



e-Framework

for education and research

e-Framework Submissions

Phil Nicholls (pjn@psydev.com)

29th May 2007

JISC, Birmingham

JISC



Australian Government

Department of Education, Science and Training

About Me:



Phil Nicholls

- Independent technical consultant
- Libraries and e-learning background (software development)
 - “Services from 1996”
 - CORBA 1998 (GAIA project)
 - SOAP 2001-present (COINE, COLIS, TELCERT)
- UK editor of e-Framework
- Specific interest in interoperability and service testing

Email: pjn@psydev.com

- We spoke a little about the components of the e-Framework:
 - Service Genres
 - Describe the idea
 - Service Expressions
 - Describe the means
 - Service Usage Models
 - Describe the linkages

Now...

- What is your project going to do?
 - In a paragraph (or two)?
- Does your work 'fit' with the e-framework?
 - Services
- How to submit your work:
 - Logistics...

Describe Your Project:

Describe Your Project:



R2Q2: QTI Rendering and Response Processing Services

Project website: <http://www.r2q2.ecs.soton.ac.uk/>

“The R2Q2 project was a response to feeling in the general assessment community that there was little incentive to move to the new version of IMS QTI that had been released. The community felt that the implementations of IMS QTI v2 were incomplete and difficult to use. There was a lack of robust tools and services that conformed to the new specification, and this was a barrier to entry. A central function required by many assessment systems is that of rendering questions and responding to users answers. The R2Q2 project produced a core set of web services to provide this functionality.

R2Q2 provides a complete rendering and response engine to all of the IMS QTI v2 question types. In itself it is likely to be a component of interest to other developers working in the assessment area – such as those working on authoring tools, question banks or learning environments for example.”

Is there a “Fit” with the e-Framework?

