

Modelling Systems-of-Systems in UML

Software Architect 2008

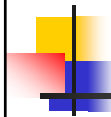
Eoin Woods
Barclays Global Investors

www.barclaysglobal.com/careers
www.eoinwoods.info



Introductions

- I'm a software architect at Barclays Global Investors
 - responsible for Apex, a new portfolio management system for Active Equity
 - also involved in regional technology architecture, trading systems and global architecture council
- Software architect for ~9 years
 - With enterprise architecture for about 2 years
- Author of "*Software Systems Architecture*" book with Nick Rozanski
- IASA Fellow, BCS and IET member



Acknowledgements

- A lot of the work underpinning this presentation was completed when I worked for UBS Investment Bank
 - primarily during my role as a stream architect for ETD-IT
- Andrew Ward, lead architect for ETD-IT, was tremendously interested in and supportive of this work
- Many other UBS employees, particularly those working on ETD-IT's integration programme, directly or indirectly contributed to, critiqued or used it



Content

- Introducing Systems-of-Systems
- Modelling Systems-of-Systems
- Using and Extending UML for Systems-of-Systems Models
- A Profile for Systems-of-Systems Models
- Using a Custom Profile Effectively
- Summary and Conclusions



What are Systems-of-Systems?

- Most people build single systems
 - single cohesive focus, single architect and development team
 - one stakeholder group (albeit often diverse)
 - single point of management control
- Enterprise architects and others create SoS
 - systems where the components are themselves systems
 - the individual systems are operationally independent
 - each system has its own point of management control
- Systems-of-systems are common in
 - large defence / C3I type projects
 - internet scale systems
 - large enterprises (e.g. enterprise application integration)
 - service oriented architectures

v1.20080519

(C) Eoin Woods 2008

5




Systems-of-Systems Challenges

- Understanding the whole without drowning in details
- Keeping track of current state in the face of change
- Analysing the whole to understand it
 - predicting the impact and cost of change
 - fault finding
 - SOX, compliance, ...
- The lack of a single expert who knows it all
- The scale of any description you attempt to create
- The diversity of the environment
 - technology, architectural style, system function, ...

v1.20080519

(C) Eoin Woods 2008


6



Content

- Introducing Systems-of-Systems
- Modelling Systems-of-Systems
- Using and Extending UML for Systems-of-Systems Models
- A Profile for Systems-of-Systems Models
- Using a Custom Profile Effectively
- Summary and Conclusions

v1.20080519 (C) Eoin Woods 2008 7



Systems-of-Systems Modelling

- We use models to understand single systems
 - Comprehension of the system's mechanisms
 - Capture design structures not obvious from the code
 - Allow estimation and analysis of properties
 - Allow planning for construction and change
- How can models help with systems-of-systems?
 - provide a comprehensible overview
 - act as a reference source to support analysis
 - provide a queryable catalogue of the SoS's content
 - capture information that no one knows or isn't available elsewhere (e.g. traceability between systems)

v1.20080519 (C) Eoin Woods 2008 8



SoS Modelling Approaches

- Boxes and lines
 - ad hoc notations captured in Visio and PowerPoint
- UML
 - informally in Visio
 - semi-formally in Magic Draw, RSM, Poseidon, ...
- Database of elements and connections
 - Excel, Access, Oracle, ...
- Special purpose SoS or EA notation
 - e.g. Archimate + BizDesign Architect

v1.20080519

(C) Eoin Woods 2008

9



SoS Modelling with Boxes and Lines


- Ubiquitous approach (+ve)
- Very flexible, tailorable (+ve)
- Accessible for writer and reader (+ve)

- Can't be queried once created (-ve)
- Redundancy between similar views (-ve)
- No link to associated definitions (-ve)
- Typically quite an ambiguous result (-ve)

v1.20080519

(C) Eoin Woods 2008

10




SoS Modelling with UML

- Widely used in technical circles (+ve)
- Semi-formal and extensible language (+ve)
- Unifies definitions and the pictures (+ve)
- Little redundancy between similar views (+ve)

