

## Paul Ramsay

---

**From:** WWISA New Zealand Chapter [info@wwisa.org.nz]  
**Sent:** Tuesday, 17 February 2004 10:29 a.m.  
**To:** WWISA New Zealand Chapter  
**Subject:** WWISA New Zealand Chapter - Update Number 27

Welcome to the latest Worldwide Institute of Software Architects (WWISA) New Zealand Chapter newsletter.

In this issue we look at:

- Activities in Auckland
- Architecting the Gallagher Group
- IT architecture - art or science?
- Software development conference, and
- Email addresses and mailing lists.

### ACTIVITIES IN AUCKLAND

Our first meeting for the year in Auckland will be held on:

Thursday 19 February 2004

6:00pm for 6:30pm - 7:50pm

Peace Software  
100 Symonds Street  
Auckland

The speaker will be Richard Vowles, Solutions Architect, Borland New Zealand Limited (<http://www.borland.co.nz>).

Richard works with clients in the .NET and J2EE spaces, and has been working in distributed systems since the formation of a local special interest group with Borland's acquisition of CORBA vendor Visigenic in 1996.

These days more and more projects are spending time on application integration within or across the enterprise. Challenges especially exist when integrating across different platforms.

Richard will be exploring current practices and approaches in application integration, their benefits and shortcomings. We will also look at Borland "Janeva" which aims to simplify integration between Microsoft and Java/CORBA technologies.

Please feel free to extend an invitation to attend to any interested friends or colleagues (there is currently no cost involved with any chapter events).

This is a change in topic from that previously advised.

### ARCHITECTING THE GALLAGHER GROUP

When Peter Neil, Software Architect at the Gallagher Group, recently joined our mailing list, we asked him to us about his organisation and some of the architectural challenges they face. Peter kindly gave us this report:

Gallagher Group is a company best known for producing electric fencing products rather than for producing software. In 1999 Gallagher Group acquired Cardax (International) Limited, a company specialising in the production of access control systems (ie. systems which control personnel access through doors via card presentation and/or PIN entry), and with it they acquired a large software base supporting current and legacy products.

At the time of purchase Cardax was in the process of developing its next generation of access control system broadly termed Cardax FT (Future Technology) which comprised a performant and robust Windows-based administration engine, TCP/IP enabled intelligent field controllers and new range of RS485 field units (card readers, digital camera etc). As part of the acquisition of Cardax, the development group was required to relocate from their current base in Marton to the Gallagher Group headquarters in Hamilton, and with this relocation there was (not unexpectedly) a degree of attrition by the development staff. However, it was this attrition which opened the opportunity for me to join Gallagher Group to fill the roll as Software Architect for the Cardax FT product in January of 2001.

When I joined Gallagher Group as part of the Cardax FT development team, my predecessors had established the architectural framework which Cardax FT was to be developed under, and this framework is basically still in place in the current implementation of Cardax FT. The Windows-based administration engine (what we call our "head-end") is a three-tier application utilising SQL Server as the database engine, Windows Services as the middle tier and a rich Windows-based client. The services and client are written in Visual C++, strongly object-oriented and utilise COM for both intra and inter process communication. Our intelligent field controllers are written in C++, also object-oriented and utilise the Pharlap RTS for base services. Our field units are written in C and Assembler.

One of the immediate challenges I faced in joining the team was inconsistent design artefacts and, as a consequence, one of my early goals was to provide a consistent design approach through building a domain model of the Cardax FT head-end.

In previous roles I have found that comprehensive domain models provide a pivotal role in the ongoing maintenance and development of a system and I considered this would be valuable for the Cardax FT system as well. In these previous roles I have utilised Rational Rose as the design tool, in particular utilising the round trip engineering capabilities (as it seems to be only tool which handles the Visual C++ syntax adequately) and I looked toward using this tool for building the Cardax FT domain model.

I went through the process of getting an evaluation version of Rose, reverse engineered all of the projects which comprised our head-end and proceeded to put together sequence and collaboration diagrams of aspects of the system. However, as part of this exercise I made some observations of the Cardax FT architecture which made creating useful models difficult. Also during this process and additionally over the growth of the Cardax FT system in the past three years I have observed that the way of componentizing our system (through COM) would not best support the long term future development of the Cardax FT system.

The Cardax FT system was architected using strong object-oriented principals, however, the COM interfaces were not explicitly designed to provide modularity to the overall system. In Cardax FT the COM interfaces have not been used to provide consistent, functional demarcation points which has resulted in complex and inconsistent object referencing. This contributes to our ongoing maintenance of the system, because if the code in a COM component changes it is difficult to trace other users of this component in the Cardax FT system and what that effect that change will have. This has meant our testing department has taken the fall-back position of having to re-test the entire Cardax FT system for each release just in case a COM component which has changed is going to affect some other (unexpected) aspect of the system.

