

# Investigating IBIS in a Distributed Educational Environment: the Design of a Case Study

Daniela Damian, Filippo Lanubile, Teresa Mallardo <sup>1)</sup>

## **Abstract**

*In this paper we present a first experience in conducting distributed software inspections in support of the remote communication between clients and developers collaboratively developing a requirements specification. We intend to assess if remote synchronous requirements negotiations can benefit from asynchronous discussions for reducing the number of open issues to be resolved in order to short the negotiation agenda.*

*The case study took place in the context of a global software development course and involved three universities from three different continents.*

## **1. Introduction**

Meeting-centered activities are very critical activities to conduct in a geographically-distributed environment. We are investigating approaches that enable effective requirements negotiations [Boe98] and software inspections [Lai00] in a distributed software development, as they are activities that should support collaborative software engineering in remote teams as well as they do in traditional software teams [Grü04, Hal03].

In this position paper we report the design of our research that investigates the usefulness of asynchronous discussions, as part of the requirements inspection process, to facilitate more effective synchronous requirements meetings in distributed teams [Dam03]. In particular, we describe how we are planning to study the use of a web-based inspection tool, IBIS [Lan03], in support of the remote communication between clients and developers collaboratively developing a requirements specification.

IBIS supports remote teams during the inspection of requirements documents, and in particular supports teams through stages of issue Discovery, Collection, and Discrimination. During the Discovery stage, inspectors review individually the document with the help of checklists or scenarios, and records issues. In the Collection stage the inspection leader or the document's author collate recorded issues and eliminate duplicates. In the Discrimination stage the inspection team makes decisions about collated issues. The Discrimination stage is designed as a structured asynchronous discussion with two mechanisms: posting of messages for each issue under discussion and voting as to whether an issue is a true issue or not (false positives).

---

<sup>1</sup> Daniela Damian, Department of Computer Science, University of Victoria, BC, Canada, email: danielad@cs.uvic.ca  
Lanubile Filippo, Dipartimento di Informatica, University of Bari, Italy, email: lanubile@di.uniba.it  
Mallardo Teresa, Dipartimento di Informatica, University of Bari, Italy, email: mallardo@di.uniba.it

## 2. Case study setting

To investigate the usefulness of asynchronous discussions to facilitate effective synchronous requirements negotiation meetings, we studied IBIS in six educational global project teams in a global software development (GSD) course. Each software project followed an iterative development process in which designers in collaboration with clients were to develop a requirements specification (RS): after a requirements elicitation stage, a requirements inspection of an early draft of RS involved the discovery as well as asynchronous discussion of requirements issues (in IBIS) and was further followed by requirements negotiations and prototype demonstrations before the final draft of the RS was delivered.

The Global Software Development course was offered in a three University collaboration involving University of Victoria, Canada, University of Technology, Sydney, Australia, and University of Bari, Italy during January and May of 2005<sup>2</sup>. The course involved a total of 32 students. 12 of them were Master's and Doctorate students at the University of Victoria, 2 graduate and 8 undergraduate students at the University of Technology, Sydney, and 10 Master's students at the University of Bari.

As shown in Table 1, the Canadian students worked on software projects with the Australian and Italian groups as follows: the 12 Canadian students formed three groups of 4 (Gr1-3), the Australian students formed two groups of 5 (Gr4-5), and the Italian students formed two groups, of 3 and 7 students respectively (Gr6cl and Gr6dev). Each Canadian and Australian group was involved in two different projects, playing the role of client (C) and developer (D) respectively. Each of the two Italian groups was involved in only one project, either as a client (Gr6cl) or as a developer (Gr6dev).

**Table 1. Groups and allocation to course projects**

Country	Group	Project A (A1, A2)		Project B (B1, B2)		Project C (C1, C2)	
		PT1	PT2	PT3	PT4	PT5	PT6
Ca	Gr1	Client (C)					D
	Gr2		D	C			
	Gr3				D	C	
Au	Gr4	Developer (D)			C		
	Gr5		C			D	
It	Gr6cl						C
	Gr6dev			D			

<sup>2</sup> More information can be found on the course website: <http://segal.cs.uvic.ca/csc576b>

## 2.1. The software projects

There were three distinct projects in the course (A, B and C).

Project A (A1 and A2 in Table 1): *Global software development system*. The students were to design a system to facilitate collaboration in GSD by supporting informal communication as well as document exchange in remote teams. Tasks supported by the tool included: displaying people's availability information, viewing changes between different versions of documents and authors of those changes, visualizing the evolution history of a particular document, and discovering who has been working on a particular document or section of a document.

Project B (B1 and B2): *iMedia system*. The students were to design the interface for a "iMedia" software that will allow users to purchase movies online, organize their movie library, and play movies. One of the key requirements was that the interface be simple to use even for inexperienced computer users, without sacrificing key features.

Project C (C1 and C2): *Virtual Realty system*. The students were to design a system that provides accurate and easy-to-find information to real estate agents and home buyers in the Victoria area. The system had to display an interactive map, where the end-user can zoom in, zoom out, pan, etc., and click on it to get the information of the property.

Two global software teams were allocated to each project, each with the client and developer group in two different countries (see Table 1). The projects were assigned to groups before group membership was determined. The project assignment was done so that each group worked with a different partner group for each of the two projects it was assigned (with the partner group always located in a different country), and so that the two projects it worked on were on a different topic.