- No systems-of-systems support in the language (-ve)
- Tools are typically needed to use it effectively (-ve)
- Need to learn the language is a barrier to entry (-ve)

v1.20080519 (C) Eoin Woods 2008 11



SoS Modelling Using a Database

- Result is directly queriable with conventional tools (+ve)
- Data and query results are precise (+ve)
 - although not necessarily correct!
- SQL interface provides easy integration and reuse (+ve)

- Entering the data is awkward and needs a tool (-ve)
- No graphical representation (-ve)
 - unless you write one
- Needs database development / usage skills (-ve)

v1.20080519 (C) Eoin Woods 2008 12



Special Purpose SoS Languages

- Good example of a language for enterprise SoS modelling would be Archimate
 - result of a Dutch-led EU research project
- Notation designed specifically for this problem (+ve)
- Provides a model rather than pictures (+ve)
- Unfamiliar notation to most people (-ve)
- Limited range of tools available (-ve)

v1.20080519

(C) Eoin Woods 2008

13



Choosing the Modelling Approach

- Each approach can be used effectively
 - depends on situation, needs and desire to make it work
- **Boxes and Lines** doesn't generally provide enough benefit for the cost due to the lack of a real model
- **Database** of information is expensive to develop and the lack of graphical representations is difficult
- **Special purpose notations** like Archimate are promising, but not widely understood or well supported
- Lack of a better option means that I've found **UML** to be the best place to start
 - but its *extensibility* is the key to its effective use

v1.20080519

(C) Eoin Woods 2008

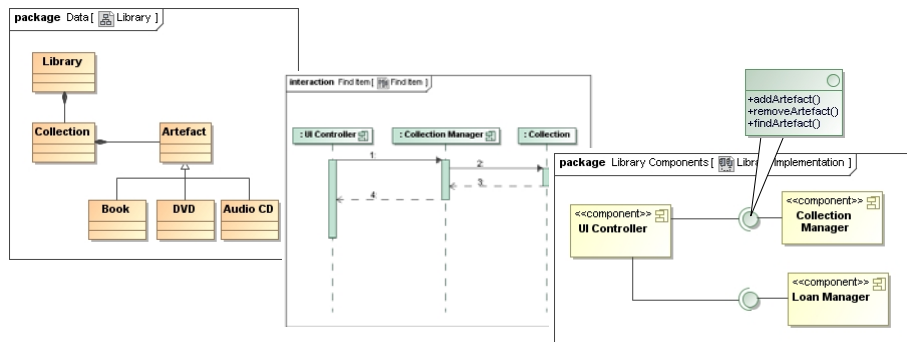
14


Content

- Introducing Systems-of-Systems
- Modelling Systems-of-Systems
- Using and Extending UML for Systems-of-Systems Models
- A Profile for Systems-of-Systems Models
- Using a Custom Profile Effectively
- Summary and Conclusions

UML and SoS Modelling

- UML is an object-oriented software design notation
 - class, interface, components, association, sequence diagrams showing method calls, ... all software design ideas






Talking Point – Missing Types

- What kinds of model element do you want to capture for SoS models that aren't in UML?
 - applications?
 - middleware?
 - hardware?
 - interfacing?
- Try to sketch a class model of them

v1.20080519 (C) Eoin Woods 2008 17



UML and SoS Modelling

- SoS modelling isn't well supported by UML
 - in fact, neither is software architecture modelling
- Some of UML's built-in elements can be used
 - e.g. component, interface, dependency, interaction
 - quickly show their heritage though (e.g. interfaces of methods)
- Some common SoS concepts not found in UML include
 - desktop application, server application, hosted application, ...
 - messaging, queues, pub/sub, ...
 - databases, file systems, ...

v1.20080519 (C) Eoin Woods 2008 18



UML and SoS Modelling

- UML may not support SoS modelling well, but it has a saving grace - it's **extensible!**
- The **UML** language is itself defined using **UML**
 - the "UML meta model"
 - defined in "infra structure" and "super structure" specifications
- The **meta-model** is just a UML model can be extended by the user of the language via "**UML profiles**"
- A UML Profile is a set of UML classes that extend the standard UML meta-model
 - usually the super-structure specification parts

