3TZ Collaborative Team Environments incorporating the Hybrid Holonic Architecture

Zenon Chaczko
Faculty of Engineering and IT
University of Technology (UTS)
Sydney, NSW, Australia
zenon.chaczko@eng.uts.edu.au

Chris Chiu
Faculty of Engineering and IT
University of Technology (UTS)
Sydney, NSW, Australia
christopher.chiu@uts.edu.au

Ryszard Klempous & Jan Nikodem
ICECRE,
Wrocław University of Technology,
Wroclaw, Poland
ryszard.klempous@pwr.wroc.pl
Jan.Nikodems@pwr.wroc.pl

***Abstract*— The paper describes a business reengineering process (BPR) approach to address multi-timezone (3-timezone or 3TZ) collaborative teamwork environments by combining the Holonic architecture with the Zachman Metamodel Framework. While the use of collaborative project systems is not new, the methodology to share time resources from different timezones seeks to address pedagogical and engineering process concerns in team-based project development. The benefits of collaborative project management tools go beyond a uniform platform to deploy project resources, but to also enhance systemic processes and engineering practice. This facilitates team members to dedicate their time towards common work tasks, delineates individual and shared work packages, and improves student-tutor feedback techniques as teachers can actively monitor progress of development throughout the project lifecycle.**

Keywords-3 Time Zone (3TZ), Collaborative Project Development, Business Process Reengineering (BPR) Holonic Architecture, 3TZ Virtual Labware, Zachman Framework

I. THE NEED FOR MIDDLEWARE IN 3TZ

Large software development companies [4], including Cisco, Google, IBM and Microsoft have embraced the idea of working with geographically and temporally spaced teams. The Open Source community, including the Linux kernel and Apache web server projects, has worked in this fashion for years, but their approach usually involve either a singular core team working in one location, or various singular developers working on separate tasks and integrating at the end of the development cycle. To effectively prepare university students to work in global environments in the 24/7 development cycle, the principles of continuous collaborative development needs to be taught and practiced within the curriculum [5][9]. Although the development of collaboration software platforms continues apace, there needs to be an emphasis on tools to integrate the facilitation of task management aspects with project management in a distributed team project.

The premise of 24-hour continuous development is modelled towards project-oriented tasks that have strict deadlines, in a finite allocation of time. If a functional or security bug is discovered in a mission critical application, there is a need to find a solution within the shortest period of time [8]. As an example shown below in Figure 1, a problem that would take 24 hours to resolve would require 3 workdays.

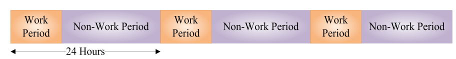

Figure 1. Single Timezone Work Allocation

Following the 24-hour continuous time period process as shown in Figure 2, the work task can be followed through with a one day turnover. While the resources of a single site would not have the capacity to meet the intensive activities of a continuous work strategy; a three-site 24-hour continuous development platform would meet this goal [9]. For companies offering 24/7 systems support, this can mean the difference between an inexpensive and an expensive support cost [5][10].

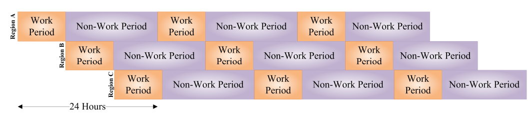

Figure 2. Multi-Timezone (3TZ) Work Allocation [6][9]

A. *Agile Development Methodologies and 3TZ*

Scrum is an agile software development methodology that overlaps with the concept of continuous software development, an iterative process where development phases are overlapped and performed by a cross-functional team [6][7]. For example, the handover-synchronisation components of continuous collaborative development can be achieved, from a project management perspective, in the daily team meetings that occur during a scrum. This synchronisation would run smoother when dealing with intensive development tasks linked to a scrum handover, using iterative and incremental development frameworks [9][11]. As small iterations of project development allow for periodic re-synchronisation, these iterations can be utilised as a period where the project management team can ensure the vision is shared across sites. Issues arising in the previous iteration can be isolated and rectified before moving to the next iteration, as depicted in Figure 3.

