Pre-conference workshops

TUESDAY 29TH SEPTEMBER 2009

The following workshops run for a full day (from 09.30 to 17.30), with a short break in the morning and afternoon, and a lunch break at 13.00.

A Day of Enterprise Design Patterns in .NET

DAVE WHEELER & ANDY CLYMER

WORKSHOP REF: SA1

Design patterns can largely be categorised as tactical, such as those documented by the Gang of Four, and strategic. These more strategic, or enterprise, patterns address the broader concerns on how you architect and design entire applications; and it is these enterprise patterns that will be the focus for the day.

The day is divided into four sections.

We begin with an examination of the fundamental aspects of design patterns, examining practical issues such as the use of layers and tiers; the costs of distribution (and how to mitigate them); and the use of dependency injection to facilitate test-driven development and to ease concerns over coupling of components.

In the next section we delve into one of the more complex aspects of enterprise design: modelling the domain logic. We will look at the three most common approaches: Transaction Script, Domain Model and Table Module. Each has clear advantages and disadvantages, and these will be examined in detail.

Overcoming the friction that exists between the object oriented world, favoured by developers, and the tabular world of the database is one of the more fascinating challenges for the designer. In this third section of the day we examine this area in some detail, providing examples that range from the pure code approach to how you can use mapping technologies such as nHibernate and Entity Framework.

Our focus in the final section of the day is on user interface patterns, such as the various implementations of Model-View-Controller/Model-View-Presenter and the myriad of spin-offs that stem from these. We'll look at how these patterns can be best implemented within ASP.NET, WPF and Silverlight.

This is not just a day of pure theory; this is a practical analysis of the main enterprise design patterns that all architects and developers need to know, with a focus on how they are best implemented within a .NET application.

A Day of Unit Testing

SIMON HORRELL

WORKSHOP REF: SA2

Unit testing has become more prevalant over the last few years as more and more teams have come to realise the importance of ensuring code has some degree of test coverage. Developers are aware that code quality is an important issue and that to get high quality code they often need to refactor. Refactoring safely requires unit tests.

We will spend the day looking at unit testing, how to write a unit test, how to test external resources such as databases and web services, how to work with legacy code and how to ensure that your code is designed to be tested.

A Day of .NET 4.0

RICHARD BLEWETT

WORKSHOP REF: SA3

.NET 4.0 is coming, and brings with it many new features: from declarative WCF services to support for parallelised execution, from new functionality in the Entity Framework to full support for dynamic languages.

This pre-conference day takes you through all the major new elements of .NET 4.0, so that you can start planning on how this new functionality will affect your applications.

From Developer to Architect

SIMON BROWN & KEVIN SEAL

WORKSHOP REF: SA4

This session is an interactive introduction to software architecture and what it means to be a software architect. It's aimed at software developers who are looking towards their first software architect role, as well as architects that are new to the role.

You will gain:

- An understanding of what it means to be a software architect, and the role's responsibilities.
- An understanding of the trade-offs connected with architectural decisions.
- Some experience of what it feels like to be an architect, including gathering non-functional requirements, determining the drivers for architecture, and defining an architecture.
- An understanding that, as a software architect, it's okay to do some coding.

Combining presentation, group discussion and group working, throughout the day you'll be solidifying everything you learn by defining the architecture for a small software system. The overall goal is that you can take the experience gained here and apply it to your own projects.

No architecture experience is required, but software development experience is assumed.

Speakers

Dave Wheeler

Dave is an independent software consultant specialising in Microsoft .NET technologies. He's particularly fond of WPF and Silverlight, but to remain grounded in reality he also works extensively with ASP.NET and is a moderator on Microsoft's ASP.NET forums. When not writing software, Dave teaches various .NET courses for DevelopMentor, writes articles and speaks at conferences.

www.rocksolidknowledge.com

Andy Clymer

Andy has been working as a Software Engineer since 1993, using a variety of languages, technologies and operating systems. His last 'real' job was at Cisco Systems, where he was a lead architect for Cisco's identity solutions. Prior to Cisco he worked in various small startups, and after leaving he has focused on .NET technology, specialising in patterns and the chaotic world of multithreading/parallel programming. He teaches various .NET courses for DevelopMentor, including Guerrilla .NET and Patterns.

