Klaus-Dieter Althoff Andreas Dengel
Ralph Bergmann Markus Nick
Thomas Roth-Berghofer (Eds.)

LNAI 3782

Professional Knowledge Management

Third Biennial Conference, WM 2005
Kaiserslautern, Germany, April 2005
Revised Selected Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Klaus-Dieter Althoff
Universität Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
E-mail: althoff@iis.uni-hildesheim.de

Andreas Dengel
Thomas Roth-Berghofer
Deutsches Forschungszentrum für Künstliche Intelligenz DFKI GmbH
Erwin-Schrödinger-Straße 57, 67663 Kaiserslautern, Germany
E-mail: {andreas.dengel,thomas.roth-berghofer}@dfki.de

Ralph Bergmann
Universität Trier, 54286 Trier, Germany
E-mail: bergmann@uni-trier.de

Markus Nick
Fraunhofer Institut Experimentelles Software Engineering IESE
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
E-mail: markus.nick@iese.fraunhofer.de

Library of Congress Control Number: 2005937929

CR Subject Classification (1998): I.2, H.4, H.3, J.1, K.6, K.3-4

ISSN	0302-9743
ISBN-10	3-540-30465-7 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-30465-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11590019 06/3142 5 4 3 2 1 0

Preface

Professional knowledge management is imperative for the success of enterprises. One decisive factor for the success of knowledge management projects is the coordination of elements such as corporate culture, enterprise organization, human resource management, as well as information and communication technology. The proper alignment and balancing of these factors are currently little understood—especially the role of information technology, which is often regarded only as an implementation tool, though it can be a catalyst by making new knowledge management solutions possible.

This conference brought together representatives from practical and research fields for discussing experiences, professional applications, and visions through presentations, workshops, tutorials, and an accompanying industry exhibition. The main focus of the conference was the realization of knowledge management strategies with the aid of innovative information technology solutions, such as intelligent access to organizational memories, or integration of business processes and knowledge management. Also of interest were holistic/integrative approaches to knowledge management that deal with issues raised by the integration of people, organizations, and information technology.

The third conference on "Professional Knowledge Management—Experiences and Visions" (WM 2005) in Kaiserslautern continued the success of the former conferences in Baden-Baden (WM 2001) and Lucerne (WM 2003). It was organized by the German Informatics Society (*Gesellschaft für Informatik GI e.V.*), especially by its Artificial Intelligence and Business Informatics division and its regional subsection (*GI Regionalverband Saar-Pfalz*), together with the German Society for Knowledge Management (*Gesellschaft für Wissensmanagement e.V.*) with support by the German Research Center for Artificial Intelligence DFKI GmbH, the Fraunhofer Institute for Experimental Software Engineering IESE, the University of Hildesheim, the University of Kaiserslautern, the University of Trier, FZI Karlsruhe, and conference consulting Harms. This conference could not have been such a success without the generous support of the sponsoring organizations Klaus Tschirra Stiftung, empolis GmbH, Filenet GmbH, Mindjet GmbH, ontoprise GmbH, Pylon AG, SAP AG, and Siemens AG.

These post-conference proceedings contain a selection of the best papers submitted to and presented at the conference. The papers have been carefully selected by the Workshop Co-chairs/organizers and the participants. The authors revised their original conference submissions according to feedback and results from the discussions.

We would like to thank the authors of the papers included in this book, the Workshop Co-chairs/organizers, and the Program Committee members for their additional work on these post-conference proceedings. We thank the keynote speakers of the conference, the tutorial authors, the organizations explaining their products and services, as well as all the presenters. We also appreciate very much all the effort invested by the local organization team and the various members of our research teams involved in this conference. They all contributed considerably to the success of WM 2005.

Last but not least we would like to thank Rudi Studer, Steffen Staab, Ulrich Reimer, Gerd Stumme, Andreas Abecker, their respective organizing teams of WM 2001 and WM 2003, the management board of the GI Special Interest Group on Knowledge Management (*GI-Fachgruppe Wissensmanagement FGWM*), and the *Gesellschaft für Wissensmanagement* for starting this conference series with much enthusiasm, courage, and foresight.

September 2005

Klaus-Dieter Althoff
Andreas Dengel
Ralph Bergmann
Markus Nick
Thomas Roth-Berghofer

Organization

The conference was organized by the following societies and groups:

Gesellschaft für Informatik e. V.:

- FG Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen
- FG Computer Supported Cooperative Work
- FG Information Retrieval
- FG Management Support Systems
- FG Vorgehensmodelle für die betriebliche Anwendungsentwicklung
- FG Wissensmanagement
- AK Grundlagen der Modellierung und Ausführung von Workflows
- AK Philosophie und Informatik
- AK Wissensmanagement in der Praxis
- GI-Regionalgruppe Saar-Pfalz

Gesellschaft für Wissensmanagement e. V.

Conference Chairs

Klaus-Dieter Althoff (University of Hildesheim)
Andreas Dengel (DFKI GmbH, Kaiserslautern)

Workshop Chair

Ralph Bergmann (University of Trier)

Tutorial Chair

Andreas Abecker (FZI, Karlsruhe)

Conference Management

Christine Harms (conference consulting Harms, Bonn)
Markus Nick (Fraunhofer IESE, Kaiserslautern)
Thomas Roth-Berghofer (DFKI / University of Kaiserslautern)

Keynote Talks

Irma Becerra-Fernandez (Florida International University, USA)
Andreas Günter (HITec e. V., Hamburg, Germany)
Larry Kerschberg (George Mason University, USA)
Klaus Kornwachs (Brandenburg University of Technology, Cottbus, Germany)
Naoyuki Nomura (Ricoh, Japan)

Conference Sponsors

Klaus Tschirra Stiftung	www.kts.villa-bosch.de
empolis GmbH	www.empolis.de
Filenet GmbH	www.filenet.de
Mindjet GmbH	www.mindjet.de
ontoprise GmbH	www.ontoprise.de
Pylon AG	www.pylon.de
SAP AG	www.sap.com
Siemens AG	www.siemens.de

Press and Media Partners

DoQ	www.doq.de
interim 2000	www.interim2000.de
Voice of Information	www.voi.de
Wissensmanagement	www.wissensmanagement.net