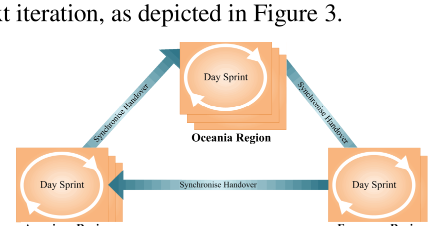

Figure 3. 3TZ Synchronisation of a Sprint Handover [6]

Specifically, the main concern of our research is to develop a software middleware platform for theoretical development and experimental verification of the 24-7 mode of telecollaborative engineering. The core infrastructure is required to enable the collaborative dynamics of people working on different regions, in which both human, computer and software actors could interact in a networked environment gateway. The eventual goal is to setup a systematic design approach to solve various problems pertaining to applications described in the aforementioned scenario, from both a project management and pedagogical assessment perspective.

II. EMERGING TRENDS IN MIDDLEWARE

Software systems middleware technologies including commercial of-the-shelf (COTS) products, such as Microsoft's .NET Remoting, Sun's Remote Method Invocation (RMI), and Object Management Group's (OMG) Common Object Request Broker (CORBA) have dramatically matured and become de-facto standards in IT industry [13]. These solutions are being used to reduce the software development life cycle (SDLC) and improve the effectiveness of building systems by reducing costs (time, efforts and resources) in a range of business domains. Whilst middleware solutions have traditionally been used in business applications, including asset management systems, enterprise management resource planning and e-commerce reservation systems; they rapidly have become in dominant use for Distributed Sensor Actuator Networks systems (SANETs) that are built on evolvable, self-healing non-in-situ networks with actuation and control. This encapsulates monitoring and control processes, operations, networks and hardware systems in telecommunications, defence, manufacturing and infrastructure industries.

SANET applications possess distinct characteristics relating to its mission critical aspects and time constraints. Time criticality and strict deadlines is essential, as the correct data response that is delivered beyond a given threshold can result in unpredictable or catastrophic consequences. Therefore, the need for SANET middleware models to meet stringent Quality of Service (QoS) qualitative requirements such as scalability, robustness, usability, security, efficiency, latency, privacy and trust [1][3]. For all application domains, the ultimate goal of middleware is to support the process of software development by facilitating integration of components and protecting engineers from inherent and accidental complexities related to heterogeneous computing environments, management of resources, security and fault tolerance.

III. ENCAPSULATION OF DISTRIBUTED DATA MODELS

The unifying characteristic of modern middleware platforms, for project development work and SANET applications, is the paradigm of common data repositories that can be accessed concurrently, within the existing domain concerns and constraints. Modern component middleware systems [1][13] is a type of middleware that allows reusable services to be composed, configured, integrated and installed to build software applications efficiently and reliably, while adhering to needs of distributed shared memory across disparate environments. As shown in Figure 4, distributed data space concerns are addressed through Tuple Space implementations as supported in modern component middleware systems, following the multi-verse concepts of elemental slicing of a core entity of representing the structures and interconnections between internal entities [1]. This provides users with a specific set of capabilities:

- Connector Facilities within Components: Includes remote procedure calls, remote method invocation or message passing mechanisms.
- Horizontal Models of Infrastructure Services: Request brokers or publish-subscribe mechanisms between components within the same platform.
- Vertical Models of Domain Paradigms: Common semantics and context awareness, and high-level reusable services spanning from transaction and lease support, to multilayer security and privacy for multiple platforms

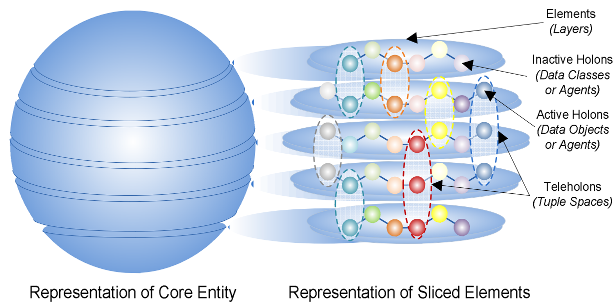

Figure 4. Tuple Space paradigm consists of distributed Memory Spaces overlayed on multiple platforms [1]

Common examples of commercially available component middleware include Sun's Java Enterprise Edition Platform (Java EE), Microsoft's Windows Communication Foundation (WCF) and OMG's CORBA Component Model. Each software technology uses different component models, protocols and application programming interfaces (APIs) to address distributed domain concerns. The utilisation of component middleware follows a ubiquitous development framework, combining SANET solutions and forming distributed systems interconnected by shared networked environments.