www.rocksolidknowledge.com

Richard Blewett

Richard has been working in the software industry for over 20 years, starting with mainframes through the early years of client/server to today's service-oriented world. He has spent his professional life working on large distributed systems including being the middle tier architect on the UK national police systems. He now focuses on technologies that enable developers to build large-scale systems on the Microsoft platform such as WCF, BizTalk, Workflow and Azure. He is a well-known conference speaker having spoken at TechEd, DevWeek, Software Architect and Oredev, and can often be found helping people on various newsgroups, mailing lists and web forums. Richard is also a DevelopMentor instructor where he writes and teaches material on WCF, Workflow, BizTalk, Azure, Oslo and a whole range of other technologies.

www.rocksolidknowledge.com

Simon Brown

Simon is a senior consultant at C5 Alliance in Jersey. He's a hands-on software architect with a BSc in Computer Science, and over the past 12 years he's been involved in projects ranging from rich desktop clients and web applications through to highly scalable distributed systems and service-oriented architectures; predominantly within the finance industry. He's also undertaken consulting and training roles with a broader focus on people, process and technology.

Conference sessions

WEDNESDAY 30TH SEPTEMBER 2009

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5
08.30	REGISTRATION & COFFEE				
09.30	**Opening Keynote** **Saving Software Architecture** TIM EWALD Lots of developers want to be architects. But for many, "architect" has become synonymous with "over 40 and over confident" or just plain "useless". Why do so many people feel this way? Where are architects going wrong, and what can they do to fix it? Houses, patterns, aspects, quality attributes, iron triangles, astronauts, committees, standards, Highlander, Nazgul, technologies, practices, skill sets, education, dynamic typing, testing, abstraction, and the nature of change all lead down a path to an answer that can, hopefully, help you design and build better software, and redefine what an architect should be.				
11.00	COFFEE BREAK				
11.30	**Architecting for the Cloud – design considerations for Windows Azure** RICHARD BLEWETT Cloud computing promises access to high levels of scalability and availability for the masses. Running your applications in someone else's data centre removes large amounts of upfront costs in rolling out software, allowing cost to be based on usage. However, as always, to gain benefit you have to ensure your applications are designed to allow the platform to manage your application effectively. This talk looks specifically at Windows Azure as a cloud computing platform, and discusses what impact deploying on to this platform has on how you design your applications.	**Are singletons evil?** ANDY CLYMER & DAVE WHEELER You've probably heard the comment that "Singletons are Evil" and also the counter-argument that they are completely necessary. This talk will be take the form of a debate between the two presenters, one presenting the reason that Singletons are necessary and the other why they are evil. Along the way we will cover the Singleton pattern, unit testing, IoC, Dependency Injection and the idea that Singletons are global variables. Hopefully we will end up with a clearer picture of how, when and where to use the Singleton pattern (or not!).	**Lean software architecture** KEVLIN HENNEY Agile software development covers a broad range of practices and ideals, but is often characterized (and even caricatured) in terms of just being a synonym for Extreme Programming or Scrum or of having little concern for architecture. The truth is invariably more subtle, and it is worth examining agility from a different perspective to see where architecture and process variation play a role. Such a perspective is provided by Lean Software Development, which takes the lessons of Lean Product Development and Lean Production and applies them to software development. This session explores the implications of Lean thinking on software architecture and the design choices and styles that best support a more sustainable approach to development.	**Broadening the T** SIMON BROWN A breadth of experience is key for successful software architects, but how do you get this and what should you know? In this technical session we'll look at the things every software architect should know, drilling down into the following topics: • Architecture and design patterns • Technologies, frameworks and tools • Designing for and testing non-functional requirements All of this will done in the context of the Microsoft .NET and Java technology stacks, so you'll get some concrete guidance on what knowledge will help you in your role as a successful software architect. This session is aimed at anyone with software development experience.	**MorganDirect – the client-side architecture of an extensible trading platform** KEVIN SEAL MorganDirect is a flexible user interface delivering foreign exchange, rates and options trading to employees and clients of J.P. Morgan. Built using Java and OSGi, MorganDirect offers a platform for deploying user-specific functionality while maintaining a responsive approach to development. This session will look at some of the design choices made in developing MorganDirect. In particular it will look at the Whiteboard Pattern and OSGi, and how they help deliver the desired modularity of a modern client platform, from simplifying pluggable components to providing consistent threading behaviour and supporting concurrent development efforts. This session should be suitable for developers and architects, particularly those with an interest in client-side development and OSGi.
13.00	LUNCH BREAK				