v1.20080519

(C) Eoin Woods 2008

19



Extending UML

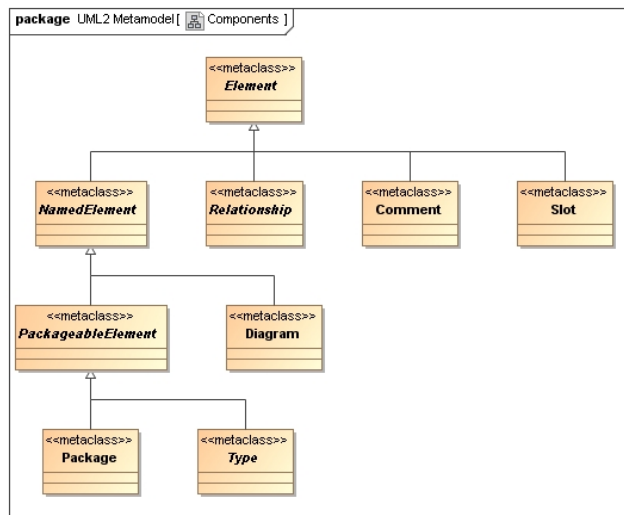
- A UML profile allows the meta-model to be extended
- This can add extensions ("stereotypes") of any of the primitives in the base language
 - new types of class, component, actor, ...
 - new types of relationship
 - add mandatory / predefined features like attributes
- Element stereotypes can have a custom icon for display
 - just as UML "component" has a different icon to "class"
 - can't do this for relationships (no icon, just a line style)
 - various options for using the icon for display
- Facilities and ease of use do vary by tool!

v1.20080519

(C) Eoin Woods 2008

20

A Fragment of the UML Meta Model

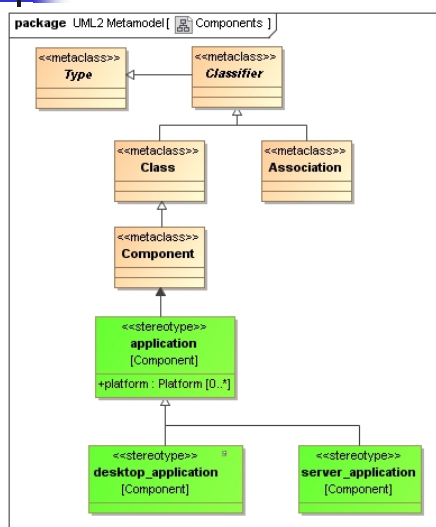


v1.20080519

(C) Eoin Woods 2008

21

Simple Extension to the Meta Model



v1.20080519

(C) Eoin Woods 2008

22

- Elements in green are stereotypes extending the language
- Adds three new types of components to represent new types of systems
- Adds a meta-attribute "platform" for them (of type Platform, which is an enum)
- Result is three "stereotypes" that can be applied to model elements of type "component"



Extending UML

- Once defined, stereotypes can be used like native UML element types
 - some tools like Magic Draw allow stereotype driven tailoring
- Profiles of stereotypes define new languages in UML
 - already used for DODAF, SysML and others
- I've used a profile to create a simple enterprise architecture or systems-of-systems modelling language
 - originally done in RSM, now ported to Magic Draw

v1.20080519

(C) Eoin Woods 2008

23



Talking Point – Extending UML

- Can you take the output of the previous exercise and relate it to UML?
- What sorts of UML element could you specialise for each missing type you identified?
 - a class?
 - a component?
 - a package?
 - an actor?
 - a relationship? (what sort?)
 - a dependency? (what sort?)
- What custom attributes or constraints would you add?

v1.20080519

(C) Eoin Woods 2008

24



Content

- Introducing Systems-of-Systems
- Modelling Systems-of-Systems
- Using and Extending UML for Systems-of-Systems Models
- A Profile for Systems-of-Systems Models
- Using a Custom Profile Effectively
- Summary and Conclusions

v1.20080519

(C) Eoin Woods 2008

25



A Profile for SoS Modelling

- Some concepts I needed were missing from UML
 - Server application and subtypes (e.g. hosted vs. shared)
 - Desktop application
 - Interfaces based on files and messaging
 - Information flow between applications
 - Standard attributes of applications and flows (owner, status, ...)
- Having wasted time with boxes and lines and struggled with base UML I created a UML profile for my SoS models
 - allowed easier, more consistent modelling
 - allowed checking if I wanted it
 - allowed me to publish the resulting model
 - allowed me to write queries over the resulting model

