminate information from the central server to all participants. However, as we have seen, centralized solutions have severe scalability problems. When cooperative servers are used, group communicaton can play an even more relevant role.

Additionally, MOOs can be communication intensive and the communication infrastructure may become a bottleneck if not used wisely. Unfortunately, short latencies are also desirable, which represents a conflicting goal with optimal resource utilization. Thus, MOOs also pose significant efficiency and performance constraints.

Finally, some of the MOOs objects have strong consistency requirements while others, in the same system, may accept weaker requirements or even no reliabillity at all. Several MOOs have been implemented using total order protocols which are known for their high latency and number of communication rounds, which conflicts with the goals previously stated. Satisfying exactly the requirments imposed by each application object is a strong contribution to the overall performance of the MOO.

The Infrastructure Protocols Lasige team is currently involved in testing and developing group communication infrastructures that support fine grain configuration. This includes experience developing layers for third-party communications kernels, such as Horus[Renesse95] and Ensemble[Hayden98], and experience with the develpment of new mechanisms for an in-house prototype system [Miranda99]. The MOOSCo project is the ideal test-bed for this class of systems.

In this task, a configurable group communication infrastructure will be deployed, experimented and evaluated. The actual infrastructure to be used will be selected based on the maturity of state of the art systems at the time the project is initiated. Feedback from middleware and application designers will be key to the configuration of the communication services and protocols.

The development of specialized high-level communication protocols, in order to better satisfy specific requirements of the MOOSCo architecture, is not excluded. The performance of these tailored solutions will be compared with that of generic protocols currently available.

4.4 Integration and Validation Task

As a result of this project we propose to develop an application where the results of all the previous tasks will be integrated and validated. The application will be a multi-user virtual environment with the following characteristics:

- The virtual environment will be formed by a set of interconnected multi-user Worlds. The clients can navigate through the worlds and between the worlds using a client application called a world browser. Each world may define its own user models, its distributed architecture (centralized, fully distributed, etc), and use different communication infrastructures (reliable multi-point, unreliable multi-point, point-to-point). The client should be able to navigate between the worlds using its world browser without having to be aware of those differences (except for the user models).

- It should be possible to use different world partitioning policies. One can partition the world in different managed regions, and each region will be managed by a different world server. The regions can be explicit or implicit. With explicit regions the user can have a sense of discontinuity, i.e., the user knows he is changing region (entering a room, jumping to a different world). With implicit regions the user moves between regions and worlds with the perception of a continuous space and continuous coordinates.

- The application should support extensibility and customization so that special purpose worlds and objects can be developed. For instance, a special conference room could be defined where only the person doing the presentation is able to talk, and all the other participants will only be able to ear and post questions. Or a meeting table could be defined where the users near the table could talk with each other while the others could only be aware of a group of users sit together around a table.

- It should be possible to specify the preferred quality of service at different levels. At the user model level, it should be possible for the user to specify the amount and type of information that he is interested in (ex: he may wish only to communicate with users that are at pre-defined distance from him, he may want to ignore all the communication happening in the world, he may specify the level of-detail for the appearance of the objects in the world); at the communication infrastructure level, it should be possible to use different quality of services for different types of data (unreliable for sound and video, reliable for world state transfers, and reliable and ordered for some operations performed in some special objects).

To validate the results it will be measured: (1) the usability of user models definition; (2) the expressive power and simplicity of middleware abstractions and infrastructure protocols to support user models; (3) dynamic change of user models by replacing middleware abstractions and infrastructure protocols; (4) adaptability to change the distributed architecture according to scalability requirements.

5 References

[Aksit94] M. Aksit, K. Wakita, J. Bosch, L. Bergmans and A. Yonezawa. Abstracting Object Interactions Using Composition Filters. ECOOP'93 Workshop on Object-Based Distributed Programming. LNCS 791. Springer-Verlag. pp. 152-184. 1994.

[Bhatti98] N. Bhatti, M. Hiltunen, R. Schlichting, and W. Chiu. Coyote: A System for Constructing Fine-grain Configurable Communication Services. ACM Transactions on Computer Systems, 16(4):321-366, November 1998.

[Garbinato98] B. Garbinato, and R. Guerraoui. Flexible Protocol Composition in Bast. IEEE Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS-18), pages 22-29, Amsterdam, The Netherlands, May 1998.

[Greenhalgh95] C. Greenhalgh, and S. Benford. MASSIVE: A Virtual Reality System for Tele-conferencing. ACM Transactions on Computer Human Interfaces (TOCHI), 2 (3), pp. 239-261, September 1995.

[Hagsand96] O. Hagsand. Interactive MultiUser VEs in the DIVE System. IEEE Multimedia Magazine, Vol 3, Number 1. 1996.

[Hayden98] M. Hayden. The Ensemble System. PhD thesis, Cornell University, Computer Science Department, 1998.

[Kicsales97] G. Kicsales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier and J. Irwin. Aspect-Oriented Programming. XEROX PARC Technical Report SPL97-008 P9710042. February 1997.

[McAffer95] J. McAffer. Meta-level Programming with CodA. Proceedings of ECOOP'95. pp. 190-214. Aarhus, Denmark. August 1995.

[Miranda99] H. Miranda and L. Rodrigues. Flexible Communication Support for CSCW Applications. 5th Internation Workshop on Groupware, Cacún, México, September 22-24, 1999.

[Renesse95] R. van Renesse, K. Birman, R. Friedman, M. Hayden, and D. Karr. A framework for protocol composition in Horus. 14th Annual ACM Symposium on Principles of Distributed Computing, pages 80-89, Ottawa, Ontario, Canada, 2-23 August 1995.

[Rodrigues2000] L. Rodrigues and K. Guo. Partitionable Light-Weight Groups. IEEE Proceedings of the 20th International Conference on Distributed Computing Systems, Taipe, Taiwan, April, 2000.

[Silva98a] A. R. Silva. Development and Extension of a Three-Layered Framework. Handbook of Object Technology. Saba Zamir Editor. CRC Press, 1998.

[Silva99a] A. R. Silva. Concurrent Object-Oriented Programming: Separation and Composition of Concerns using Design Patterns, Pattern Languages, and Object-Oriented Frameworks. PhD Thesis. Technical University of Lisbon. March 1999.

[Silva99b] A. R. Silva and M. Antunes. From DASCo to MOOSCo: MOO with Separation of Concerns. ECOOP '99 Multi-User Object-Oriented Environments Workshop. Lisboa, Portugal. June 1999.