Simon is additionally active in wider aspects of the IT industry. He founded the 'Coding the Architecture' web site, and he has also written and co-written a number of books about Java EE web technologies, spoken at a various conferences, developed a software architecture training course, and actively maintains a public blog.

www.codingthearchitecture.com

Kevin Seal

Kevin is an enterprise Java architect and developer with over 12 years experience in the defence, internet, retail and finance industries. Through roles for various clients he has worked on desktop, web, service-oriented and middleware projects. Outside work he also helps run the 'Coding the Architecture' web site and London-based user group.

www.codingthearchitecture.com

Neal Ford

Neal is an application architect at ThoughtWorks, a global IT consultancy with an exclusive focus on end-to-end software development and delivery. He is also a designer and developer of applications, instructional materials, magazine articles, courseware, video/DVD presentations, and author of various books including 'The Art of Java Web Development' and the 2006 'No Fluff, Just Stuff' anthology. His primary consulting focus is the building of large-scale enterprise applications. He is a regular speaker at various conferences worldwide.

www.nealford.com

Kevlin Henney

Kevlin is an independent consultant and trainer based on the UK. He specialises in programming languages, OO design, patterns, development process and software architecture. He has been a columnist for various magazines and web sites, including Better Software, The Register, Application Development Advisor, Java Report and C/C++ Users Journal. Kevlin is co-author of 'A Pattern Language for Distributed Computing' and 'On Patterns and Pattern Languages', two volumes in the Pattern-Oriented Software Architecture series. He is also a regular speaker at various conferences, including Software Architect and DevWeek.

www.curbralan.com

Tim Ewald

Tim is a Senior Consulting Software Engineer at SeaChange International, the world's leading vendor of video-on-demand and ad-insertion systems for television operators. He is responsible for the architecture of the company's next-generation video delivery platform. Tim has 20 years experience building complex distributed systems on a range of technology stacks. He is an internationally recognized speaker who always tries to have something interesting to say.

www.schange.com

Eric Nelson

Eric has been with Microsoft UK for 13 years, and has spent most of that time helping ISVs to explore and adopt the latest technologies and tools from Microsoft. His current focus is working with ISVs to help them take advantage of .NET Framework 3.5 and IIS 7.0 in their solutions. Before joining Microsoft, Eric was a Unix developer working exclusively with C and 4GLs, writing high-performance calculation engines and delivering line of business solutions built on top of Sybase, Ingres and other RDBMS packages. He can be found blogging about "ISV Stuff" at:

geekswithblogs.net/iupdateable

Dino Esposito

Dino is a trainer and software consultant based in Rome. A member of the IDesign team, he specializes in Microsoft .NET technologies, and spends most of his time teaching and consulting across Europe and the USA. He has hands-on experience in architecting and building distributed systems for banking and insurance companies. A prolific author, Dino writes columns for various magazines, and has written a number of books on various cutting-edge topics. An ASP.NET MVP, he regularly speaks at industry conferences around the world, including TechEd, DevConnections and DevWeek.

weblogs.asp.net/despos

Steven Kelly

Steven is CTO of MetaCase and co-founder of the DSM Forum. He has over fifteen years of experience of tool building and consultancy in Domain-Specific Modelling. As architect and