v1.20080519

(C) Eoin Woods 2008

26

Preconditions for Success

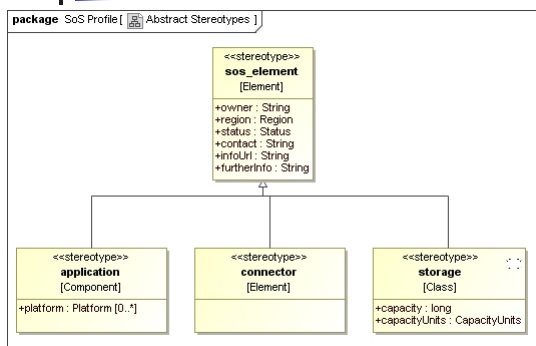
- Good understanding of base UML and some enthusiasm for it and learning more
- A software tool that supports profiles well
 - Magic Draw makes it very easy
 - RSM makes it possible, with some awkwardness
- A software tool that allows model data to be extracted
 - via a query interface or a *simple* programming model
 - OCI, Magic Draw OpenAPI, Rational's EMF implementation, ...
- Enough time to develop and refine the profile
- Ideally a real problem to work on

v1.20080519

(C) Eoin Woods 2008

27

Stereotypes in the Profile



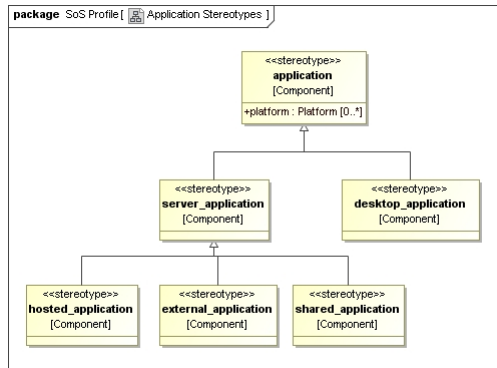
- An abstract sos_element stereotype to capture the attributes I need to have for all SoS model elements
- Three abstract stereotypes for applications, connectors and storage

v1.20080519

(C) Eoin Woods 2008

28

Application Stereotypes



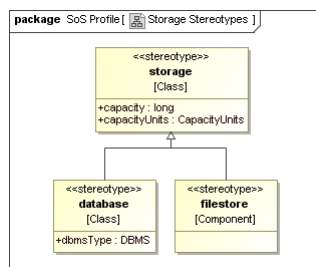
- Additional subclasses of “application” to indicate the different sorts of application in the environment

v1.20080519

(C) Eoin Woods 2008

29

Storage Stereotypes



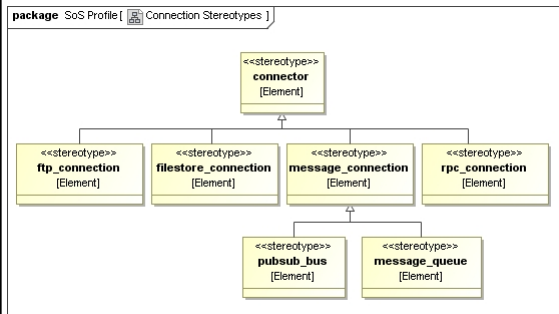
- An abstract “storage” type with capacity attributes
- Stereotypes for DBMS and file store

v1.20080519

(C) Eoin Woods 2008

30

Connector Stereotypes



- Types of “connector” to link applications
- file store and messaging connectors distinct from elements
 - allows modelling either way
- RPC connection covering RMI, WS-*, REST, ...
- Can be applied to elements *or* relationships
 - Again for flexibility
- Usually used with information flows

