LNCS 3675

Yuhua Luo (Ed.)

Cooperative Design, Visualization, and Engineering

Second International Conference, CDVE 2005
Palma de Mallorca, Spain, September 2005
Proceedings

Springer

Yuhua Luo (Ed.)

Cooperative Design, Visualization, and Engineering

Second International Conference, CDVE 2005
Palma de Mallorca, Spain, September 18-21, 2005
Proceedings

Springer

Volume Editor

Yuhua Luo
University of Balearic Islands
Department of Mathematics and Computer Science
07122 Palma de Mallorca, Spain
E-mail: dmilyu0@uib.es
http://www.cdve.org

Library of Congress Control Number: 2005932201

CR Subject Classification (1998): H.5.3, H.5.2, H.5, H.4, C.2.4, D.2.12, D.4, H.2.8

ISSN 0302-9743
ISBN-10 3-540-28948-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28948-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11555223 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3675

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3579

please contact your bookseller or Springer

Vol. 3710: M. Barni, I. Cox, T. Kalker, H.J. Kim (Eds.), Digital Watermarking. XII, 485 pages. 2005.

Vol. 3703: F. Fages, S. Soliman (Eds.), Principles and Practice of Semantic Web Reasoning. VIII, 163 pages. 2005.

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005. (Subseries LNAI).

Vol. 3697: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, Part II. XXXII, 1045 pages. 2005.

Vol. 3696: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Biological Inspirations - ICANN 2005, Part I. XXXI, 703 pages. 2005.

Vol. 3687: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Image Analysis, Part II. XXV, 809 pages. 2005.

Vol. 3686: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Data Mining, Part I. XXVI, 689 pages. 2005.

Vol. 3675: Y. Luo (Ed.), Cooperative Design, Visualization, and Engineering. XI, 264 pages. 2005.

Vol. 3674: W. Jonker, M. Petković (Eds.), Secure Data Management. X, 241 pages. 2005.

Vol. 3672: C. Hankin, I. Siveroni (Eds.), Static Analysis. X, 369 pages. 2005.

Vol. 3671: S. Bressan, S. Ceri, E. Hunt, Z.G. Ives, Z. Bellahsène, M. Rys, R. Unland (Eds.), Database and XML Technologies. X, 239 pages. 2005.

Vol. 3670: M. Bravetti, L. Kloul, G. Zavattaro (Eds.), Formal Techniques for Computer Systems and Business Processes. XIII, 349 pages. 2005.

Vol. 3665: K. S. Candan, A. Celentano (Eds.), Advances in Multimedia Information Systems. X, 221 pages. 2005.

Vol. 3664: C. Türker, M. Agosti, H.-J. Schek (Eds.), Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures. X, 261 pages. 2005.

Vol. 3663: W.G. Kropatsch, R. Sablatnig, A. Hanbury (Eds.), Pattern Recognition. XIV, 512 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005. (Subseries LNAI).

Vol. 3660: M. Beigl, S. Intille, J. Rekimoto, H. Tokuda (Eds.), UbiComp 2005: Ubiquitous Computing. XVII, 394 pages. 2005.

Vol. 3659: J.R. Rao, B. Sunar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2005. XIV, 458 pages. 2005.

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 pages. 2005. (Subseries LNAI).

Vol. 3655: A. Aldini, R. Gorrieri, F. Martinelli (Eds.), Foundations of Security Analysis and Design III. VII, 273 pages. 2005.

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3653: M. Abadi, L.d. Alfaro (Eds.), CONCUR 2005 – Concurrency Theory. XIV, 578 pages. 2005.

Vol. 3649: W.M.P. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management. XII, 472 pages. 2005.

Vol. 3648: J.C. Cunha, P.D. Medeiros (Eds.), Euro-Par 2005 Parallel Processing. XXXVI, 1299 pages. 2005.

Vol. 3646: A. F. Famili, J.N. Kok, J.M. Peña, A. Siebes, A. Feelders (Eds.), Advances in Intelligent Data Analysis VI. XIV, 522 pages. 2005.