Tutorials

Industrielles Wissensmanagement mit Fallbasiertem Schließen
Ralph Bergmann (University of Trier, Germany)
Mehmet Göker (PricewaterhouseCoopers, San Jose, USA)

Systematische Nutzung von Wissen in Geschäftsprozessen - Modellierung und Analyse wissensintensiver Geschäftsprozesse
Norbert Gronau (University of Potsdam, Germany)
Claudia Müller (University of Potsdam, Germany)
Julian Bahrs (University of Potsdam, Germany)

Referenzmodellgestütztes Wissensmanagement - Wissensorientierte Analyse, Gestaltung und Unterstützung von Geschäftsprozessen in kleinen und mittleren Unternehmen
Thomas Mühlbradt (GOM Gesellschaft für Organisationsentwicklung und Mediengestaltung mbH, Aachen, Germany)

Methoden zur engen Integration von Arbeiten und Lernen - Ansätze zur Integration von Wissensmanagement und eLearning
Stefanie Lindstaedt (Know-Center Graz, Austria)
Tobias Ley (Know-Center Graz, Austria)

Ontologien und Semantic Web Technologien für das Wissensmanagement
York Sure (University of Karlsruhe, Germany)
Hans Peter Schnurr (Ontoprise GmbH, Karlsruhe, Germany)

Tutorials (contd.)

Wissensmanagementsysteme
Ronald Maier (Martin Luther University, Halle-Wittenberg, Germany)
Thomas Hädrich (Martin Luther University, Halle-Wittenberg, Germany)

Praktische Erfahrungen bei der Wissensbewertung und Wissensbilanzierung in Unternehmen
Claus Nagel (EBIS GmbH, Hof, Germany)

Human Resource Management und Wissensmanagement
Ernst Biesalski (DaimlerChrysler, Wörth, Germany)
Andreas Abecker (FZI, Karlsruhe, Germany)

Workshop Co-chairs/Organizers

Sven Abels (University of Oldenburg, Germany)
Thomas Allweyer (University of Applied Sciences Kaiserslautern, Zweibrücken, Germany)
Susanne von Baeckmann (Denken & Wissen, Munich, Germany)
Brigitte Bartsch-Spörl (BSR Consulting GmbH, Munich, Germany)
Ansgar Bernardi (DFKI GmbH, Kaiserslautern, Germany)
Andreas Birk (sd&m, Germany)
Gregor Büchel (University of Applied Sciences Cologne, Germany)
Heike Dalinghaus (University of Osnabrück, Germany)
Björn Decker (Fraunhofer IESE, Kaiserslautern, Germany)
Torgeir Dingsøyr (SINTEF ICT, Norway)
Ludger van Elst (DFKI GmbH, Kaiserslautern, Germany)
Hiromichi Fujisawa (Hitachi, Japan)
Markus Gelhoet (University of Osnabrück, Germany)
Norbert Gronau (University of Potsdam, Germany)
Axel Hahn (University of Oldenburg, Germany)
Liane Haak (University of Oldenburg, Germany)
Harald Holz (DFKI GmbH, Kaiserslautern, Germany)
Hisashi Ikeda (Hitachi, Japan)
Larry Kerschberg (George Mason University, USA)
Bertin Klein (DFKI GmbH, Kaiserslautern, Germany)
Jan Kuhlmann (Attorney-at-Law, Hamburg, Germany)
Edith Maier (Danube University Krems, Austria)
Heiko Maus (DFKI GmbH, Kaiserslautern, Germany)
Kerstin Maximini (University of Trier, Germany)
Martin Memmel (DFKI GmbH/ University of Kaiserslautern, Germany)
Marita Muscholl (formerly Sergl) (University of Trier, Germany)
Dirk Muthig (Fraunhofer IESE, Kaiserslautern, Germany)
Naoyuki Nomura (Ricoh Company Ltd., Bunkyo-ku, Tokyo, Japan)

Workshop Co-chairs/Organizers (contd.)

Klaus North (University of Applied Sciences Wiesbaden, Germany)
Peter Pawlowsky (University of Chemnitz, Germany)
Dirk Ramhorst (Siemens Business Services, Germany)
Eric Ras (Fraunhofer IESE, Kaiserslautern, Germany)
Ulrich Reimer (Business Operation Systems, Switzerland)
Bodo Rieger (University of Osnabrück, Germany)
Martin Schaaf (University of Hildesheim, Germany)
Steffen Staab (University of Koblenz-Landau, Koblenz, Germany)
Armin Stahl (DFKI GmbH/ University of Kaiserslautern, Germany)
Ralph Traphöner (empolis GmbH, Kaiserslautern, Germany)
Stephan Weibelzahl (National College of Ireland, Dublin, Ireland)
Oliver W. Wendel (PYLON AG, Frankfurt / Main, Germany)
Marcus Willamowski (Latham & Watkins LLP, Hamburg, Germany)
Michael Wolters (University of Osnabrück, Germany)

Program Committee

Pekka Abrahamsson (VTT Electronics, Finland)
Gerhard Ackermann (Siemens, Austria)
John D'Ambra (UNSW, Australia)
Fulvio D'Antonio (IASI, National Research Council (CNR), Italy)
Dimitris Apostolou (Planet Ernst & Young, Greece)
Aybuke Aurum (University of New South Wales, Australia)
Gabriela Avram (Centre de Recherche Public Henri Tudor, Luxembourg)
Yukika Awazu (The Engaged Enterprise, USA)
Steven Bashford (Business Solutions Group, Mindjet GmbH, Germany)
Stephan Baumann (DFKI GmbH, Kaiserslautern, Germany)
Irma Becerra-Fernandez (Florida International University, Miami, USA)
Wolfgang Behme (Continental AG, Hannover, Germany)
Katy Börner (Indiana University, Bloomington, USA)
Frank Bomarius (University of Applied Sciences Kaiserslautern/Fraunhofer IESE, Germany)
Matteo Bonifacio (University of Trento, Italy)
Paolo Bouquet (University of Trento, Italy)
Peter Brössler (msg systems AG, Munich, Germany)
Peter Chamoni (Universität Duisburg-Essen, Duisburg, Germany)
Jürgen Cleve (University of Wismar, Germany)
Hans Czap (University of Trier, Germany)
Jürgen Dammert (Robert Bosch GmbH, Stuttgart, Germany)
Jörg Denzinger (University of Calgary, Canada)
Kevin C. Desouza (University of Illinois, USA)
Virginia Dignum (University of Utrecht, The Netherlands)
Georg Disterer (University of Applied Sciences and Arts Hannover, Germany)

Program Committee (contd.)