The important issue for component middleware systems is being able to alleviate the compositional complexity and management of distributed SANET systems. Reducing the software development life cycle (SDLC), thus shortening the time-to-market is essential in modern engineering and business concerns [12]. As the majority of developer roles are to assemble distributed networked systems by selecting a combination of custom made components and compatible COTS frameworks [6][7], the process of selection is an important focus of this research. The construction of an effective system requires components to possess compatible interfaces (APIs), semantics, context and protocols which makes the analytical process of selection and development of a compatible set of software components a challenging task. Problems are exacerbated by the availability of various vendor-driven strategies for configuring and deploying the underlying software middleware to leverage dedicated hardware and software features.

IV. THE PROPOSITION OF AGENT-BASED HOLONIC 3TZ MODELS

A. Biomimetic Applications to Technology Models

The model of Bio-inspired Hardware Systems elaborated in Sipper et al [12] has an underlying basis on living organisms, with complex processes and functions as a whole being. Applications of this approach can be found in neural networks and programmable logic circuit design. The improvement in technology and procedural practice has facilitated the shift towards evolutionary computation methods in engineering processes.

It is from the fundamental basis of adaptive computing models in which bio-inspired software models by Chaczko [1][2] has advanced over the years. The direct relationship between the need for a middleware platform to support large, heterogeneous team environments ties with the logical basis in which biomimetic software principles by can be applied to solve the problem of 3TZ infrastructure [14][15], *to resolve the social and technological synchronisation concerns between disparate teams or organisations*.

As shown in Figure 5, the biomimetic model perspective can be projected onto a three-dimensional space, according to the POE Classification Model [1][2]. The POE model represents the different organisational levels of organization, with POE standing for Phylogeny, Ontogeny and Epigenesis.

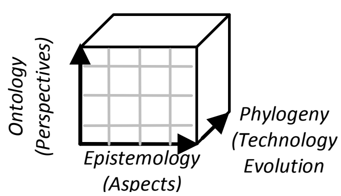

Figure 5. The POE Classification Model (Phylogeny, Ontology, Epistemology) [2][12]

- **Phylogeny:**
  Biological Context: This entails the evolution of genetics in species.
  Engineering Correlation: This relates to the implementation of heuristic problem solving algorithms and the evolution of technology.

- **Ontogeny:**
  Biological Context: This is concerned with cellular growth process, multi-cellular organisation, cellular division and differentiation from the parent cell to the child cells. Each child cell processes a copy of the original genome.
  Engineering Correlation: This is the perspectives of a domain space and how it fits into the relationship of related or shared domains.

- **Epigenesis:**
  Biological Context: This involves the adaptation and learning processes. The nervous, immune and endocrine system are characterised by epigenesis.
  Engineering Correlation: This space corresponds to the facets or aspects of knowledge acquisition that can either be in a shared context or common domain.

B. The Hybrid Holonic-Telonic Architecture

Following the concerns of modern middleware systems [13][15], the key driver for developing a 3TZ middleware framework is to solve the core problems prohibiting engagement of different regional teams to collaborate in a coordinated and synchronised fashion:

- Team and Project Management concerns driving the task deliverable agenda, and tracking the development pace from pedagogical and learning perspective.

- Vendor and technology specific applications such as communication and protocol specifications/definitions.

A vendor agnostic approach to addressing distributed data concerns, while following the conceptual framework of tuple space models is the adoption of Holonic architectural framework for the 3TZ Labware environment [3][9]. The concept of a holon or whole is to distinguish and differentiate the aspects of an entity that can be a whole and a part simultaneously. As shown in Figure 6, the application of the Holonic architectural model from a distributed data space perspective can be applied using a dimensional triality applying the POE classification model [2][12]:

- **Epistemological Dimension:** The Holonic model works on a horizontal orientation, which focuses on individual holons residing in the same layer or domain space. This can be represented as project tasks from within the same region or institution.

- **Ontological Dimension:** The Tele-holonic/telonic model works on a vertical orientation, where cross-layer address the needs from an alternative perspective. This is equivalent to the tuple space of a task, where the handover or synchronisation of the project occurs.

- **Phylogenetic Dimension:** The unification of holonic and telonic models takes place when specific holons in a given axis is shared amongst holons takes place. It is important to note that teleholons do not just cross holonic layers and remain in the same vertical orientation, but can cross disciplines and merge amongst multiple holons in different layers.