Vol. 3645: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part II. XIII, 1010 pages. 2005.

Vol. 3644: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part I. XXVII, 1101 pages. 2005.

Vol. 3642: D. Ślezak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIII, 738 pages. 2005. (Subseries LNAI).

Vol. 3641: D. Ślezak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005. (Subseries LNAI).

Vol. 3639: P. Godefroid (Ed.), Model Checking Software. XI, 289 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3637: J. M. Moreno, J. Madrenas, J. Cosp (Eds.), Evolvable Systems: From Biology to Hardware. XI, 227 pages. 2005.

Vol. 3636: M.J. Blesa, C. Blum, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics. XII, 155 pages. 2005.

Vol. 3634: L. Ong (Ed.), Computer Science Logic. XI, 567 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3631: J. Eder, H.-M. Haav, A. Kalja, J. Penjam (Eds.), Advances in Databases and Information Systems. XIII, 393 pages. 2005.

Vol. 3630: M.S. Capcarrere, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (Eds.), Advances in Artificial Life. XIX, 949 pages. 2005. (Subseries LNAI).

Vol. 3629: J.L. Fiadeiro, N. Harman, M. Roggenbach, J. Rutten (Eds.), Algebra and Coalgebra in Computer Science. XI, 457 pages. 2005.

Vol. 3628: T. Gschwind, U. Aßmann, O. Nierstrasz (Eds.), Software Composition. X, 199 pages. 2005.

Vol. 3627: C. Jacob, M.L. Pilat, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 500 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005. (Subseries LNAI).

Vol. 3624: C. Chekuri, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization and Combinatorial Optimization. XI, 495 pages. 2005.

Vol. 3623: M. Liśkiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory. XV, 576 pages. 2005.

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005. (Subseries LNAI).

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3618: J. Jedrzejowicz, A. Szepietowski (Eds.), Mathematical Foundations of Computer Science 2005. XVI, 814 pages. 2005.

Vol. 3617: F. Roli, S. Vitulano (Eds.), Image Analysis and Processing – ICIAP 2005. XXIV, 1219 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005. (Subseries LNAI).

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005. (Subseries LNAI).

Vol. 3612: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part III. LXI, 1326 pages. 2005.

Vol. 3611: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part II. LXI, 1292 pages. 2005.

Vol. 3610: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part I. LXI, 1302 pages. 2005.

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3606: V. Malyshkin (Ed.), Parallel Computing Technologies. XII, 470 pages. 2005.

Vol. 3604: R. Martin, H. Bez, M. Sabin (Eds.), Mathematics of Surfaces XI. IX, 473 pages. 2005.

Vol. 3603: J. Hurd, T. Melham (Eds.), Theorem Proving in Higher Order Logics. IX, 409 pages. 2005.

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3599: U. Aßmann, M. Aksit, A. Rensink (Eds.), Model Driven Architecture. X, 235 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Vol. 3597: S. Shimojo, S. Ichii, T.W. Ling, K.-H. Song (Eds.), Web and Communication Technologies and Internet-Related Social Issues - HSI 2005. XIX, 368 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3595: L. Wang (Ed.), Computing and Combinatorics. XVI, 995 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005. (Subseries LNBI).

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005. (Subseries LNAI).

Vol. 3592: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust, Privacy and Security in Digital Business. XII, 332 pages. 2005.

Vol. 3591: M.A. Wimmer, R. Traunmüller, Å. Grönlund, K.V. Andersen (Eds.), Electronic Government. XIII, 317 pages. 2005.

Vol. 3590: K. Bauknecht, B. Pröll, H. Werthner (Eds.), E-Commerce and Web Technologies. XIV, 380 pages. 2005.

Vol. 3589: A M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery. XVI, 538 pages. 2005.