Jörg Eckardt (Maria Hilf GmbH, Dernbach, Germany)
Dieter Ehrenberg (University of Leipzig, Germany)
Jörn Erbguth (Juris GmbH, Germany)
Walter Fabbri (Coop, Italy)
Raimund Feldmann (Fraunhofer USA Center for Experimental Software Engineering, USA)
Luciano Floridi (University of Oxford, United Kingdom)
Ulrich Frank (University of Koblenz, Germany)
Tessen Freund (altavier GmbH, Berlin, Germany)
Beatrice Fuchs (University Lyon III, France)
Peter Funk (Mälardalen University, Sweden)
Roland Gabriel (University of Bochum, Germany)
Peter Gluchowski (University of Düsseldorf, Germany)
Mehmet Göker (PricewaterhouseCoopers, San Jose, USA)
Reiner Gratzfeld (Henkel KGaA, Düsseldorf, Germany)
Alexander Greisle (FhG IAO Competence Center New Work, Stuttgart, Germany)
Pierre Grenon (Centre for Ontology, Geneva, Switzerland)
Gunter Grieser (University of Darmstadt, Germany)
Dieuwke de Haan (Technical University of Eindhoven, The Netherlands)
Ruth Hagengruber (University of Koblenz-Landau, Koblenz, Germany)
Axel Hahn (University of Oldenburg, Germany)
John D. Haynes (University of Central Florida, USA)
Kai U. Heitmann (University of Cologne, Germany)
Scott Henninger (University of Nebraska-Lincoln, USA)
Knut Hinkelmann (University of Applied Sciences Solothurn, Switzerland)
Josef Hofer-Alfeis (Siemens, Munich, Germany)
Wilhelm Hummeltenberg (Universität Hamburg, Germany)
Jozef Hvorecky (College of Management, Bratislava, Slovakia)
Ioannis Iglezakis (DaimlerChrysler, Ulm, Germany)
Josef Ingenerf (University of Lübeck, Germany)
Hayo Iversen (Jurion GmbH, Germany)
Makoto Iwayama (Hitachi, Kokubunji, Japan)
M. Letizia Jaccheri (Norwegian University of Science and Technology)
Norbert Jastroch (MET Communications GmbH, Bad Homburg, Germany)
Manfred Jeusfeld (Tilburg University, The Netherlands)
David Jonassen (University of Missouri-Columbia, USA)
Bernd-Ulrich Kaiser (Bayer AG Leverkusen, Germany)
Noriko Kando (National Institute of Informatics, Chiyoda-ku, Japan)
Dimitris Karagiannis (University of Vienna, Austria)
Hans-Georg Kemper (University of Stuttgart, Germany)
Stefan Kirn (University of Hohenheim, Germany)
Koichi Kise (Osaka Prefecture University, Japan)
Matthias Klusch (DFKI GmbH, Saarbrücken, Germany)

Program Committee (contd.)

Rob Koper (Open University of the Netherlands, The Netherlands)
Klaus Kornwachs (Brandenburg University of Technology, Cottbus, Germany)
Hermann Krallmann (TU Berlin, Germany)
Dirk Krechel (SER Solutions GmbH, Germany)
Stefanie Kreis (Freshfields Bruckhaus Deringer, Germany)
Franz Kurfeß (CalPoly, San Luis Obispo, USA)
Steffen Lange (University of Applied Sciences Darmstadt, Germany)
Manfred Langen (Siemens, Munich, Germany)
Ulrike Lechner (University of the Federal Armed Forces, Munich, Germany)
Peter Lehmann (Stuttgart Media University of Applied Sciences, Germany)
Franz Lehner (University of Passau, Germany)
Stefanie Lindstaedt (Know-Center Graz, Austria)
Jørgen Madsen (Oracle, Denmark)
Edith Maier (Danube University Krems, Austria)
Ronald Maier (Martin Luther University, Halle-Wittenberg, Germany)
Simone Marinai (University of Florence, Italy)
Wolfgang Martin (Wolfgang Martin Team, Annecy, France)
Eitel von Maur (University of St. Gallen, Switzerland)
Frank Maurer (University of Calgary, Canada)
Hermann Maurer (University of Graz, Austria)
Grigori Melnik (University of Calgary, Canada)
Mirjam Minor (Humboldt University Berlin, Germany)
Harry Mucksch (IT-Beratung und Services, Apen, Germany)
Heinz-Jürgen Müller (University of Cooperative Education Mannheim, Germany)
Klaus Müller (Robert Bosch GmbH, Germany)
Wolfgang Nejdl (University of Hannover, Germany)
Anja Neubauer (Cognilexus Knowledge Management Consultants, Germany)
Markus Nick (Fraunhofer IESE, Kaiserslautern, Germany)
Takahiko Nomura (Fuji Xerox Co. Ltd., Japan)
Dietmar Pfahl (Fraunhofer IESE, Kaiserslautern, Germany)
Sofia Pinto (University of Lisbon, Portugal)
Klaus Pommerening (University Hospital of Mainz, Germany)
Steve Probert (Cranfield University, Swindon, United Kingdom)
Thomas Reinartz (DaimlerChrysler, Ulm, Germany)
Michael M. Richter (University of Kaiserslautern, Germany)
Bodo Rieger (University of Osnabrück, Germany)
Knut-Helge Ronæs Rolland (Norwegian University of Science and Technology)
Francis Rousseaux (University of Reims and IRCAM Paris, France)

Program Committee (contd.)