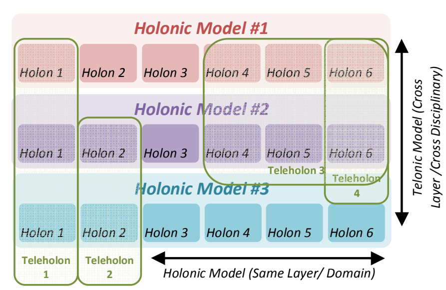

Figure 6. Holonic with Telonic Architectural Models

As the core requirement of the 3TZ Labware Infrastructure is to improve cognitive, conational and emotive (CCE) [7] aspects of teaching and learning and contrive final subject evaluation and student feedback, the identification of the main concerns of the middleware environment was indentified into 5 core concerns.

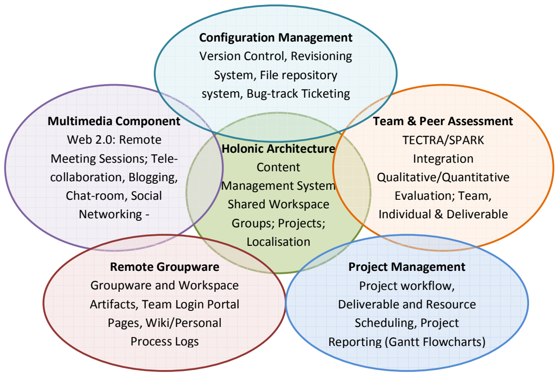

Figure 7. Holonic Architecture in scope of Collaborative Virtual Labware

The core concerns of the 3TZ Labware environment are listed as depicted as Venn diagram in Figure 7:

- **Shared Workspace:** Incorporates the 24-7 Virtual Student Exchange Server, consisting of Eventflow management, Virtual Portal (vortal) gateway and Shared Workspace Environment.
- **Team and Peer Assessment:** Quantitative and qualitative team evaluation, with flexible deliverable grading systems with academic and peer feedback.
- **Project Management:** Project scheduling and task supervisory management systems, with resource organisation and time/task scheduling.
- **Remote Groupware:** The collaborative elements of the system, including personal process work log artefacts, team and brainstorming forums.
- **Configuration Management:** This includes quality control concerns, including the version control management and file repository systems.
- **Multimedia & Project Development:** Incorporates the virtual remote laboratory systems, virtual machine deployment and Web 2.0 social networking activities.

C. Multi-layer Middleware Paradigm Realisation

The consideration of the Holonic Hybrid architecture corresponds with the needs of the 3TZ Labware middleware infrastructure, as it addresses multi-timezone project concerns from two central views:

- **Holonic Concerns:** Horizontal perspectives for a single regional institution; such as the project management for a given set deliverable or task.
- **Telonic Concerns:** Vertical perspectives amongst cross-regional institutions, such as the Shared/Ad-hoc workspace where teams can facilitate and negotiate common collaboration artefacts that require handover.

As represented in the Teleholon depiction in Figure 6, the holons are shown as individual resource concerns of which the holonic architecture encapsulates the pool of resource sharing for a given institution, and the teleholons are represented as the collaboration or synchronisation between resource holons, in which it models the telonic system architecture.

To ensure that '*that one gives the resources that can be afforded to be given away*' [3], public-key cryptography is ensured throughout the 3TZ Labware Middleware system. Asymmetric security controls are in place so that the host and the recipient must share the same public and private key before a contract exchange is taken place between the parties. This forms the core basis of the middleware infrastructure, so that only common team members have access to the private keys for holonic service agent authentication and prevent authorised access or cross-collaboration that may constitute as plagiarism.

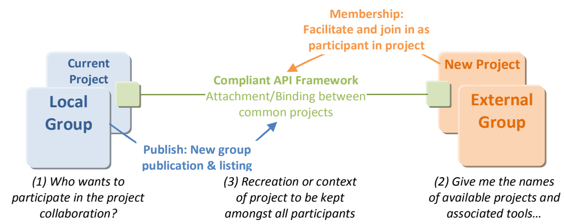

Figure 8. Cross-Institutional Project Synchronisation

The contract exchange between separate holon agents on different platforms follows the broker software design pattern, as elaborated in Figure 8:

*1) **Service Provider (Local Group):*** The provider publishes their associated project and available resources to the compliant middleware framework using a standard API set defined by the broker. This allows the local group to prepare themselves in readiness to accept willing parties to collaborate in the project.