Vol. 3588: K.V. Andersen, J. Debenham, R. Wagner (Eds.), Database and Expert Systems Applications. XX, 955 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3586: A.P. Black (Ed.), ECOOP 2005 - Object-Oriented Programming. XVII, 631 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005. (Subseries LNAI).

Vol. 3583: R.W. H. Lau, Q. Li, R. Cheung, W. Liu (Eds.), Advances in Web-Based Learning – ICWL 2005. XIV, 420 pages. 2005.

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005. (Subseries LNAI).

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Preface

After one year, the major actors in the field of cooperative design, visualization and engineering gathered together again by the side of the beautiful Mediterranean Sea to exchange the research and development experience in the field. CDVE 2005 served as a forum to promote the research in the field and it attracted great attention from the CDVE community. This year, we received contributions from over 100 authors, 5 continents and more than 20 countries.

As we can see, great progress in research and development has been achieved since the last conference. We received papers on cooperative design, cooperative visualization, cooperative engineering and other cooperative applications. As an important trend, the researchers have started to attack the problems in CDVE from a more generic base. We are happy to see contributions such as constraint maintenance, decision support, and security enforcement for CDVE. Case studies and application-specific developments are among the cooperative visualization papers. Along the line of cooperative engineering, knowledge management, reconfigurability, and concurrency control are major issues being addressed.

Cooperative working in design, visualization, engineering and other areas has different degrees of cooperation. I classify them as strong cooperation, intermediate cooperation, and light cooperation. Strong cooperation involves real-time multiple-user multiple-location modification to the same workspace. Light cooperation exists in the applications where the basic working relationship is only information or workspace sharing among the cooperative entities, no modification to the workspace is involved. Therefore, any application that is shared by more than one single user can be considered as a light-degree cooperative application. Any application between these two extremes can be considered as intermediately cooperative. Our conference addressed the common and specific issues of all of them.

I would like to express my thanks to the Program Committee and the numerous volunteer paper reviewers for their generous contribution to the conference which made it at a high-quality event. Without their support, the CDVE 2005 conference would not have been successful.

September 2005 Yuhua Luo

Organization

Conference Chair:	Professor Yuhua Luo University of Balearic Islands Spain

International Program Committee

Program Chair:	Professor Dieter Roller Institut für Rechnergestütz Ingenieursysteme University of Stuttgart Germany

Members

Katy Böner	Li Bohu	Weiming Shen
Susan Finger	Mikael Jern	Ram Sriram
Gu Ning	Bjørn Erik Munkvold	Chengzheng Sun
Kurt Kosanke	Zhang Heming	Jos P. van Leeuwen
Francis C.M. Lau	Mary Lou Maher	Carlos Vila
Moira C. Norrie	Benoît Otjacques	Nobuyoshi Yabuki

Table of Contents

Building a CSCW Infrastructure Utilizing an M&S Architecture and XML

Jenny Ulriksson[1], Farshad Moradi[1], Mattias Liljeström[2], and Nicholas Montgomerie-Neilson[2]

[1] Swedish Defense Research Agency, Department of Systems Modeling, 172 90 Stockholm, Sweden
{jenny.ulriksson, farshad.moradi}@foi.se
http://www.foi.se
[2] Royal Institute of Technology, Stockholm, Sweden

Abstract. CSCW applications provide virtual spaces for human-human cooperation. Within Modelling and Simulation (M&S), CSCW support is highly beneficial but, as in most single-user applications, not natively provided. External support, such as infrastructures for creating or adapting applications for collaborative work, is generally missing in the CSCW community. At the Swedish Defence Research Agency we study the possibilities of defence CSCW, for the primary intention of collaborative M&S. Since most of our M&S complies with the HLA (a distributed simulation architecture), HLA appeared as a candidate for building a CSCW infrastructure upon. Thus we have designed and developed a prototype of a CSCW infrastructure, based on a combination of HLA and XML. The goal is to provide a foundation for developing CSCW applications, and for adapting existing applications to collaborative work. This paper presents the design and development, and experiments conducted for verifying utilization of HLA and XML for the purpose.