Emil Røyrvik (SINTEF Technology Management, Norway)
Günther Ruhe (University of Calgary, Canada)
Alexander Schieffer (Center of Excellence for Leadership and Learning GmbH, Munich, Germany)
Rainer Schmidt (University of Rostock, Germany)
Sascha Schmitt (SAP AG, Germany)
Georg J. Schneider (University of Applied Sciences Trier, Germany)
Kurt Schneider (University of Hannover, Germany)
Detlev Schoder (University of Cologne, Germany)
Andy Schürr (University of Darmstadt, Germany)
Stefan Schulz (TU Berlin, Germany)
Barry Smith (University at Buffalo, USA)
Rini van Solingen (LogicaCMG, The Netherlands)
Marcus Spies (University of Munich, Germany)
Heiner Stuckenschmidt (Free University Amsterdam, The Netherlands)
Jürgen Taeger (University of Oldenburg, Germany)
Carsten Tautz (Mondia, Dubai, United Arab Emirates)
Rainer Telesko (Robert Bosch GmbH Stuttgart, Germany)
Oliver Thomas (DFKI GmbH, Saarbrücken, Germany)
Huaglory Tianfield (Glasgow Caledonian University (GCU), United Kingdom)
Klaus Tochtermann (Know-Center Graz, Austria)
Bidjan Tschaitschian (empolis GmbH, Germany)
Wolfgang Uhr (TU Dresden, Germany)
Thomas Uthmann (University of Mainz, Germany)
Daniel Veit (University of Karlsruhe, Germany)
Ivo Vollrath (SAP AG, Germany)
Oliver Vopel (Ernst & Young, United Kingdom)
Holger Wache (Free University Amsterdam, The Netherlands)
Aldo von Wangenheim (Federal University of Santa Catarina, Brazil)
Patrick Waterson (Fraunhofer IESE, Kaiserslautern, Germany)
Rosina Weber (Drexel University, USA)
Mathias Weber (BITKOM e. V. Berlin, Germany)
Stefan Wess (empolis GmbH, Germany)
Boris Wyssusek (Queensland University of Technology, Australia)
Hideo Yamazaki (NRI, Japan)
Sandra Zilles (DFKI GmbH/University of Kaiserslautern, Germany)

Table of Contents

Keynote Contributions

Intelligent Office Appliances (IOA)

Learning Software Organizations (LSO)

Peer-to-Peer and Agent Infrastructures for Knowledge Management (PAIKM)

Knowledge Intensive Business Processes (KiBP)

Knowledge Management and Business Intelligence (KMBI)

Intelligent IT Tools for Knowledge Management Systems: (KMTOOLS 2005)

Second Workshop on Knowledge Management for Distributed Agile Processes: Models, Techniques, and Infrastructure (KMDAP 2005)

Harald Holz[1], Heiko Maus[1], Naoyuki Nomura[2], and Martin Schaaf[3]

[1] Knowledge Management Department,
German Research Center for Artificial Intelligence DFKI GmbH, Germany
{Harald.Holz, Heiko.Maus}@dfki.de
[2] Software R&D Group,
Ricoh Company Ltd., Japan
naoyuki.nomura@nts.ricoh.co.jp
[3] Data and Knowledge Management Group,
University of Hildesheim, Germany
schaaf@dwm.uni-hildesheim.de

Traditional process-oriented knowledge management (KM) approaches are inadequate for highly dynamic and volatile processes, whose steps cannot be planned in advance, and during which new, unanticipated "knowledge needs" frequently arise: such processes handle mostly informal documents and rely on face-to-face communication between participants. During the workshop, practitioners and researchers presented novel approaches for decision support within agile processes, agile process modeling, and communities of practice.

Decision Support for Agile Processes. In order to support agility in the realm of project management, Karni and Kaner propose a methodology for decision making within project management processes based on case-based reasoning (CBR).

Weber and Wild demonstrate the integration of workflow systems and conversational CBR to enable workflow participants in exceptional situations to change the current workflow instance according to their needs. The problem solving is then retained as case for supporting other participants in similar situations.

Freßmann et al. present a workflow environment for supporting persons and teams in agile processes also using a CBR approach but here with the focus on mobile teams in pressing situations requiring advice and context-dependent information support.

Agile Process Modeling. Zacarias et al. propose to model relations between actors involved in complex processes with action and interaction contexts. The goal is to provide better modeling means for analyzing and finally support people in knowledge-intensive business processes.

Fenstermacher discusses the notion of agility for process-oriented KM systems and presents some next steps towards supporting knowledge workers in agile

K.-D. Althoff et al. (Eds.): WM 2005, LNAI 3782, pp. 398–399, 2005.

processes. He argues that before studying agile processes, means are required for observing user activities and analyze it for a process-based use.

Communities of Practice. Bellini et al. describe an experiment on the impact of the educational background of developers on the knowledge sharing during the design phase. The study offers evidence that forming pairs with the same educational background emphasizes the expected benefits, in contrast to pairs from different educational backgrounds.

Program Committee

- Klaus-Dieter Althoff, University of Hildesheim, Germany
- Dimitris Apostolou, Planet Ernst & Young, Greece
- Torgeir Dingsøyr, SINTEF ICT, Norway
- Ludger van Elst, DFKI GmbH, Germany
- Knut Hinkelmann, University of Applied Sciences Solothurn, Switzerland
- Frank Maurer, University of Calgary, Canada
- Heinz-Jürgen Müller, Berufsakademie Mannheim, Germany
- Dirk Ramhorst, Siemens Business Services, Germany
- Steffen Staab, Karlsruhe University, Germany
- Heiner Stuckenschmidt, Vrije Universiteit Amsterdam, The Netherlands
- Oliver Thomas, Institute for Information Systems (IWi), Germany
- Klaus Tochtermann, Know-Center Graz, Austria
- Bidjan Tschaitschian, empolis GmbH, Germany
- Rosina Weber, Drexel University, USA
- Oliver Wendel, Pylon, Germany
- Hideo Yamazaki, NRI, Japan

The Impact of Educational Background on Design Knowledge Sharing During Pair Programming: An Empirical Study

Emilio Bellini[1], Gerardo Canfora[1], Aniello Cimitile[1], Felix Garcia[2], Mario Piattini[2], and Corrado Aaron Visaggio[1]

[1] RCOST- Research Centre on Software Technology, University of Sannio, Italy
{bellini, canfora, cimitile, visaggio}@unisannio.it
[2] Alarcos Research Group, University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
{Felix.Garcia, Mario.Piattini}@uclm.es
http://alarcos.inf-cr.uclm.es/english/

Abstract. The management of knowledge in software processes is becoming a challenging concern for researchers and practitioners. Explicit knowledge can be formalized in many kinds of documents and rules, and consequently transferred in a number of manners. On the contrary, tacit knowledge cannot be formalized, because it is mainly retained in personal cognitive models and consists of individual capabilities of dealing with problems. The design of software systems requires a consistent deployment of tacit knowledge, and pair programming has shown great promises for helping to share knowledge between programmers. It is a common experience that programmers come not only from computer science and engineering curricula, but also from other education degrees, such as mathematics, natural sciences, and social sciences. In this case they attend proper specialist post graduation courses. We have executed an experiment in order to verify the relationship between educational background of pair's components and knowledge sharing throughout working in pairs while designing software systems.