*2) **Broker (3TZ Labware Arbitrator):*** This consists of the middleware infrastructure that ensures all parties communicate through a compliant API set, establishes and registers all available project contexts, and facilitates in the contract binding between two common parties.

*3) **Service Consumer (External Group):*** The consumer accesses the broker's registry to list services it has authority to consume. Membership to the service provider's project facilities takes place when authentication and identification checks are successfully made between the groups.

D. Basis of SOM-Zachman Transformation Model

The 3TZ Labware architecture is enhanced through the Self Organised Map-Zachman (SOMZ) [1][16] transformation model for business process reengineering to identify the formation of teleholons within the hybrid Holonic architecture. 3TZ project management processes to need to be qualified in terms of acceptance, for both priorities and qualitative drivers, for completing the reformation engineering paradigm loop. The high-level SOMZ methodology shown in Figure 10 is encapsulated in the steps below:

*1) **Model initial process inputs:*** Idealisation of problems consists of artefacts including the results of surveys, technical reports, Total Quality Management process evaluations and logs, expressed in a quantified form or metric of completion. *Realisation:* The group and project artefacts for a team awaiting collaboration in the 3TZ middleware infrastructure are collated together.

*2) **Heuristic input projection via process template:*** Using an unsupervised neural network (Self Organised Maps), a lookup of possible tuple spaces that match common criteria is established, for ontological and epistemological requirements.
Realisation: Weights/quantitative values are applied to input artefacts, based on the priority or relevance of the process for a given model to be transposed. As an example, the architect's view wants to follow 6σ processes, such that the teleholon formed focus on product model quality.

*3) **Heuristic process evaluates artefacts compared to benchmark to highlight anti-patterns/ concerns in existing structures:*** The collapse or reduction of the problem to a realisable form results in the reengineered process.
Realisation: Collecting the results from the inputs and with a heuristic weighting process, inputs above a specified threshold (95% Confidence Interval) are the elements where the team should focus their efforts to improve their existing processes.

*4) **Feedback loop to apply reengineered process into existing system and reapply SOMZ transformation:*** As an open system, project processes will change over the course and duration of the deliverable outcome. Hence, the anti-patterns in a system will change through the project phases.
Realisation: Reuse the existing weights/values of the last reengineered teleholon and factor the change in constructs/form into the 3TZ Holonic architectural space.

From a pedagogical perspective, the remaining holons formed after the SOMZ transformation is complete show to the project managers and academics that these particular elements within the 3TZ collaborative framework is where the focal attention should be made to improve existing engineering processes [5][8]. From an engineering context, the synchronised collaboration between groups, particularly from an integration point of view, is where the most issues or social anti-patterns tend to surface. Therefore, the early identification and rectification of these team collaborative concerns is essential to highlight and teach students of the cascading effects resulting from unchecked problems within team dynamics [14].

V. The 3TZ Virtual Labware System in Practice

In the subject ICTD Design in Spring 2009, a team of 20 students perform their project development work by connecting to the 3TZ Virtual Labware environment using a web browser interface, with the main lessons gained in project management and peer work coordination activities. Introductory seminars are provided to allow students to setup and customise their teamwork session for their given project with an example setup in Figure 9. The core elements of the 3TZ Labware Infrastructure that are implemented in the final design strategy are listed in Table I, with a storyboard elicitation based upon the 3TZ software middleware infrastructure holons [3][4][11].

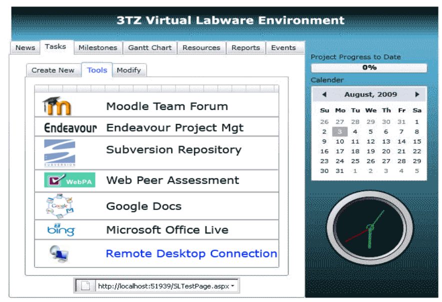

Figure 9. Cross-Institutional Project Synchronisation

TABLE I. 3TZ Labware Infrastructure & Utilisation Results

Core Component	Design Artefacts and Requirements	
	Technology	***Storyboard and Typical Utilisation***
Shared Workspace Core *(Compulsory component – 100% team utilisation)*	Custom Arbitrator & Broker: *Apache Tomcat with J2EE*	This comprises of the 3TZ arbitrator and coordinator of participating holon components in the 3TZ Labware system. This backend system provides a common API setup to allow future holon modules to be added as the needs and requirements of project deliverables change over time.
	Open Source Authentic-ator: *Apache DS LDAP Server*	When the team is officially organised and approved by the lecturer, the lecturer will notify the tutor to request a setup for a new Labware project, which is then approved by the lecturer. The designated Project Manager (PM) sets user roles and allocates resource pools in conjunction with sub-team leads.