1 Introduction and Motivation

CSCW *(Computer Supported Cooperative Work)* is a term often used for human collaboration in computer environments. CSCW enables people to share applications and tools, and at the same time communicate, to collaboratively perform or solve tasks. Hence virtual spaces for human-human cooperation can be created to support people and organizations to overcome geographical distances. The problem is that most single-user applications are specialized for their purpose, and do not naturally provide CSCW support. External support, such as infrastructures for developing CSCW software or adapting existing applications to collaborative work, is generally a missing part within the CSCW community.

At the Swedish Defense Research Agency (FOI) a project called NetSim conducts research activities concerning among other things the possibilities of collaborative M&S within the defense. The need for a CSCW infrastructure was identified within this work, and the lack of such existing ones determined. Consequently, the decision was made to develop our own. Most M&S within our research complies with an IEEE

Y. Luo (Ed.): CDVE 2005, LNCS 3675, pp. 1 – 13, 2005.

standardized simulation architecture called the *High Level Architecture* (HLA), and the architecture came to mind as a potential candidate technology also for CSCW. The technology requirements for building a CSCW infrastructure are many, and HLA not only provides some of them, but also offers additional valuable features. But the question that rises is: is HLA really suitable for use in distributed CSCW?

We propose an infrastructure for CSCW that is based on a combination of HLA and XML (*Extensible Markup Language*). The design has partly been implemented in a prototype called *Collaborative Core* (CC), which is based on a replicated distributed architecture. All information transmitted in the system is structured according to XML information models, and HLA services are used for transmission and other functions. Performance experiments have been conducted to verify both the use of XML for the purpose, and for verifying the combination of HLA and XML as the foundation of a CSCW infrastructure. The result confirmed the suitability of our proposal, but also revealed some HLA limitations that must be considered.

This paper presents the proposed design and the developed CC prototype. It also presents experiment results, our conclusions, some recommendations, and future work.

Clarifications for the reader: when further on discussing *nodes*, we mean client computer environments (i.e. PCs with a client using it). When referred to, one *node* hosts only one *client* (though in reality a node may host several clients). When discussing *applications*, what are intended are the applications used for collaboration (i.e. shared tools) within a *collaboration session,* and not the application providing CSCW services. A *collaboration session* is a session that transpires whenever two or more users jointly and concurrently perform tasks using interconnected software.

2 NetSim and HLA

At FOI we have performed research within network based M&S for several years. Recently we initiated activities concerning development of a common platform for defense M&S, and we believe CSCW support to be a very valuable service in such an environment. In this work we quickly identified the lack of such infrastructures that could be easily integrated with M&S applications and within the proposed common platform, and decided to develop our own.

2.1 The NetSim Project and CSCW

The NetSim project was initiated in 2003, at the Department of Systems Modeling at FOI. The primary project goal is to study, develop and modify methods and techniques for the purpose of network based M&S, and is described in [1]. One of the main activities aims at constructing an architecture, and develop a prototype, that can act as a common platform for defense M&S – the NetSim Platform. Apart from issues within distributed simulation such as fault-tolerance and distributed execution, one of the major project actions has concerned exploiting the potential of CSCW for defense applications. Since defense related operations engage actors which are often spread over long distances, cooperation is not easily achieved and the correct knowledge and support are not accessed without difficulty. A common defense platform that provides CSCW support may not replace real human-human cooperation, but can constitute an essential alternative.

2.2 Collaborative M&S Within NetSim

The area of M&S can substantially benefit from collaborative services, since CSCW can facilitate the provision of support that is required in the M&S process. Here it does not only make M&S expertise and knowledge more easily available, but also activities such as distance training and education more easily accessed. Moreover, the quality of M&S activities and products can be secured and controlled. The problem is that M&S applications are specialized for their purpose, and do not naturally provide support for collaborative services.