1 Introduction

The management of knowledge when producing and maintaining software, both in the explicit and tacit form [1], is assuming a significant role in software development. Software processes require capabilities which are both technical (e.g., programming languages, network, database, and operating systems), and attitudinal (e.g., problem solving, decision making). Methods for measuring the former exist, generally by listing the skills and the related experience; current literature reports several attempts in this direction [2, 3]. The latter is harder to describe, especially from the perspective of either a project manager or a developer. The two forms of knowledge are commonly classified as explicit and tacit. Explicit knowledge can be formalized and transferred. Tacit knowledge regards the individual capability of solving problem, and it can be built basically *by doing* [5, 6, 7, 8]: applying explicit knowledge, registering personal observations, and making personal models for retaining it (*interiorization* in

K.-D. Althoff et al. (Eds.): WM 2005, LNAI 3782, pp. 455–465, 2005.

SEKI Model [4]). This kind of knowledge cannot be easily formalized and transferred, except for the dialogue. Software design asks for a continuous and significant application of tacit knowledge, due to the high abstraction that characterizes software design: as matter of fact, a software programmer becomes a software designer only after some years of practice.

A technique that helps sharing the knowledge is pair programming [10]; it is a practice of extreme programming [9], where two programmers, working side by side, develop and contemporarily review the same piece of code. In the past few years, due to the skill shortage, a scenario has become frequent in Italian software industry: not only computer scientists or computer engineers are called to develop software, but also graduates with different educational profiles (physics, mathematics, engineering, economics) after a period of adequate training. The educational background of the pair's components can affect knowledge transfer within the pair. In order to validate this idea, an experiment has been designed and executed, specially focusing on the design phase rather than on pure coding.

Similarly to pair programming, we name "pair designing" the practice where two designers work side by side at the same design document; one of the two actively edits the document whereas the second performs continuous review.

The research goal of this paper is: *to analyze a pair designing task with the purpose of evaluating how educational background affects the knowledge building within the pair from the viewpoint of the designer, in the context of an actual post-graduate student project.*

The paper continues as follows. Section 2 discusses related work. Section 3 describes the experiment, while results are discussed in Section 4; Section 5 discusses the experimental threats. Finally, conclusions are drawn in Section 6.

2 Related Work

When the term 'pair programming' was not yet widespread, Nosek investigated *Collaborative Programming* [13]. Nosek executed an experiment with experienced programmers and it showed that collaborative programmers outperformed the individual programmers. Initially the attention of researchers has focussed mainly on quality and productivity, as in [14, 15, 20]. Recently, the target of pair programming investigation is turning to learning and knowledge transfer [21]. Williams and Kessler [11] found that pair programming fosters knowledge leveraging between the two programmers, particularly tacit knowledge. In [16], authors investigate, throughout an experiment, which are the knowledge needs to be addressed in order to implement effectively pair programming, when pair's components are distributed. Williams and Upchurch [17] examine the ways pair programming may enhance teaching and learning in computer science education. Some authors [18, 19] investigated the effects of pair programming on student performance in an introductory programming class. The main concerns emerging from the state of the art are two: first, the greatest part of the study involves students rather than professionals; second, the analyses accomplished are mainly qualitative instead of quantitative. To our best knowledge, however, no study has been published that addresses pair designing.

The current work is part of a family of experiments, aiming at evaluating the relationship between the practice of working in pairs applied to any phase of software process and knowledge building about the 'big picture' of the system. The results of the first experiment about the relationship between pair designing and knowledge leveraging were discussed in [12]. Two main outcomes were obtained. First, along all the experiment subjects who worked in pair showed a greater knowledge with respect to those who worked as singletons. Second, the knowledge building was more stable for pairs than for singletons: the knowledge growth of pairs can be predictable and repeatable within certain limits.

3 Experiment Description

The experiment was executed with the purpose of testing the following null hypothesis:

H_0: the difference in education between the pair's components does not affect the building of system knowledge realized by the pair's components.

The alternative hypothesis is:

H_1: the difference in education between the pair's components affects the building of system knowledge realized by the pair's components.

Subjects. The experiment was executed with students of the Master of Technologies of Software (MUTS) and Master of Management and Technologies of Software (MUTEGS), high education university courses for post-graduates, at University of Sannio (http://www.ing.unisannio.it/master/). Students of MUTS own a scientific graduation (engineering, mathematics, physics), whereas students of MUTEGS own an economic/humanistic graduation (economics, philology, literature, philosophy). Both the courses provide the same basic education in computer engineering (operating systems, programming languages, network, database, and software engineering), but MUTS students are specialized for developing and maintaining Software Systems, whereas MUTEGS students are trained for dealing with the economic and organizational issues of software lifecycle. The two Master courses are held contemporarily and both last one year, during which students attend theoretical classes and lab sessions, with the same professors and lecturers, develop a large and complex project in connection with an enterprise, participate to seminaries from international experts, perform a three month stage in software companies.

The subjects were organized as follows:

- 4 couples with one MUTS student and one MUTEGS student;
- 5 couples with two MUTS students;
- 5 couples with two MUTEGS students;
- the other 16 subjects, MUTS and MUTEGS, worked as solo designers.

All the groups were formed randomly.

Variables. System's knowledge represented the dependent variable and it was evaluated by grading a questionnaire, answered by subjects after having performed a maintenance task on the system. The questionnaires were evaluated in this way: each correct answer was evaluated 1; each incorrect answer was evaluated 0. The independent variables were the kinds of paired classes of subject's graduation: scientific with scientific, non-scientific with non-scientific, and scientific with non-scientific.

Rationale for the Sampling of the Population. Students of Software Engineering courses are suitable for such an experiment because they study software architecture and software system design. Furthermore they usually are employed as software architects or designers after the graduation. MUTS and MUTEGS students are a fine population's sample, considered that they experienced an actual project work during the overall master. Since the students have comparable curricula, there is not a relevant bias in the sample.