At this point in our development we feel that to better support the future development of Cardax FT we should progressively re-engineer the head-end to utilise a "service" oriented integration architecture rather than an object-oriented (COM based) one. By "service" we mean logical services rather than Windows services and is similar to the concepts outlined by Microsoft for their Indigo initiative (<http://msdn.microsoft.com/longhorn/understanding/pillars/indigo/default.aspx?pull=/msdnmag/issues/04/01/Indigo/default.aspx>).

With a shift to these services we would look to encapsulate a much greater level of common

functionality than our current COM interfaces currently do, thereby creating a more modular system. Because these services will encapsulate a logical grouping of functionality it will also be easier for us to describe the important aspects of the underlying architecture, both informally (eg. diagrams on a white board) and formally (through domain modelling). Another aim of these services will be that they are transactional, rather than relying on persistent object references, which will avoid the requirement for complex object referencing paths and therefore reducing our testing requirement for each release.

Part of the rationale for me signing up to the WWISA is to discuss the potential merits of a services oriented architecture versus other approaches. I would also like to discuss the merits various technology platform options (eg. .NET) and approaches for including these into existing software.

I look forward meeting some of you at the next WWISA meeting in Auckland on 19 February 2004.

## IT ARCHITECTURE - ART OR SCIENCE?

Ido Lelie gave a comprehensive presentation on "IT Architecture: Art or Science?" at our last meeting in Wellington.

With a title such as this, it was a wide ranging presentation but was centred around the "CAFRC" model developed by Philips as part of its Gaudí system architecting project (<http://www.extra.research.philips.com/natlab/sysarch>).

The model addresses three key architectural process - "understanding why, describing what and guiding how" - under the following categories:

- C - Customer objectives (what?)
- A - Application (how?)
- F - Functional (what?)
- C - Conceptual (how?)
- R - Realisation (how?)

A full copy of Ido's presentation can be found on the Chapter website at <http://www.wwisa.org.nz/downloads.asp> and includes:

- Introduction and overview
- The process of architecting
- Architecture vs. engineering
- CAFRC
- Types of models
- Integrating the views, and
- The most important driver.

His presentation was also covered in Computerworld New Zealand under the heading "SolNet architect sees more flexibility without Sun: IT architect often seen as meddler by engineers".

A full copy of the article can be found on the Computerworld website at <http://www.computerworld.co.nz/news.nsf/UNID/7AB19891CE667DD0CC256E2E0005BE93!opendocument>

## SOFTWARE DEVELOPMENT CONFERENCE

Don't miss this premier event - Software Education's "Software Development Conference 2004".

Held at the Duxton Hotel in Wellington on Monday 8 March 2004 - Wednesday 10 March 2004, this conference brings you some of the world's best thinkers and practitioners in the field of software development.

Keynote speakers include Tom DeMarco (principal of the Atlantic Systems Guild and author of the classic "Peopleware"), Larry Constantine (a pioneer of development best practice and usage-centred design), Rob Thomsett (an internationally respected authority on software development project

management) and Wolfgang Strigel (founder of the Software Productivity Centre and a leader in the area of software process improvement).

Check out the details on this special conference at <http://www.softed.com/sdc2004>.

## EMAIL ADDRESSES

With the ever increasing growth in spam, many Internet Service Providers and other organisations have implemented varying degrees of mail filtering.

Your right to privacy is respected at all times by WWISA which is why your email address is never disclosed in any mailing.

The drawback of this approach is that some mail filters may regard our mailings as spam.

One quick and easy way to prevent any inadvertent filtering is to save our email address (info@wwisa.org.nz) in your address book - that way you will continue to receive the information you've requested.

This may not work in every instance, in which case talk to your IT services team.

If you wish to be removed from our mailing list at any time, just send an email to info@wwisa.org.nz.

## BOOKMARKS

- Agile Data

<http://www.agiledata.org>

- Agile Modelling

<http://www.agilemodeling.com>

- Asilomar Institute for Information Architecture

<http://aifia.org>

- Enterprise Unified Process (EUP)

<http://www.enterpriseunifiedprocess.info>

- IBM "Ease of Use" Webpages

[http://www-3.ibm.com/ibm/easy/eou\\_ext.nsf/Publish/558](http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/558)

- OASIS

<http://www.oasis-open.org/home/index.php>

- User Interface Markup Language (UIML)

<http://www.uiml.org>

Don't forget to check out our chapter website (<http://www.wwisa.org.nz/Links.asp>) which includes links to a wide range of software architecture and related sites.

## QUOTABLE VALUE

"Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns - the ones we don't know we don't know."

- Donald Rumsfeld, US Secretary of Defense and 2003 Winner of the Plain English Campaign's "Foot in Mouth" Award

Kind regards ... Paul

Paul Ramsay  
WWISA New Zealand Chapter

info@wwisa.org.nz  
<http://www.wwisa.org.nz>

Phone: +64 4 494-3716  
Mobile: +64 27 444-9023

---

This e-mail message and any attachments contain information that is confidential and may be subject to legal privilege. If you are not the intended recipient, you must not peruse, use, pass on or copy this message or any attachments. If you have received this e-mail in error, please notify us by return e-mail and erase all copies of this message including any attachments. Thank you.

---