## 2.2. The RS development process

Each project followed an iterative process of developing a requirements specification (RS) through collaboration between developers and clients over a period of 7 weeks. The RS development life-cycle consisted of ten phases of requirements discovery and validation, and through which the understanding and documentation of requirements was to be improved:

1. Kickoff Meeting
2. Create RFP
3. Analyze RFP
4. Requirements Elicitation.
5. Create RS 1.0
6. Discover issues on RS 1.0
7. Asynchronous discussions
8. Requirements negotiation
9. Create prototype demo
10. Create RS 2.0

The purpose of the asynchronous discussion was to come to an understanding of each issue and those issues that could be closed online (i.e. where resolution could be reached without further negotiation) or remained open issues (anything else, and which had to be further negotiated in real-time discussion). Discussants attempted to close issues by using the two mechanisms in IBIS:

posting messages with respect to a certain issue, and voting as to whether it is still an open issue or is resolved and thus could be closed. Those issues that could not be resolved during the asynchronous discussion in IBIS (i.e. open issues) were then discussed during a scheduled requirements negotiation held in a videoconference meeting between developers and clients.

Each of which stages included either client, developer or group tasks and ended with a project deliverable on which students were graded for the class. The final deliverable was the final version of the SRS, which reflected the shared understanding of the project that the clients and the developers built over the previous four phases. The project finished at the point where the developer group would start writing the code for the system called for by the project.

### 3. Research design: exploring the usefulness of asynchronous discussions to facilitate effective synchronous requirements negotiations

The usefulness of asynchronous discussions prior to requirements negotiations consists in focusing the synchronous meeting, in which requirements are negotiated, on the issues that could not be resolved earlier, during the asynchronous discussion.

As a first assessment, we instructed half the projects to conduct the asynchronous discussion before the negotiation, and half the projects to jump into the negotiation without asynchronous discussion. Table 2 indicates which projects conducted the Asynchronous discussion (AD) and which did not (No AD). Then, the process variant (AD or No AD) was the main independent variable that we manipulated for experimental purposes.

**Table 2. Experimental design**

project team (client/developer )	process variant
A1 (gr1/gr4)	No AD
B1 (gr2/gr6dev)	No AD
C1 (gr3/gr5)	No AD
A2 (gr5/gr2)	AD
B2 (gr4/gr3)	AD
C2 (gr6cl/gr1)	AD

When asynchronous discussions were scheduled for a project team, both clients and developers used the IBIS tool over a week, as a threaded discussion forum. The aim was to come to an understanding of each issue by exchanging messages and to an early resolution through a common agreement expressed by voting. Those open issues that could not be closed during asynchronous discussion in IBIS were then left for the synchronous negotiation meeting. For those project teams which skipped the asynchronous discussion, all collated issues were thus considered as open issues to be dealt at the negotiation.

To measure the usefulness of asynchronous discussions, we are going to collect data about: collated issues, closed issues during asynchronous discussion, open issues before synch negotiation, closed issues during sync negotiation, open issues after sync negotiation. In addition, to understand better

what actually happened during asynchronous discussions, we are also collecting the following information from the asynchronous discussions in IBIS: discussants, posted messages, messages per issue and participant, votes per issue and participant.

To complement the quantitative data, we gathered the students' perceptions on the usefulness of the asynchronous discussion through two questionnaires, administered at middle and end point in the course. The two surveys were designed to elicit feedback about processes as well as tools used in the distributed project.

#### **4. Future work**

We are planning to analyze our data. Firstly we want to measure the dependent variables to assess our research goal. Unfortunately, because the sample size is so small (only 3 projects for each process variant) we think that it will make no sense running any kind of statistics for measuring effectiveness. Then we will reinforce our quantitative results from the questionnaires' analysis and, in particular, from the comments on open questions we can try to explain unexpected results.

We seek feedback from the workshop with regards to the research directions presented here. We hope to present some findings from analyzing this data at the workshop.

#### **Acknowledgements**

We gratefully acknowledge the technical support of Luis Izquierdo and Fabio Calefato during the duration of the three-University course. Thanks also to Dr. Ban Al-Ani for her instrumental role in this collaboration and to all the students who participated to the remote projects.

#### **References**

- [Boe98] B.W. Boehm , A. Egyed, J. Kwan, D. Port, A. Shah, R. Madachy. Using the WinWin Spiral Model: A Case Study. *Computer*, 31(7): 33-44, July 1998.
- [Dam03] D. Damian, and D. Zowghi. Requirements Engineering challenges in multi-site software development organizations. *Requirements Engineering Journal*, 8: 149-160, 2003.
- [Grü04] P. Grünbacher, M. Halling, S. Biffel, H. Kitapci, and B.W. Boehm. Integrating Collaborative Processes and Quality Assurance Techniques: Experiences from Requirements Negotiation. *Journal of Management Information Systems*, 20(4): 9-29, 2004.
- [Hal03] M. Halling, S. Biffel, and P. Grünbacher. An economic approach for improving requirements negotiation models with inspection. *Requirements Engineering Journal*, 8: 236-247, 2003.
- [Her03] J.D. Herbsleb, A. Mockus. An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Transactions on Software Engineering*, 29(3): 1-14, 2003.
- [Lai00] O. Laitenberger, and J.M. DeBaud. An encompassing life cycle centric survey of software inspection. *The Journal of Systems and Software*, 50 (1): 5-31, January 2000.
- [Lan03] F. Lanubile, T. Mallardo, and F. Calefato. Tool Support for Geographically Dispersed Inspection Teams. *Software Process: Improvement and Practice*, 8(4): 217-231, October/December 2003.