Assignment. In order to evaluate the knowledge built by doing while working on the system, the assignment for the subjects consisted of improving the design of the system. The design of the system was formalized in UML and included: textual specification of the system's requirements, two use cases diagrams, and two class diagrams (for a total of 15 classes). The design was developed by experimenters. Considered the time available, we preferred to avoid bulky documentation. The maintenance tasks were basically two: (i) reduce complexity, by erasing entities or relationships between entities not fundamental for understanding; and (ii) improve readability, by changing existing entities (use cases, actors, classes, methods), or adding new ones.

This kind of assignment was targeted at maximizing the knowledge built by doing; as matter of fact, maintenance needs the programmer to analyze in depth the system. The system design was realized by taking into account the knowledge of subjects, with the aim of making objects representative of the population. An excerpt of the documentation is provided in the appendix.

The Process. The process of the experimental run was the following:

- each subject studied documentation for 30 minutes, individually;
- each subject answered an entry questionnaire, individually, for about 15 minutes. The entry questionnaire was aimed at establishing the baseline, i.e. level of knowledge of the system before working on it;
- the pairs and the solo designers performed the maintenance tasks for 2 hours;
- each subject answered an exit questionnaire individually, in order to understand the knowledge built by practicing the maintenance in the two different ways, pair and solo.

Questionnaires. We prepared two questionnaires, QA and QB, in order to measure the dependent variable. Both the questionnaires were distributed as entry and exit questionnaire, so that each subject had randomly QA (or QB) at entry and, conversely, QB (or QA) at exit. This avoided that the results depended on the questionnaire itself. The questions concerned architectural and functional aspects of the system. One of the two questionnaires is shown in the appendix.

Although we would have liked to use a CASE tool, such as Rational Rose, or ArgoUML, we finally decided to use only pen and paper. The reason was that some subjects could be more familiar with this kind of tools and this could inject bias in the results. As consequence, we would have needed more time for preparation in order to equalize the ability of subjects to work with tools. In Table 1 the experimental design is provided.

Table 1. Experimental Design

Subjects	Treatment	Input	Output
4MUTS 4MUTEGS	Paired MUTS MUTEGS	Requirement Specification; Use case Diagram; Class Diagram; Entry questionnaire QA (or QB); Exit questionnaire QB (or QA).	Modifications to Use Case Diagram and Class Diagram; Answered entry questionnaire QA (or QB); Answered exit questionnaire QB (or QA).
5MUTEGS 5MUTEGS	Paired MUTEGS MUTEGS		
5MUTS 5MUTS	Paired MUTS MUTS		
8MUTS	Solo		
8MUTEGS	Solo		

4 Preliminary Results

Table 2 shows the descriptive statistics of data. It appears that the pairs MUTS-MUTS obtained results better than all the other subjects. As matter of fact, the average is the greatest one, as well as the maximum and the minimum values. The pairs MUTS-MUTEGS obtained an average value close to that of the pairs MUTEGS-MUTEGS. This suggests that forming pairs with professionals having different background does not give benefits with respect to the pairs formed by people with 'non-scientific' background.

Table 2. Descriptive Statics

Subjects	Avg	Standard Deviation	Max	min	moda
MUTS MUTS (pairs)	5,8	1,75	9	4	4
MUTS MUTEGS (pairs)	4	0,76	5	3	4
MUTEGS MUTEGS (pairs)	3,9	1,59	7	1	3
MUTS (solo)	4,3	1,04	6	3	4
MUTEGS (solo)	5,1	1,55	7	3	4

The pairs MUTS-MUTS exhibit the greatest variation in results; but also MUTEGS-MUTEGS show high value of standard deviation. This suggests that by coupling individuals with the same background the individual potentialities can be stimulated more than by coupling subjects with different background.

A more surprising result is that solo MUTEGS performed the closest average to the pairs MUTS-MUTS (only 12% less) and anyway higher than the solo MUTS (about 21% more). This can be explained by considering that in the group of solo MUTEGS a few strong individualities were present: this appears evident form the high variance (the std dev = 1,55) and from individual results plotted in Fig.1.

The null hypothesis can be rejected with statistical significance, by fixing the p-level at 0.05. We executed Mann Whitney tests, as the samples' data had no normal distribution. In Table 3 each row denotes one of the tests we made. For example, row 1 shows the results of exit questionnaire of subjects working in pairs MUTS-MUTS compared with those of subjects working in pairs MUTS- MUTEGS; whereas row 6 shows the results of questionnaire QA compared with the results of questionnaire QB.

Table 3. Results of Statistical Tests

Tests	**Rank Sum (a)**	**Rank Sum (b)**	**p-level**	**p-level 1 tail**
MUTS-MUTS (a) MUTS-MUTEGS (b)	121,00	50,00	**0,020**	**0,020**
MUTS-MUTS (a) MUTEGS-MUTEGS (b)	135,00	75,00	**0,023**	-
MUTS-MUTEGS (a) MUTEGS-MUTEGS (b)	135,00	75,00	**0,020**	**0,023**
MUTS(a) MUTS-MUTS (b)	54,50	116,50	**0,049**	0,054
MUTEGS (a) MUTEGS-MUTEGS (b)	57,50	78,50	0,270	0,278

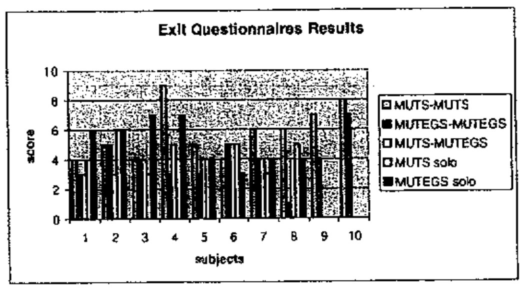

Fig. 1. Exit Questionnaire results

From Table 3 it is evident that: there is empirical evidence that the pairs MUTS MUTS performed better than the other pairs and solo designers, and the difference between the MUTEGS MUTEGS and the solo MUTEGS have not empirical evidence: that can be explained by the fact that the latter group had very strong individualities, as showed in Fig.1.

5 Experimental Threats

This section discusses the validity of the experiment, throughout the analysis of the experimental threats as classified in [26].