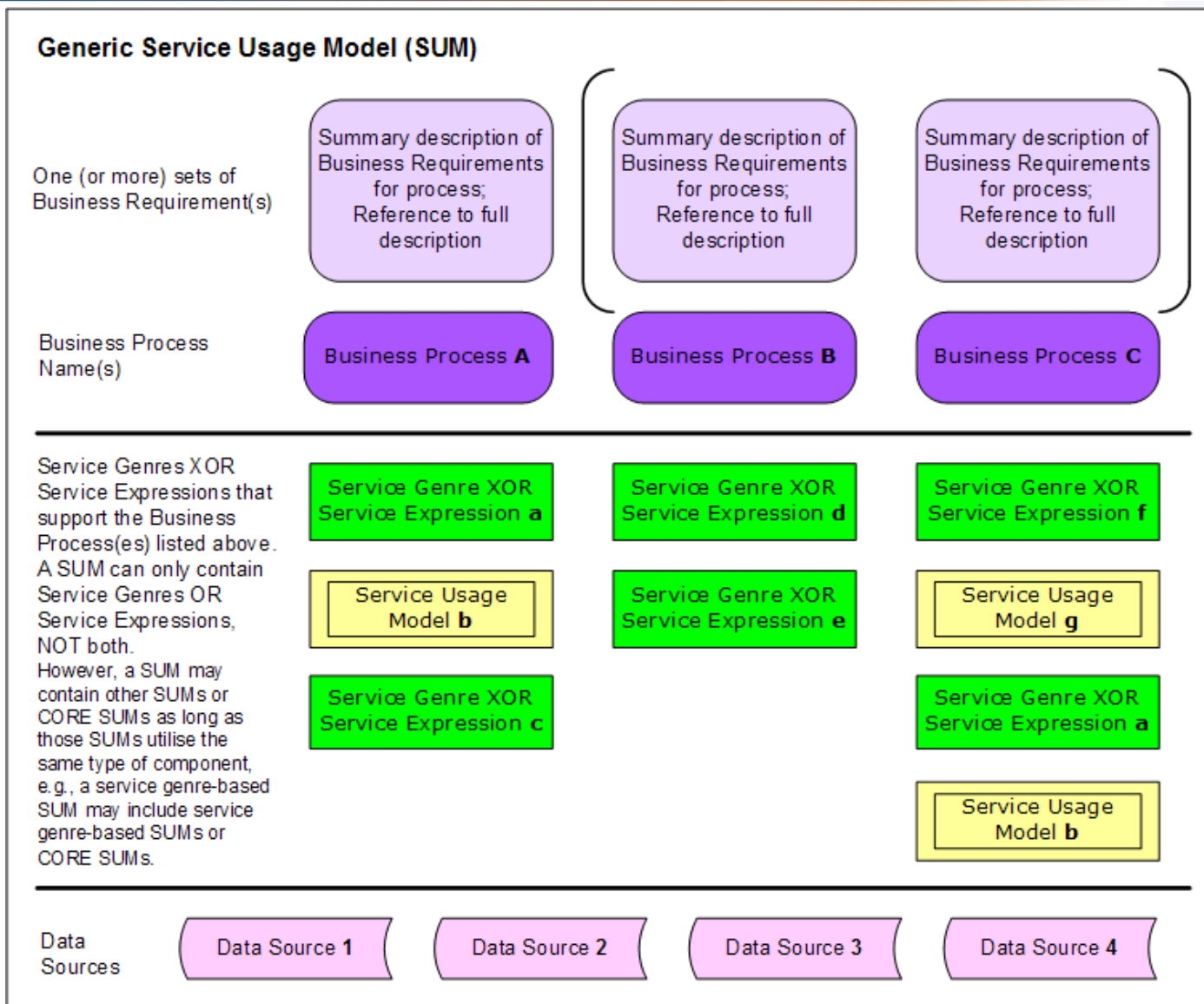
e-Framework

- Taking a service based approach
 - Building software (R2Q2)
 - Modelling processes
- Working directly with Services
 - Explaining to consumers - ‘How’
- Specification Development
 - What’s the ‘business’ problem?
 - Who is going to implement the specification?
 - Are you specifying behaviours?

- Services address specific ‘business’ problems.
- e-Framework supports the “service oriented approach”
 - Instead of one big application, there are lots of services.
 - The services are built to share data
 - New services can be “plugged in”
 - Services can be re-used by others
 - Services can be replaced
 - Services are dedicated

- Services:
 - Loose Coupling
 - Service Contracts
 - Autonomy
 - Abstraction
 - Reusable
 - Composable
 - Stateless
 - Discoverable

Linking Requirements to Services



Where does a project fit?