v1.20080519

(C) Eoin Woods 2008

31

Using the Stereotypes

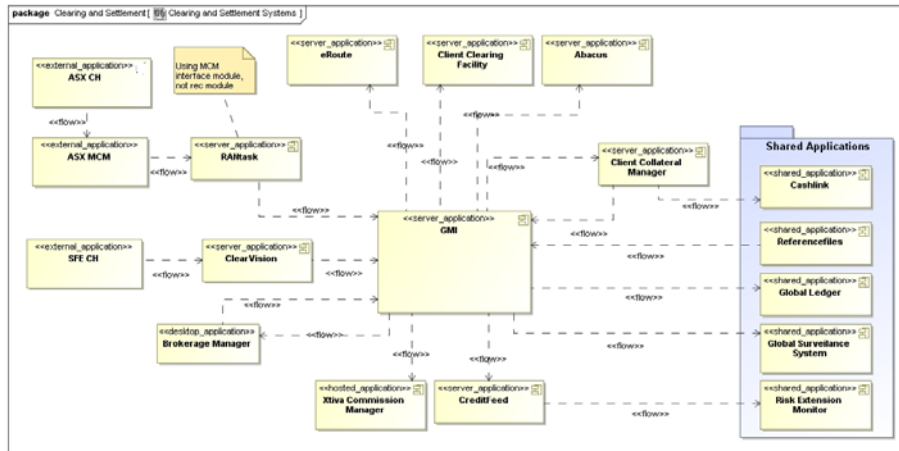
- Create components for the systems in the environment
- Stereotype them with the closest type from the profile
 - or create a new type if appropriate
- Link them with UML2 information flows if available
 - RSM doesn't have these so use dependencies instead
 - if you have them, link to a domain model for conveyed types
- Review the connections and refine using the profile
 - use a profile stereotype to “tag” the connection type
 - OR
 - introduce one or more elements that “implement” the connection (e.g. a queue or a database)
- Make sure you capture the custom attribute values