Threats to Construct Validity. The dependent variables aims at capturing the knowledge. We proposed questionnaire grading that surely cannot capture the overall aspects of the object to be measured. Tacit knowledge for its intrinsic nature is hard to formally describe and quantify. We consider what we measure an approximation of what we intend to measure.

Threats to Internal Validity. The following issues have been dealt with:

- Differences among subjects. Using a within-subjects design, error variance due to differences among subjects is reduced. In this experiment, students had a good degree in using UML. It is one of the main topics of their curriculum.
- Learning effects. The subjects were required to deal with only one run with only one assignment, so learning threat was cancelled.
- Fatigue effects. On average the experiment lasted a time short enough that fatigue was not very relevant. As a confirmation, the students asked for longer time to accomplish better the assignment.
- Persistence effects. In order to avoid persistence effects, the experiment was run with subjects who had never done a similar experiment.
- Subject motivation. The participants were volunteer, in order to help us in our research. We motivated students to participate in the experiment, explaining to them that they were learning a practice that should be useful in their professional career.
- The experimental package. The results of the run were independent from the experimental package, as showed in Table 4. We made a Mann Whitney test with the p-level fixed at 0.05, and there is no evidence that the differences due to the questionnaires were statistically significant.

Table 4. Testing the independency from the questionnaires

Test Between	Rank Sum 1	Rank Sum 2	p-level
Questionnaire A (1) Questionnaire B (2) in the experiment	540,00	406,00	0,161

Threats to External Validity. Two threats of validity have been identified which limit the possibility of applying generalization:

- Materials and tasks used. In the experiment we have used System design's documentation prepared by experimenters. The system showed a discrete degree of complexity, because it describes an existing system.
- Subjects. Students play a very important role in the experimentation in software engineering, as pointed out in [23]. In situations in which the tasks to perform do not require industrial experience the experimentation with students is viable [24].

Randomization of Sample. In order to accept the outcomes of the experiment as valid, it is necessary to make sure that there are not relevant differences in the samples to compare: the samples of solos and pairs have to be equivalent. If some differences on the entry questionnaires are detected, the randomization was not accomplished correctly. Table 5 shows this analysis for the experiment.

Mann-Whitney's method was used in all the tests because the data of samples were not normally distributed and the p-level threshold value was fixed at 5%.

The tests show that the MUTS subjects working as solos and those working in the pairs did not present significant differences at the entry questionnaire; similarly, the MUTEGS subjects of the solos' set and those of the pairs' set did not present significant differences. It is possible to conclude that the randomization was realized correctly.

Table 5. Testing the randomization of samples

Test Between	Rank Sum 1	Rank Sum 2	p-level
Entry Questionnaires of MUTS Pairs sample (1) MUTS Solos sample (2)	171,000	39,000	0,214768
Entry Questionnaires of MUTEGS Pairs sample(1) MUTEGS Solos sample(2)	112,000	59,000	0,130919

6 Conclusion

A growing interest in the software engineering community is turning toward agile practices, such as pair programming. Moreover, a lack of experimental validation of supposed benefits constitutes a main concern. Pair programming is expected to foster knowledge leveraging between pair's components. We have applied pair programming to the design phase, naming it pair designing. We have performed an experiment in order to validate how different educational backgrounds of the designers forming the pair can affect the knowledge leveraging.

The main conclusions follow:

- Forming pairs with individuals with the same educational background emphasizes the expected benefits of pair designing. Coupling a person with a scientific background and one with a non-scientific background does not seem to improve the latter but to make worst the former. We have evidence of this outcome.
- The individual can apprehend a lot by working alone, but this event is related to the single person; cannot be generalized.
- The tacit knowledge involved in the design is more articulated and complex than the programming knowledge, because concerns different levels of abstraction and includes also the consciousness of the implementation. Thus the scope of pair designing in terms of knowledge diffusion is wider than the one of pair programming.

The experiment owns an (apparent) point of weakness: it is not executed in industrial setting. Students play a very important role in the experimentation in software engineering: in situations in which the tasks to perform do not require industrial experience the experimentation with students is viable [22, 23, 24, 25]. The experiment owns also a point of strength: the number of subjects is approximately 50, and it sounds to us discretely valuable for statistical dependability. Some limitations of the work should deserve a major discussion, but due to matter of space, it is possible only list them: (i) knowledge building is influenced by many other factors of individuals, such as: learning styles, competence, and experience, and (ii) it should be useful to understand deeply the process of knowledge construction in the pair's components, that this study does not address.

Bibliography

[1] IEEE Software Special Issue on Knowledge Management, IEEE Computer Society, 3 (19), 2002.

[2] W. J. Hallinan, "*Improving Software Engineering practice through competency based personnel reviews*", Master Thesis, College of Graduated Studies, University of Idaho, 2001.

[3] G. Klein, J. J. Jiang, and D. B. Tesch, "*Wanted: Project teams with a blend of IS Professional orientations*", Communications of the ACM, ACM, 45 (6), 2002.

[4] I. Nonaka, "*A dynamic theory of organizational knowledge creation*", Organization Science, 5, 1994.

[5] F. Blackler, "Knowledge, Knowledge Work and Organizations: An Overview and Interpretation.", Organization Studies, SAGE Publications, 16 (6),1995.

[6] Y. Engeström, "Developmental Work Research: Reconstructing Expertise through Expansive Learning", in (eds). Nurminen M. and Weir G., Human Jobs and Computer Intefaces, North Holland, Netherlands: Elsevier, 1991.

[7] F. Blackler, "*Knowledge and the Theory of Organizations: Organizations as Activity Systems and the Reframing of Management*", Journal of Management Studies, Blackwell Publishing, *30* (6), 1993.

[8] C. W. Choo, *The Knowing Organization*, Oxford University Press, 1998.

[9] K. Beck, *Extreme Programming Explained: embrace change*, Reading, Massachusetts, Addison-Wesley, 2000.

[10] http://www.pairprogramming.com/.

[11] L. Williams and B. Kessler, "*The Effects of "Pair-Pressure" and "Pair-Learning"*", proc. of 13th Conference on Software Engineering Education and Training , IEEE Computer Society ,2000 .