Software Architect 2009

LONDON
29 SEP–1 OCT '09

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5
14.00	**Design considerations for storing data in the cloud with Windows Azure** **ERIC NELSON** The Microsoft Azure Services Platform includes not one but two (arguably three) ways of storing your data. In this session we will look at the implications of Windows Azure Storage and SQL Data Services on how we will store data when we build applications deployed in the Cloud. We will cover "code near" vs "code far", relational vs none relational, blobs, queues and more.	**Patterns dartboard** **ANDY CLYMER & DAVE WHEELER** Imagine a (virtual) dartboard. That dartboard has a selection of patterns pinned to it. Members of the audience can throw darts at the dartboard. Where the darts hit will determine how the talk goes. The talk will be about patterns, although which ones it's impossible to know – it's up to you to choose what goes into the talk! Will it be MVC? Factory? Command? Your darts! Your choice!	**Slicing design over time** **KEVLIN HENNEY** It is implicit in many agile approaches that design is treated as a continuous activity, rather than one exclusive to a single phase of development. From prototyping to TDD, from architecture envisioning to review, from refactoring to retrospectives, design is treated as a continuous – although sometimes lumpy – process of feedforward and feedback. On its own, however, this recognition of an unfolding and reactive approach to design does not help developers reason, discuss or envision how a design can grow in response to increasing coverage of scope, new information or other forms of feedback. There sometimes appears to be an air of mysticism surrounding the idea of growing and nurturing design – an unqualified optimism that assumes a leap of faith rather than a more intentional and empirical approach. There are many valuable aspects of pattern-based thinking that are overlooked in the common perception of design patterns. The original vision of patterns embodies a notion of incremental, feedback-based design – something that may come as a revelation to anyone who had mentally pigeonholed patterns together with heavier-weight design approaches. This session explores pattern concepts such as pattern stories and pattern languages with a view to helping agile developers reason about their designs.	**Using an MVC Framework to build REST Services** **TIM EWALD** REST APIs have gained a lot of traction recently. They are far simpler than SOAP and WS-* and uniquely well suited for Web and Cloud scenarios. Their increasing popularity has driven many traditional middleware vendors to add support for building them, but these solutions are almost always "bolted-on" and are not particularly satisfactory. Since REST is the basis for the Web, traditional MVC frameworks offer a much better alternative, with support for the core tenets of the REST style baked in. This talk explains how to design and implement a REST service using an MVC framework, including mapping processes to resources, resource representations, and using linking and forms to increase loose-coupling.	**Visualizations for code metrics** **NEAL FORD** Judicious use of metrics improves the quality of your code. But interpreting metrics presents a challenge. You have a list of numbers for a project – what does it mean? This session shows how to produce visualizations for software metrics, making them easier to understand and more valuable. It covers metrics at the individual method level all the way up to the overall architecture of the application. This isn't a talk about how tools produce visualizations: this session shows how to generate your own visualizations, allowing you to customize it to the level in information density that shows real value.
15.30	COFFEE BREAK				
16.00	**Writing REST-based Systems with WCF** **RICHARD BLEWETT** Traditionally people have thought of services as the realm of SOAP. However, many large web-based organisations (Google, Yahoo and Amazon among them) have chosen another style of writing services: Representational State Transfer, or REST. Microsoft has also exposed much of its Azure infrastructure via REST. So what is it about this architectural style that has led these huge organisations to adopt it? This talk looks at how you can use WCF to build REST-based systems to give you architectural choices when using Microsoft's main communication platform.	**Coding a solid user interface** **DAVE WHEELER** Microsoft Visual Studio 2008, in keeping with its predecessors, is a wonderful development environment that favours rapid application development. However, in the hands of naïve developers, it can lead to rapid generation of exceedingly smelly code, particularly in the user interface. This session will show you how to truly build maintainable and testable user interface code, covering everything from adopting the Command pattern, to the correct use of threads, to implementing one of the many varieties of Model-View-Controller. A key feature of this session is its practical mixture of code and design patterns.	**Design for testing** **SIMON HORRELL** Many developers now recognize the importance of unit testing, and either use Test First or Test Driven development. However, many developers and architects are still struggling to come up with designs that enable easier testing of their applications. This talk will examine some common unit testing problems and use various unit testing patterns to solve those problems.	**Top 10 software architecture mistakes** **EOIN WOODS** Today, the role of the software architect is widely recognised, and over the last few years, we've also seen a welcome increase in the number of useful books, websites and blogs aimed at the aspiring software architect. However, the success of a project is still usually related to the amount of hard-won experience that the project team has – by which we really mean how many mistakes they've learned from so far! Experience is a hard but very effective teacher. In this talk I'll share some of the pitfalls that I've encountered in my experience as a software architect and in doing so, hopefully make sure that you avoid making just the same mistakes on your own projects. This session is aimed at new or moderately experienced architects, developers and project managers, although those with more experience will probably find it valuable too.	**Emergent design & evolutionary architecture** **NEAL FORD** Most of the software world has realized that BDUF (Big Design Up Front) doesn't work well in software. But lots of developers struggle with this notion when it applies to architecture and design. Surely you can't just start coding, right? You need some level of understanding before you can start work. This session describes the current thinking about emergent design & evolutionary architecture, including both proactive (test-driven development) and reactive (refactoring, composed method) approaches to discovering design. The goal of this talk is to provide nomenclature, strategies, and techniques for allowing design to emerge from projects as they proceed.
17.30	DRINKS RECEPTION				