v1.20080519

(C) Eoin Woods 2008

32

Example Basic Profile Use

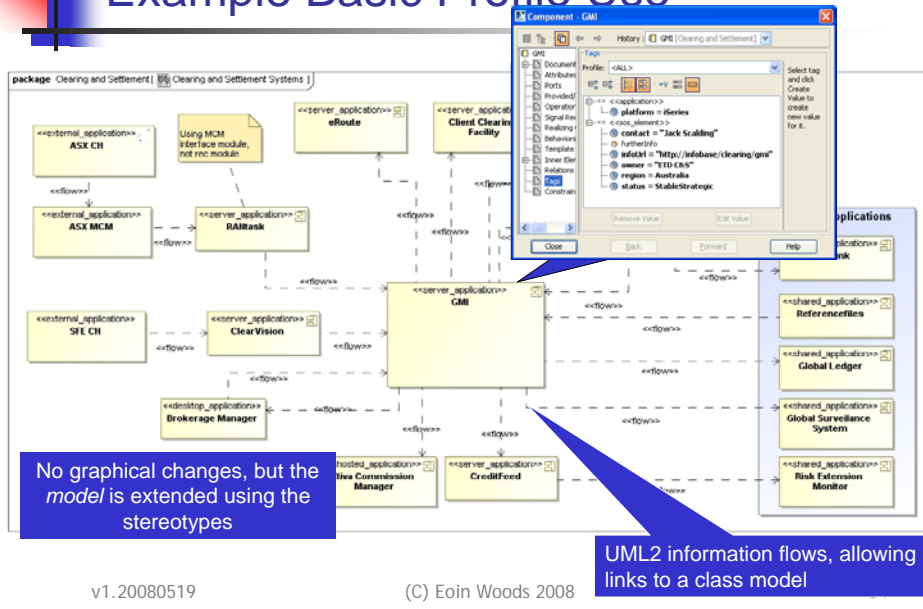


v1.20080519

(C) Eoin Woods 2008

33

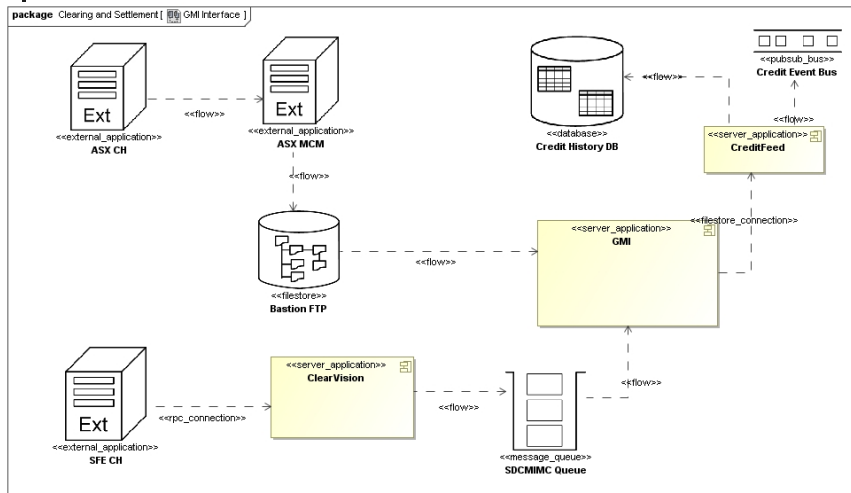
Example Basic Profile Use



v1.20080519

(C) Eoin Woods 2008

Fuller Use of Profile

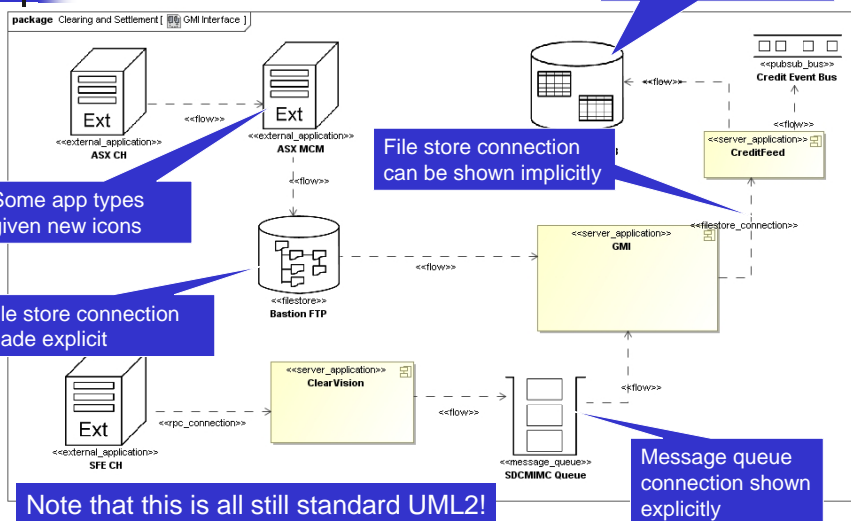


v1.20080519

(C) Eoin Woods 2008

35

Fuller Use of Profile



v1.20080519

(C) Eoin Woods 2008

36



Results of Applying the Profile

- A model containing elements specialised to the type of model we're building
- All elements picked from a standard, type safe palette
 - this means you can find them again with confidence
 - no problems with "dbms" vs. "database" conventions
- Elements from the profile all contain standard attributes
 - owner, region, information url, ... things I've needed to collect
 - more specialised types have additional values (e.g. storage size)
- Graphical presentation can be enhanced with icons
 - results in a "rich picture" like look, but in the UML2 framework
- A queryable database of elements (i.e. the model)

v1.20080519

(C) Eoin Woods 2008

37



Demonstration

v1.20080519

(C) Eoin Woods 2008

38



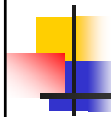
Content

- Introducing Systems-of-Systems
- Modelling Systems-of-Systems
- Using and Extending UML for Systems-of-Systems Models
- A Profile for Systems-of-Systems Models
- Using a Custom Profile Effectively
- Summary and Conclusions



Using the Profile – Tool Requirements

- This approach has been implemented in Magic Draw and Rational Software Modeller (RSM)
- It can undoubtedly be done in other UML tools too
- The key requirements for the tool are:
 - Full support for UML2 including profiles
 - Easy authoring, application *and modification* of profiles
 - A reporting engine allowing selection of elements by stereotype and access to stereotype attributes
 - Ideally a plugin facility to allow access to the model via an API
- The key points are good UML profile support and some way to get model information back out again
 - many tools are lacking in one or both of these



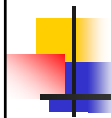
Tools – UML Profile Support

- Tool support for UML2 profiles varies widely
 - many tools don't provide easy customer authoring of profiles
 - applying custom profiles can be awkward
- The key questions to establish for your tool are
 - can the profile be worked on as easily as a UML model?
 - how complicated is it to release a profile for use?
 - once in use, how difficult is it to change a profile?
 - How difficult is it to specify custom scalable graphics?
 - most promise SVG support but it doesn't always work!
 - When using a custom profile, how difficult is it to apply stereotypes and specify custom attribute values?