As a first attempt to address the problem, a simple prototype of a collaborative M&S environment was designed and developed [2]. This was based on JXTA[1] Peer-to-peer technology. The prototype was dismissed but gave us valuable experiences. As an example, CSCW support should not be based on an application specific approach, but rather constitute a more general infrastructure for various applications and purposes. More advanced support for time mechanisms were also needed. A final lesson was that it is preferable to use a more mature technology, one that supports more functionality than the one used. But we identified no infrastructures that suited above functionality. Various applications support some of the functions we desire, but none have been successful, and few support all of the desired functionality.

A general problem that developers of CSCW applications face is the complexity of integrating management of collaboration groups and activities within an application. As a result, the superior software for most computer-related professional tasks are single-user local desktop applications, and the dedicated CSCW software tend to lag behind in other than CSCW functionality [3]. Especially within domain specific applications, such as M&S applications, this kind of support is rarely seen. Thus we decided to address the problem through in-house design and development of the needed infrastructure, a foundation we call *Collaborative Core* (CC). CC will provide collaborative services, and should due to the nature of defense operations be distributed. CC is designed and has so far partly been implemented (see Section 3.1 for more detail). The development has been performed in close cooperation with two students from the Royal Institute of Technology [4, 5].

2.3 The High Level Architecture

Whilst parallel simulation aims at reducing the total simulation execution time (so called *speed-up*), the science of distributed simulation (DS) has different goals [6]. DS is the M&S answer to the Object Oriented thinking in the community of Computer Science. In DS, simulation components are executed on different computing nodes and coordinated in a joint simulation. This makes it possible for simulations too

[1] JXTA is a Java based, open-source Peer-to-peer framework distributed at http://www.jxta.org/ [accessed February 2005].

large for execution on one single computer to be distributed and executed on several nodes. Further it facilitates and makes it possible for different kinds of simulation components to interact, despite residing in different computer environments.

Above features with DS assume simulation component interoperability, namely that all components utilize a common standard. The *High Level Architecture* (HLA) is a framework for component-based simulation systems, originally developed within the US Defense Modeling and Simulation Office (DMSO) [7] and is the proposed simulation standard within the Swedish Defense. HLA provides a common communication structure, and rules for simulation components (so called *federates*) to follow, assuring them to be interoperable and able to together act in a distributed simulation (so called *federation*) [8]. The federation is managed and communication is transmitted through an implementation of the HLA services, a distributed operating system, called RTI (*Runtime Infrastructure*). RTI provides required services, such as:

- *Federation Management* – creating and controlling the federation execution etc.
- *Time Management* (TM) – flexible and advanced means of federation time management
- and *Data Distribution Management* (DDM) – mechanisms for efficient routing of information among federates

3 CC Design

When the CC infrastructure was developed, a concept was first designed and proposed. Thereafter technologies for implementation were chosen. The developed prototype was a partial implementation of the CC design, and is described in Chapter 4.

3.1 CC Infrastructure: The Concept

The CSCW service in the NetSim platform will be provided as a module that supports the user transparently with group management and tool sharing services. Applications using the services will do so without user specific complementary action. CC leverages support and services required for a user to be able to start, administrate, and participate in computer based collaboration groups, and hence also for sharing tools and other functions in NetSim. Besides, CC will offer development support for integrating new tools for CSCW. In general, the CC design is comprised by three main components, presented in Table 1.

Our research and development has focused on the two latter issues. Concerning the second component, the CC framework will provide a pluggable interface for collaborative applications, and that relieves tools of responsibility for managing user groups, and most of communication responsibility (discussed in Section 4.3). This way, developers will be supported in developing new CSCW tools, and in modification of existing applications to become collaborative. The third component represents straight-forward services for CSCW group management and administration, including shared group status and shared areas.

Table 1. The three CC main components, of which number 2 and 3 are treated in this paper

CC Service	Content
1. Communication Means	Integrated tools for communication
2. Application Interface	API, interface and software as development support for integrating new tools into CC
3. Group Management Services	Creating and maintaining collaboration groups, group administration etc.