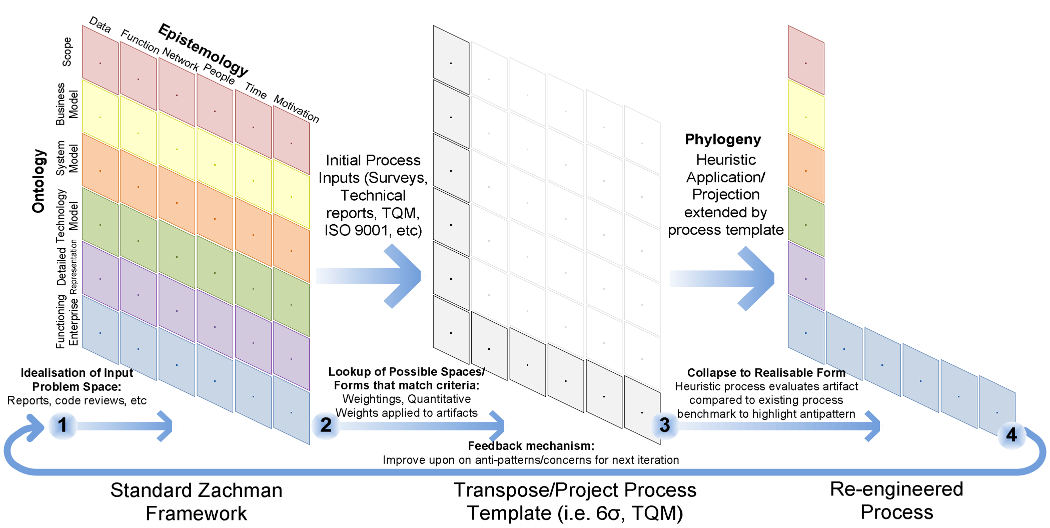

Figure 10: Formulation of the SOM-Zachman BPR Transformation Model [2][16]

Core Component	Design Artefacts and Requirements	
	Technology	***Storyboard and Typical Utilisation***
Team & Peer Assessment *(20% team utilisation)*	Open Source: *WebPA*	Team members review and critique their fellow peer's work artefacts. In tandem with lecturer and tutor, each member validates and demonstrates their claims of project contribution relative to the team.
Project Management *(100% team utilisation)*	Open Source PM Tool: *Endeavour*	The PM and Quality Assurance (QA) leader will setup the project schedule and milestone for artefact delivery. Critical paths are established to determine project course and ensure deadlines are met.
Remote Group Management *(33% team utilisation)*	Open Source: *XRDP*	The team, in conjunction with tutor and lecturer, will approve nominations for setup of Virtual Machine (VM) operating system, hardware (CPU/memory) & integrated development environments.
Configuration Management *(100% team utilisation)*	Open Source: *Subversion & TRAC with Moodle*	Once VM machine is setup, users can start development once the requirements validation and system design artefacts are complete and finalised in the Subversion repository. Students access their VM via the web interface to perform physical operations, with connection to the console by Remote Desktop Connection (RDC).
Multimedia and Web 2.0 *(15% team utilisation)*	*Google Apps API & Bing API*	Students can collaborate using Web 2.0 Tools including Google Docs, Office Live Workspace and communication chat tools .(i.e. Google Talk and MSN Messenger)

A. Further Investigation

The future investigation with the 3TZ Labware tool has been identified as follows:

- The current issues with the 3TZ The Group Labware tool are the Multimedia and the Team Assessment tool. The underutilisation has been attributed to technical problems with the VM environment, including physical hardware connectivity issues and unfamiliarity of VM concepts. Students with prior knowledge or experience can guide fellow peers to be familiar with the environment.
- Cross institutional collaboration to allow for multi-time zone engagement. Technical exchange workshops to negotiate and agree upon a common API framework are the main technological concerns to overcome; while academic subject material sharing is essential to ensure coordination of the pedagogical activities associated with the coursework [3][9][14].