lead developer of MetaEdit+, he has seen it win or be a finalist in awards from SD Times, Byte, Innosuomi, Net.Object Days, and Jolt Productivity. He has authored a book and over 20 articles in journals such as Dr. Dobb's, and regularly speaks at events like OOPSLA and SD Best Practices. Steven is a member of the IASA, on the editorial board of JDM, and a full-back in the Finnish 3rd division.

www.metacase.com

Eoin Woods

Eoin is a software architect at Barclays Global Investors, responsible for a new equities portfolio management system being built for the company's well known active management group. Prior to BGI, Eoin has worked in the software engineering field for over 15 years, for a number of companies including Groupe Bull, Sybase, InterTrust and UBS Investment Bank. His main technical interests are software architecture, distributed systems, computer security, and data management; he is co-author of the book 'Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives', published by Addison Wesley.

www.eoinwoods.info

Mark Dalgarno

Mark has worked in the software industry for over twenty years at all levels, from programmer to head of software development. He works for Cambridge-based Software Acumen – a software consultancy specialising in Architecture Management, Model-Driven Software Development and Software Process Improvement. He is also the editor of the Model Driven Software Network and organiser of the associated Code Generation and Practical Product Lines conferences. Mark regularly speaks at various events including SPA, ACCU, Developer Day Scotland and SPLC. In his spare time he tries to spend less time in front of computers and more time outdoors.