[12] G. Canfora, A. Cimitile, and C. A. Visaggio, "*Working in pairs as a means for design knowledge building: an empirical study*", IEEE International Workshop on Program Comprehension, IEEE computer Society, 2004.

[13] J.T. Nosek, "*The case for collaborative programming*", Communication of ACM, ACM, 41(3), 1998.

[14] J. Nawrocki and A. Wojciechowski, "*Experimental Evaluation of Pair Programming*", proc. of European Software Control and Metrics, 2001.

[15] L. Williams, W. Cunningham, R. Jeffries, and R. R. Kessler, '*Straightening the case for pair programming*', IEEE Software, IEEE Computer Society, 17(4), 2000.

[16] G. Canfora, A.Cimitile, and C.A. Visaggio, "*Lessons learned about Distributed Pair programming: what are the knowledge needs to address?*", proc. of Knowledge Management of Distributed Agile Process-WETICE, IEEE Computer Society, 2003.

[17] L. Williams and R. L. UpChurch, "*In Support of Student Pair-Programming*", proc. of the thirty-second SIGCSE technical symposium on Computer Science Education, ACM, 2001.
[18] C. McDowell, L. Werner, H. Bullock, and J. Fernald, "*The Effects of Pair Programming on Performance in an introductory Programming Course*", proc. of the 33rd Technical Symposium on Computer Science Education, ACM, 2002.
[19] T. VanDerGrift, "*Coupling Pair Programming and Writing: Learning About Students' Perceptions and Processes*" proc. of SIGCSE, ACM, 2004.
[20] L. A. Williams, *The Collaborative Software Process PhD Dissertation*, in Department of Computer Science. Salt Lake City, UT: University of Utah, 2000.
[21] G. B. Foresythe, J. Hedlund, S. Snook, J. A. Horvath, W.M. Williams, R.C. Bullis, M. Dennis, and R. Sternberg, "*Construct validation of tacit Knowledge for military Leadership*", Annual Meeting of the American Education Research Association, 1998.
[22] J. Carver, L. Jaccheri, S. Morasca, and F. Shull, "*Using Empirical Studies during Software Courses*". Experimental Software Engineering Research Network 2001-2003. LNCS 2765, 2003.
[23] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. El Emam, J. and Rosenberg, "*Preliminary Guidelines for Empirical Research in Software Engineering*". IEEE Transactions on Software Engineering, IEEE Computer Society, 28 (8), 2002.
[24] M. Höst, B. Regnell, and C. Wholin, "*Using Students as Subjects – A comparative Study of Students & Professionals in Lead-Time Impact Assessment*", proc. of 4th Conference on Empirical Assessment & Evaluation in Software Engineering (EASE), 2000.
[25] V. Basili, F. Shull, and F. Lanubile, "*Building Knowledge Through Families of Experiments*". IEEE Transactions on Software Engineering, IEEE Computer Society, 25 (4), 1999.
[26] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., Experimentation in Software Engineering: An Introduction, Kluwer-Springer, 2000.

Appendix

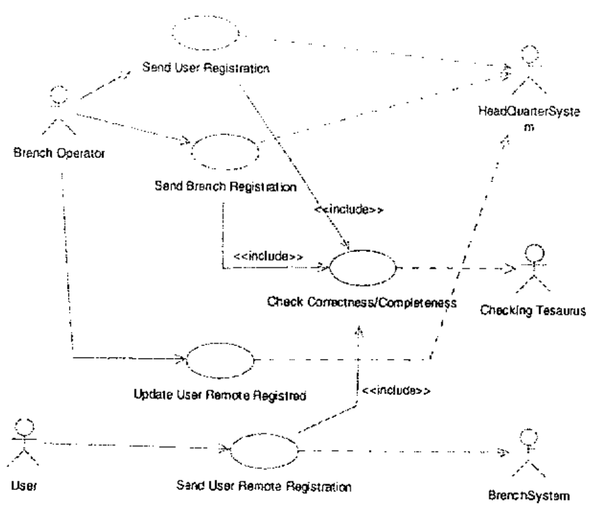

Fig. 2. Exemplar Use Case

Table 6. Excerpt of Use Case Specification

UseCase	Send User Registration
Description	The Branch operator inserts data in the registration form, provided by the user. Validation of the form is launched.
Exceptions	The form is not correct or complete. The sending of data is successfulness.
Actors	BrenchOperator, HeadQuarterSystem.
UseCase Extends	Nn
UseCase Uses	Check Correctness/Completeness
UseCase Inputs	Name, address, offered books list (in case the user is a vendor) with specifications: title, author, publisher, language, publishing year, ISBN.
UseCase Outputs	Recording of data of the new user.
Criterion of Acceptance	Data of the new user are stored in the database of the Local Branch.
Related Expectations	Database management system. Correctness and completeness checks. Data sending to the Headquarter.
Related Reqs/ Use Cases	Check Correctness/Completeness

Table 7. Questionnaire QB

1.Could Remote Registration of User (User Remote Registration Sending) extend local user registration (User registration Sending)?		
a. Yes	b. No and it does not make sense	c. Possible, with proper modifications
2. Does the updating of user data (User Remote Registered Updating) require the correctness and completeness check (Correctness/Completeness Checker)?		
a. Yes	b. No and it does not make sense	c. Possible, with proper modifications
3. Could use cases Notification of Transaction To Buyer and Notification of Transaction To Branch be merged in one use case?		
a. Yes	b. No and it does not make sense	c. Possible, with proper modifications
4. Could the use case Update Database extends the use case Local Book Search?		
a. Yes	b. No and it does not make sense	c. Possible, with proper modifications
5.Given a transaction, can information concerning vendor be obtained through the Book (object)?		
a. Yes	b. No and it does not make sense	c. Possible, with proper modifications
6. A (object) Branch Operator must have executed at least one operation, otherwise it does not exist in the System.		
a. True	b. False	c. This information is not provided by documentation.
7. It is possible to obtain the list of registered users in a local branch through Data contained in a Branch (object).		
a. True	b. False	c. This information is not provided by documentation.
8. DataHandler (object) helps query the Database.		
a. True	b. False	c. This information is not provided by documentation.
9. Checker (object) verifies if all the fields of the form are filled in.		
a. True	b. False	c. This information is not provided by documentation.
10. The user interface is provided only for the remote part of the system.		
a. True	b. False	c. This information is not provided by documentation.