v1.20080519

(C) Eoin Woods 2008

41



Tools – Getting Data Out

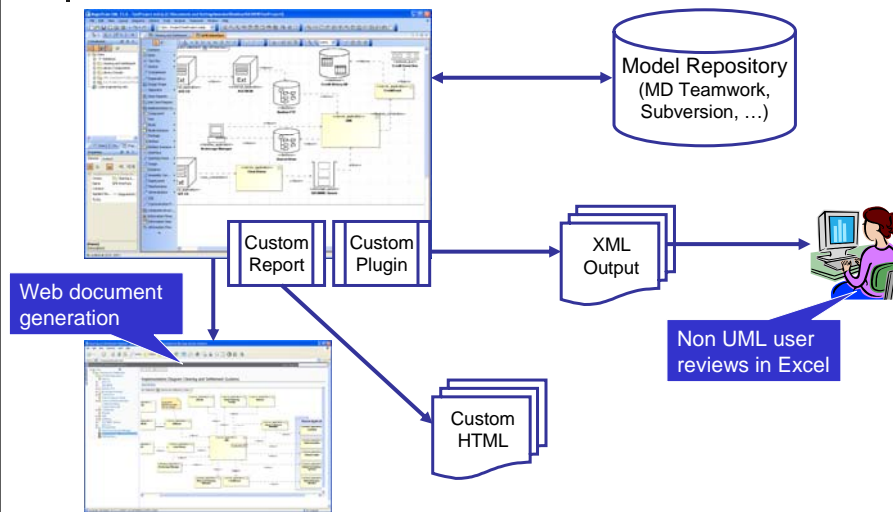
- Once information has been entered into the model, you need ways of getting it back out again
 - both reports but also ideally specific queries
- The mechanisms vary by tool but in generally are
 - a reporting engine with some degree of tailorability
 - a plugin API allowing code to "walk" the model
 - an OCL query tool
- It is worth creating a "toy" profile and experimenting with data access mechanisms before settling on a tool

v1.20080519

(C) Eoin Woods 2008

42

Using the Profile – Tool Environment



v1.20080519

(C) Eoin Woods 2008

43


Using the Profile - Phases

- Modelling
 - create UML models using the profile, minimise other elements
 - take the time to capture the custom attribute values
 - create views for different people (a model so no redundancy)
- Reporting
 - use the web published model for reference by the UML-savvy
 - create custom reports for lookup and cross-reference by others
- Validating
 - use the XML reports for non-UML users who can review in Excel
 - if you're brave you can write an import plugin for changes!

v1.20080519

(C) Eoin Woods 2008


44



Content

- Introducing Systems-of-Systems
- Modelling Systems-of-Systems
- Using and Extending UML for Systems-of-Systems Models
- A Profile for Systems-of-Systems Models
- Using a Custom Profile Effectively
- Summary and Conclusions

v1.20080519 (C) Eoin Woods 2008 45



Summary

- Systems-of-systems are a good subject for modelling
 - good payback on investment
 - often no alternative for impact analysis and comprehension
- Standard approaches don't work all that well
 - boxes and lines are never up to date, redundancy, no query
 - standard UML doesn't have the right primitives
 - a custom database is a lot of work and has no graphical form
 - special purpose notations aren't widely used or well supported
- A UML profile can give you the best of both worlds
 - standard tooling and semantics
 - rich picture or DSL style result
 - reasonably low cost of tooling and modelling

v1.20080519 (C) Eoin Woods 2008 46



Conclusions

- A custom UML profile has worked well in practice
- The approach does have preconditions
 - UML knowledge, enthusiasm
 - a decent UML tool
 - time and effort to create and use the profile for real
- The profile isn't hard to create
 - can start simple and only grow as needed
- Effective results are more likely with a good reporting environment to allow results to be shared and validated

v1.20080519

(C) Eoin Woods 2008

47

Eoin Woods
Barclays Global Investors
eoin.woods@barclaysglobal.com
www.eoinwoods.info