www.software-acumen.com

Conference sessions

THURSDAY 1ST OCTOBER 2009

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5
08.30	COFFEE				
09.30	**What's new in Windows Workflow Foundation 4.0** **RICHARD BLEWETT** Windows Workflow Foundation is having a major overhaul in the next version of .NET. This session looks at the drivers that pushed the team to take radical action, what the new features are, and what it means for your WF 3.5 applications today.	**Pattern connections** **KEVLIN HENNEY** Patterns are often considered and presented in isolation or as part of a loosely themed catalogue. However, much of their power comes from connecting them together to describe whole systems or frameworks and how to build them. This session revisits core pattern concepts, using examples to illustrate trade-offs involved in favouring one design over another and how the appropriateness of a pattern is sensitive to the context in which it is applied. Examples are presented using UML and code in various languages.	**High performance SOAs** **JESUS RODRIGUEZ** Achieving sustainable levels of performance is one of the biggest challenges of Service Oriented (SOA) solutions nowadays. Are you wondering how to scale your SOA to tens of thousands of applications? How to prioritize the messages from the different applications? How to correctly handle downtimes and failures? How to implement resource intensive capabilities such as transactions or stateful messaging without sacrificing my performance? How to monitor your SOA infrastructure and control the service level agreements (SLA)? Correctly addressing those questions can very well dictate the success or failure of your initiatives towards SOA. This session summarizes the lessons learned in the implementation of a number of large scale, high performance service oriented solutions. The session will illustrate a set of architecture patterns and techniques to enhance important SOA aspects such as performance, scalability and reliability also encompassing other features such as long running transactions, security, federation or stateful messaging. Additionally, we will provide practical implementation techniques for these patterns using technologies such as Windows Communication Foundation 3.5/4.0, Windows Workflow Foundation 3.5/4.0 and the Windows Application Server (Dublin).	**Architecting for multi core** **ANDY CLYMER** The free lunch is over, applications don't get a massive performance boost as new CPUs arrive. In order to take advantage of the increased power, applications need to be architected to take advantage of the multiple cores, this requires a shift in the algorithms and techniques we employ. This talk will introduce a variety of parallel processing patterns utilising Microsoft's Parallel Framework Extensions, soon to be included as part of .NET 4.0.	**"ConFront" – multi-solution ("sharded") .NET development using Authenticating Front Controller Rewriting Proxy at Confused.com** **PHIL MURPHY** When developing large, complex, multi-feature websites, the prescribed ASP.NET approach has been either to develop one all-encompassing .NET deployable (which can lead to problems of managing source branches, results in big-bang deployments, and requires pan-site regression testing), or by subfolders in IIS (which couples the url structure too tightly to the folder structure, and cannot share authentication/session information). This talk details a Third Way, devised by Phil and Kevin while consulting at Confused.com (dubbed "ConFront"), and currently underpinning Confused.com's very busy web site. We feel the technique could usefully be applied to other large, multi-area web applications, with many benefits.
11.00	COFFEE BREAK				
11.30	**What impact will Entity Framework and Entity Data Model have on application architecture** **ERIC NELSON** The Entity Framework is a strategic technology from Microsoft. In version 1 the focus was on helping developers use an ORM within their application and to expose data as RESTful Web Services, but version 2 extends this to include scenarios such as reporting. In this session we will look at what is the impact, both good and bad, of embracing an ORM technology and specifically, the impact when that technology is the ADO.NET Entity Framework.	**Objects of desire** **KEVLIN HENNEY** Given that object orientation underpins modern programming languages, it would be expected that anyone using languages such as C# or Java would also know about OO and that, for the architect, application of OO techniques should be a minor issue. Although object orientation is far from being a new approach (think Sergeant Pepper), and it's also far from being obscure, its mainstream adoption by programmers and programming languages alike is not always as effective as it could be, and sometimes the results can be less than desirable (to put it politely). Good programming and design practice has architectural implications, as does poor practice. This session aims to put a number of concepts practitioners may be unsure of on a firmer footing, highlighting common pitfalls in OO practice – such as abuse of inheritance, simplistic getter/setter interfaces, patternitis, use of singleton objects – and emphasising techniques that promote an easier life and reduce architectural friction, such as loose coupling, unit testability, pattern-based design thinking, incremental and sufficient design.	**How good are you at .NET software design?** **DINO ESPOSITO** Just as an architect wouldn't design a house ignoring the laws of gravity, a software architect shouldn't design a piece of software ignoring basic principles of software design such as low coupling, high cohesion, separation of concerns, information hiding. In this session, we'll first reconsider the fundamentals of software design, and then move on to review the basic principles of object-oriented design including an analysis of the pros and cons of inheritance and composition. Finally, we touch on a few more advanced principles of OOD such as dependency inversion and the substitution principle. If you're really a good software architect nothing in this session should come as a surprise. But you can only find out if you come!	**Great-looking models, and how to make them work for you** **STEVEN KELLY** We've all had bad experiences with models: you work with them for hours but get no concrete results, or you have an early spurt of enthusiasm but tire quickly. In many cases, it's a problem of communication. In this session we'll look at how to get everybody speaking the same language – the practical, repeatable steps to create your own domain-specific modeling language or DSL. The steps are incremental, solving what is for many the hardest part: starting from a blank screen to come up with something that isn't just a variant of UML, but will actually raise abstraction and productivity.	**Rich Internet Applications for architects** **DAVE WHEELER** Rich Internet Applications, or RIAs for short, have seemingly become the latest and greatest form of application. With two major contenders, Microsoft Silverlight and Adobe Flex, currently fighting it out for top spot, it can be tough to decide which one you should choose. The story gets even more convoluted when you factor in runtimes such as AIR that enable RIAs to run outside of the browser. This session will help you to understand what you can, and cannot, do with a RIA; what the main differences (and similarities) are between Silverlight and Flex; what the impacts are to your overall system architecture; the challenges that your developers are likely to face; and what specific security issues you need to consider.
13.00	LUNCH BREAK				