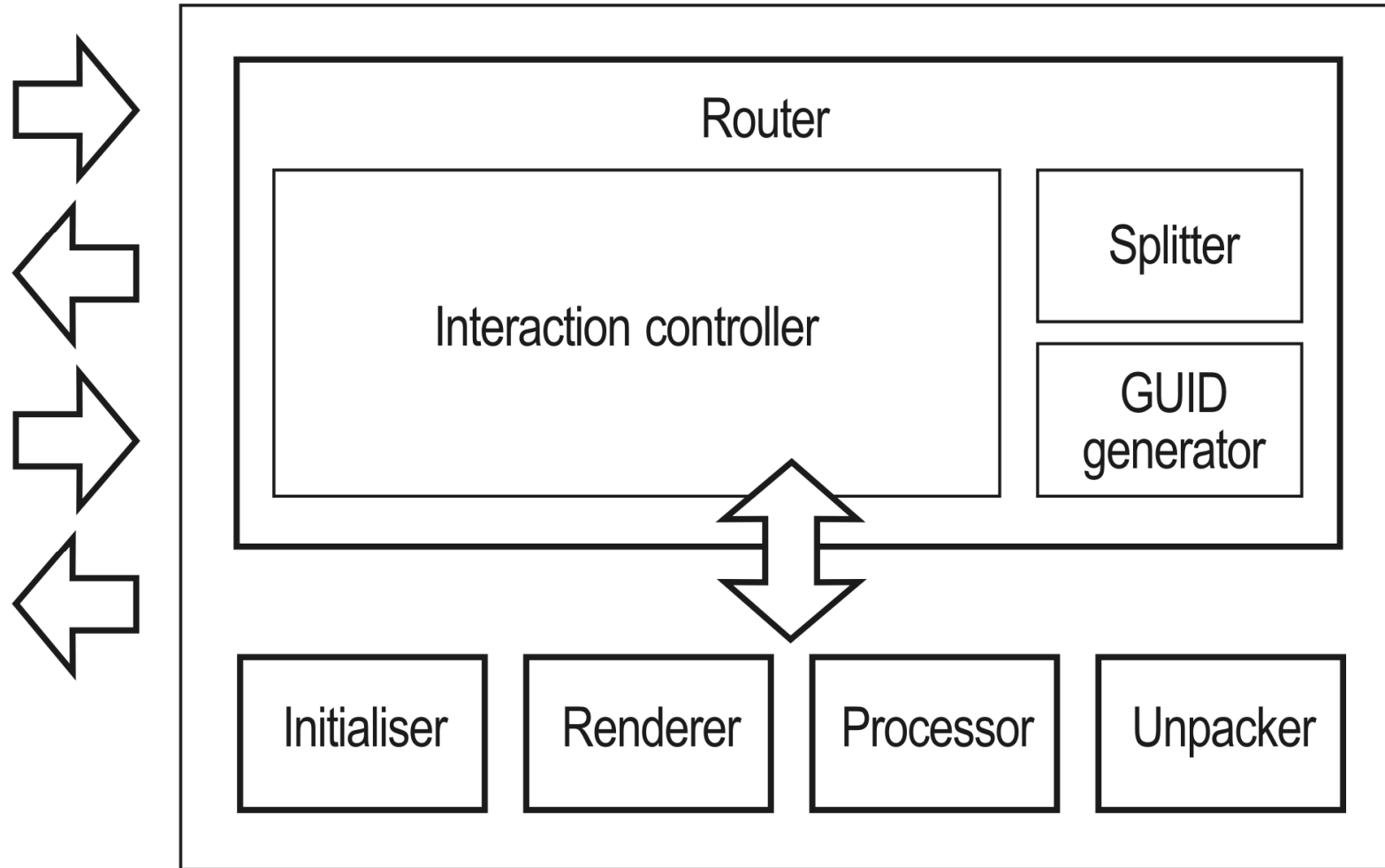


- What sort of project?
 - “How to” or “a means to” do something?
 - SUM or Implementation (Expression / Genre)
 - Developed a new service (Toolkit)
 - Implementation (Expression / Genre)
 - Adapting existing technology for new data
 - Implementation (Expression)
 - Linking services together (Demonstrator?)
 - SUM