3.2 Requirements and Design Choices

The overall requirements for CSCW in the NetSim platform, and consequently for the CC module, are described in Table 2. Considering these, some design choices were made. The last requirement lead us to choose a replicated architecture for the CC, i.e. all clients execute their own equal set of applications, and are themselves responsible for taking the correct action, to the retrieved changes from other clients. Replication was chosen since various client types may have different requirements in terms of GUI complexity and limited network connections etc. Transmitting only data, and letting the client specific application process it, is here a suitable solution.

The aim of the CC application interface is to provide support for plugging tools into the CC. There are two ways of accomplishing this, either by letting CC take all responsibility, or through requiring modification and responsibility from the application. Allowing no tool modifications at all requires a very generic CC interface, which is hardly possible. Thus we decided that CC can set some application requirements, with the primary aim to keep modifications at a minimum.

Table 2. Primary requirements for CC

Requirement	Significance
Distributed CSCW	The environment should be distributed. But this does not exclude future architecture combinations
Synchronous work	The CSCW intended here is immediate and synchronous
Short Persistence Collaboration Groups	Collaborative activities within NetSim are assumed to most often be directly task oriented, i.e. life times of collaboration sessions are assumed short
Small group sizes	Groups are assumed small, 2-8 persons. Considering scalability and regarding HLA as technology larger groups are possible. However, that requires social support to address virtual conflicts etc., an issue not handled here
Various client types	Different client types are considered, such as thin clients with poor network connections. In this regard, we assume virtual worlds to be too complex for the purpose

3.3 HLA for CSCW?

Since the acceptance as an IEEE standard (IEEE 1516), the area of application for HLA has been broadened to both non-military and non-simulation areas. An example is the multiplayer online gaming framework by Vuong et al. [9], and the collaborative virtual shopping mall by Zhao & Georganas [10]. The latter made an evaluation of the HLA as an enabler of collaborative virtual environments, with overall affirmative results [10].

Using an already existing architecture to build a CSCW infrastructure upon, may contribute to avoiding unnecessary development. HLA provides an infrastructure with essential services that beneficially could be used for CSCW, such as time management, group management (*Federation Management*), efficient information filtering (DDM), and communication management. Moreover it is a mature technology and standardized. An issue is that the HLA originally was not developed for real-time applications, something that CSCW applications highly are. Thus, the suitability of HLA for the purpose must be evaluated. And as stated, one of the challenging issues for CSCW is consistency management. Advanced, flexible consistency management has been declared a lacking part in current implementations of CSCW and in existing systems [11]. Utilizing HLA and the RTI, which provide advanced, flexible time management, this issue can be addressed appropriately. Moreover, since most simulation within our research complies with the HLA, we assumed it to be a candidate technology for our purpose.

3.4 XML for the Purpose?

XML (the *Extensible Modeling Language*) provides a way of structuring information in a platform independent, human-readable way. XML allows developers to create their own markup languages, and templates and rules (defined in *XML Schemas*), that help assuring the interoperability between data. A very beneficial feature is that XML efficiently separates data from presentation. In a CSCW infrastructure such as described above a lot of information is managed, such as tool specific information, collaboration group information and client information. A design choice could be to strictly follow the technology chosen (here HLA), but to accomplish a more generic structure and less technology dependent, XML can be used for structuring and handling information. Another reason for using XML is that here various client types are expected. Using XML, the same information is provided to all participants. At the client side, parsing of the XML formatted information can allow for user specific utilization and presentation.

3.5 Combining HLA and XML for CSCW

Concluding the above discussion and our experiences from earlier work, we propose a combination of HLA and XML to constitute the foundation for a CSCW infrastructure. It uses XML-based group definitions and a communication infrastructure built on the HLA.

4 Implementation

4.1 The Prototype

A first CC prototype has been implemented, of which the two main components were implemented in separate, but closely coordinated tracks. They are described in