Software Architect 2009

LONDON
29 SEP–1 OCT '09

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5
14.00	**Contract first design with WCF** **RICHARD BLEWETT** In WCF the contract that a service exposes is defined by code. However, is that the best starting point? If you start with the code and annotate your types, how do you know if you haven't inadvertently exposed your type system internals to service consumers? Design by contract is a different way of approaching service design. This starts by stating what functionality will be exposed at the service and expected by the client. This model means you have an implementation independent form of the contract: perhaps WSDL or a Domain Specific Language and then you generate the client proxy and the service skeleton from this abstract definition. This approach is more robust, particularly in interop scenarios. In this talk we will look at how far the WCF tooling can take you down this path and what we have to fix up any shortcomings. We will also look at other tools that start from a design by contract premise.	**Modelling in the age of agility** **KEVLIN HENNEY** The practice is of modelling is often associated with heavyweight UML diagrams that are drawn up with the best intentions, but often either leave their readers confounded or are simply left to one side while the other activities in development proceed apace. Modelling has been associated with plan-driven approaches and big up-front analysis and design, at odds with the emphasis of agile approaches. There is, however, another side to modelling that deserves the attention of anyone involved in development, whether they adopt an agile mindset or not. Modelling is not the preserve of plan-driven methods, and the problem often lies not with modeling per se but with overdosing on models and failing to use modelling as an opportunity for communication. Models that are drawn up by individuals in isolation from one another are often the culprit. Often the secret to effective modelling is more in the -ing than the model.	**Reconstructing software architectures** **MARK DALGARNO** 'Inheriting' software you don't really understand can be a painful experience. Whether it's a big ball of mud, spaghetti or just plain weird we all will typically have to work with such software at some point in our careers. This session describes some practices for systematically approaching the task of reverse engineering such software to reconstruct its architecture and so help reduce the pain of working with it. Along the way we'll pick the brains of audience members by way of group exercises looking at different aspects of the reverse engineering problem.	**From SOA to WOA: introducing Web Oriented Architectures** **JESUS RODRIGUEZ** Despite being one of the latest acronyms in the never-ending distributed programming dictionary, Web Oriented Architectures (WOA) represents one of the most important architecture styles in service orientation. In a nutshell, WOA incorporates the principles of the Web and Representational State Transfer (REST) into Service Oriented Architectures (SOA) to achieve high levels of performance, availability and scalability. This session explores the fundamental principles and patterns of Web Oriented Architectures (WOA) and how they can enhance the way we build SOA solutions today. Specifically, we will demonstrate how REST enables a new set of patterns around key SOA principles such as service composition, message brokering, routing, publish-subscribe messaging, load distribution and long running transactions. Finally, we will compare and contrast WOA with the traditional patterns we use to build services today in order to help architects and developers to select the right architecture style for the right scenario.	**Communication skills for geeks** **NEAL FORD** Software is fundamentally a communications game, and good communication skills differentiate between good and great developers. This session describes communication techniques and skills to people who skipped English 102 to hack some code. I talk about effective communication techniques for presentations, documentation, memos, and how to sell your technical ideas to a non-technical crowd.
15.30	COFFEE BREAK				