VI. Main Conclusions

The 3TZ Virtual Labware environment achieves the goal of enabling students to collaborate effectively in team-based scenarios, as the conjunction of the peer and team holon agent tools allow a cross-comparison and validation of individual contributions. Further outcomes in terms of user participation and feedback have established the main preliminary outcomes:

- ***Project Management:*** Project Managers and the QA team have used the project management tools extensively to chart team progress and update notification via the e-wiki tool in Subversion TRAC.
- ***Configuration Management:*** With the Development Architect, all users take advantage of the configuration management tools to commit code changes and integration branches to the main code trunk.

The preliminary outcomes of the 3TZ Labware tool identify that consideration of the organisational construct of a team can be reformed as a result of early identification of technological or team dynamic problems – while also facilitating pedagogues to have insight and early intervention into the project development process when required. The synergies between the project and configuration management holonic agents represent a feedback mechanism taking place where anti-patterns and concerns identified at the development stages drive the project management schedule, thus enhancing improvements in the project management process.

Acknowledgements

This work was supported by E.U. Regional Development Fund and by Polish Government within the framework of the Operational Programme - Innovative Economy 2007-2013. Contract POIG.01.03.01-02-002/08-00, Detectors and sensors for measuring factors hazardous to environment - modelling and monitoring of threats.

References

[1] Chaczko, Z. et al, (2008), A Bio-inspired Telecollaboration Service Taxonomy: Usability Related Concerns, Broadcom 2008

[2] Chaczko, Z. et al, (2007), NIE Models of Biomimetic Software Systems, CSIRO ICT Centre Conference.

[3] Chaczko, Z., Davis, D., Mahadevan, V. (2004), New Perspectives on Teaching and Learning Software Systems Development in Large Groups, IT Higher Education & Training 2004, Istanbul, Turkey

[4] Chaczko, Z., Mahadevan, V., Chaczko, E. W. (2005), Blogging in Teaching and Learning Software Systems, IT Higher Education & Training 2005, Juan Dolio, Dominican Republic

[5] Doerry, E., Klempous, R. Nikodem, J., Paetzold, W. (2004), Virtual student exchange: lessons learned in virtual international teaming in interdisciplinary design education, Proceedings of the Fifth International Conference on IT Based Higher Education and Training, ITHET 2004

[6] Freeman, E., Gelernter, D. (1996), Lifestreams: A Storage Model for Personal Data, SIGMOD Record, vol. 25, no. 1, pp. 80-86.

[7] Ganley, B. (2004), Blogging as a Dynamic, Transformative Medium in an American Liberal Arts Classroom, Middlebury College, USA

[8] Gorton, I., Motwani, S. (1996), Issues in co-operative software engineering using globally distributed teams, Information and Software Technology, Elsevier Science, vol. 38, pp. 647-655.

[9] Gupta, A., Seshasai, S. (2004), Toward the 24-Hour Knowledge Factory, Massachusetts Institute of Technology Press.

[10] Hastings, P.B. (2004), Blogging Across the Curriculum: A Course Resource for Interactive Digital Design Program, Quinnipiac University

[11] Klempous R., Maciejewski H., Nikodem J., Gonzalez, M.G., Suarez, P. (2003), Informatics engineering curricula in the European Space Higher Education Area, ICSE 2003. Coventry, 9-11 September 2003.

[12] Sipper, M., Sanchez, E., Mange, D., et al (1997), A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-Inspired Hardware Systems, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, p83-97

[13] Szyperski, C. (1998), Emerging component software technologies - a strategic comparison, Software Concepts & Tools, Vol 19, No 1, p2-10.

[14] Warhurst C. (2001), Developing Students' Critical Thinking: the Use of Debates, Walker M. (Ed), Reconstructing Professionalism in University Teaching, Open University Press.

[15] Wilensky, U., Resnick, M. (2005), New thinking for new sciences: Constructionist approaches, San Francisco Press, CA

[16] Zachman, J.A., (1987), A Framework for Information Systems Architecture, IBM Systems Journal, Vol 26, No 3. IBM Publications.