16.00	**Designing UIs with the Composite Application Guidance** **DAVE WHEELER** The Composite Application Guidance, AKA "Prism", offers a significant breakthrough in how you design and build user interfaces for Windows Presentation Foundation and Silverlight. In this session, you will gain a solid understanding of the core features of Prism along with its main benefits and weaknesses. You will also get a detailed understanding of the impacts that using Prism will have in your overall application architecture, such as exposing data-bound commands from your ViewModels, its intrinsic support for TDD, the role of DI within the UI, and the use of Prism's pub-sub event architecture. This is a "must see" technology for any serious WPF or Silverlight architect/developer.	**Test-driven design** **NEAL FORD** Most developers think that Test-driven Development (TDD) is about testing, but testing is only a small benefit from using TDD techniques. This session demonstrates how stringent TDD improves the structure of your code. I discuss TDD as a technique for vetting consumer calls, using mock objects to understand complex interactions between collaborators, and some discussions of improved code metrics yielded by TDD. This session shows that TDD is much more than testing: it fundamentally makes your code better at multiple levels.	**ASP.NET MVC vs. ASP.NET Web Forms** **DINO ESPOSITO** It risks becoming an endless debate: should you use ASP.NET Web Forms or the newest ASP.NET MVC to build your next application? The simple answer is that there's no winner and no loser. The frameworks are functionally equivalent and just bring two different philosophies into ASP.NET development. So it is mostly a matter of preference, but it is also an architectural decision not to be made lightly. In this session, we'll compare and contrast the two frameworks from a number of different perspectives including reporting capabilities, data entry, AJAX, data binding, HTML, styling, and more.	**REST in the real world** **JESUS RODRIGUEZ** Are you wondering how other companies are leveraging Representational State Transfer (REST) in their Service Oriented (SOA) solutions? What are the lessons they are learning and the challenges they are facing? This session illustrates how a large airline is taking advantage of REST in order to enable their next generation solutions. The session explores in detail a large number of scenarios and challenges faced in key areas like service modeling, caching, error handling, resource categorization, security, management and governance, among many others. Additionally, we demonstrate how the capabilities and extensibility model of Windows Communication Foundation (WCF) and the REST Starter Kit helped to address those challenges and enable a true RESTful distributed environment. Finally, we will emphasize the best practices, techniques and lessons learned that developers can follow in order to take advantage of RESTful services in their organizations.	**Documenting your software architecture – why and how?** **SIMON BROWN** A description of your software architecture is essential for any project of any size, explaining the rationale behind the design decisions and how the software will work in the real world. Such descriptions should include an explanation of the software structure, the architectural principles adopted, the constraints in force, the platforms on which the system is deployed and an explicit justification of how the architecture satisfies the requirements. A good software architecture document should describe what the code itself doesn't. This session will tell you how to do this. This session is aimed at anyone with software development experience.
17.30	END OF CONFERENCE				

Jesus Rodriguez

Jesus is the Chief Architect of Tellago, Inc. He is also a Microsoft BizTalk Server MVP, an Oracle ACE, and a member of the Microsoft Connected Systems Advisor team, as a member of which he has participated in various software design reviews with Microsoft's product teams, including Windows Communication Foundation, Windows Workflow Foundation and BizTalk Server. He has extensive experience with business process integration and messaging through numerous implementations of disparate systems founded on the principles of SOA and BPM. He contributes articles to various publications, and is a regular speaker at a number of conferences. Jesus is a prolific blogger on all subjects related to integration.

weblogs.asp.net/gsusx

Phil Murphy

Phil is a consultant architect/lead developer, specialising in Microsoft .NET technologies, typically delivered using Agile processes. He has 16 years of experience in analysis, architecture and development. His role has broadened considerably over time to encompass business analysis, user experience design, interface design, information architecture, systems architecture, pre-sales and training. Phil was recently part of a team that helped Confused.com migrate their entire web architecture from JSP to ASP.NET MVC, using a groundbreaking flexible architecture to deliver a newly-designed motor insurance quotation system in ASP.NET, while seamlessly proxying a suite of legacy JSP micro sites.

nvelop.co.uk