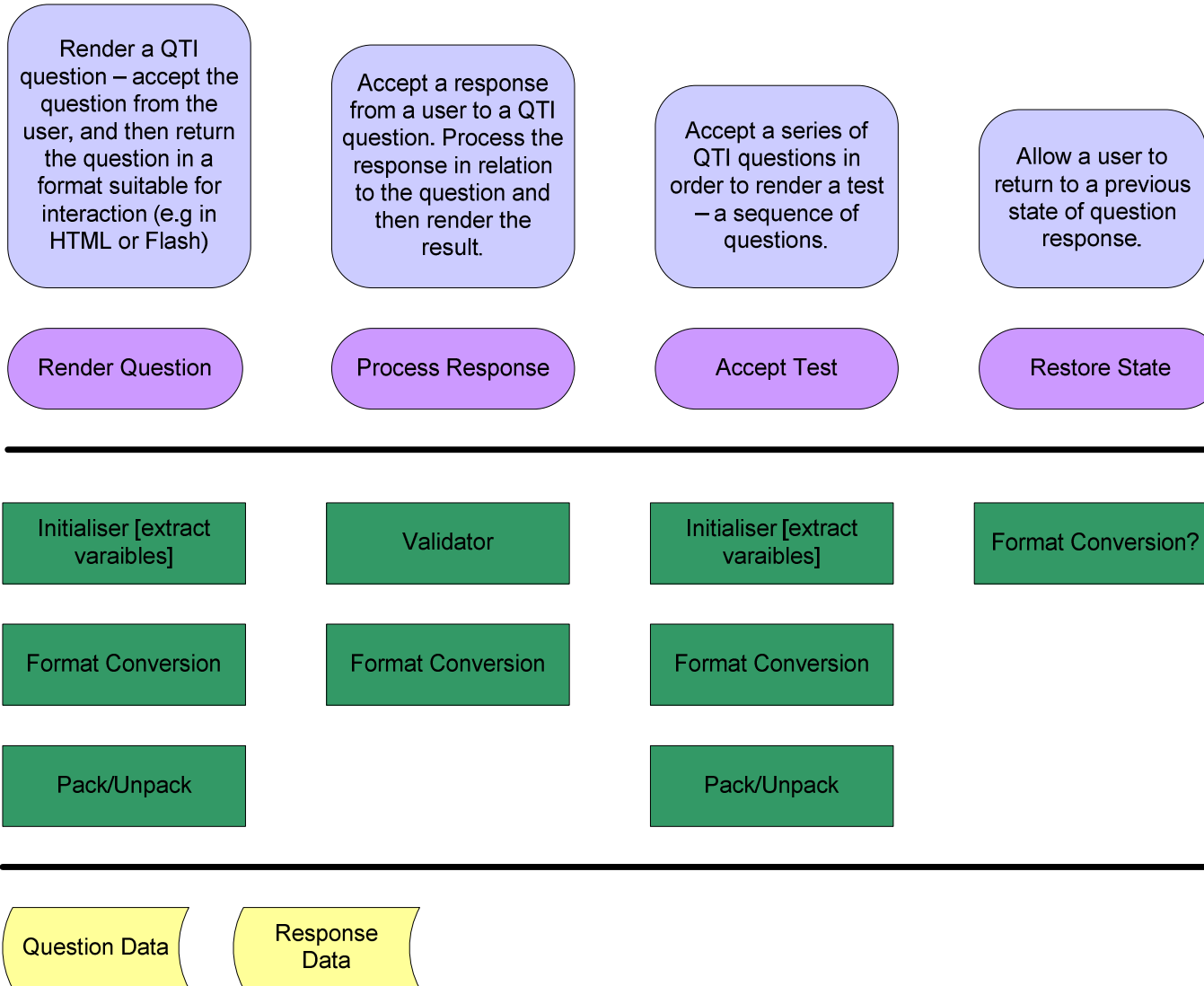
What's your fit?

- Does any of this relate to your project?

R2Q2 QTI v2 Rendering and response engine



R2Q2



Good Submissions:



- Need careful thought around:
 - Which functions come out as services?
 - How to define the interfaces to the services?
 - Granularity?

- A service is needed when more than one application needs to use a function
 - Think of business functions rather than software functions
 - Is the function useful to other applications?
 - If a function is only ever used in one specific application, it is not worth pulling out
 - Need to try to do cross application analysis to pull out common functions

- Factoring is easier with experience...
- Some questions that might help:
 - Is there already a standard or specification in the area?
 - Is there already a Genre or Expression in the e-Framework?
 - Does it make sense for the function to be autonomous? Is it useful on its own?

- Boundaries are explicit
 - Clear interface definition (e.g. WSDL)
- Services are autonomous
 - Independently deployed, versioned and managed
- Share contracts, not classes
 - Services separate data from behaviour
 - Changing contracts is expensive
- Compatibility is based upon policy
 - Cannot communicate everything in the interface – this is where the specs / standards come in

Defining a Service Interface



- Guided by service tenets
 - Well defined message that dictates the action to be taken (think business)
 - Strong types (passed as documents)
 - Use acknowledgements
 - Encapsulation is critical – avoid blurring public and private
 - Avoid RPC interfaces
 - Avoid loose interfaces (e.g. query() and execute())

- Builds on previous points
 - Need to think in terms of business objects and business processes
 - Coarse grained functions
 - Avoid lots of get/set methods
 - Do not call lots of methods to change data state
 - Cohesion according to goal / function
 - Data as documents, not lots of primitives

Factoring today is different from factoring tomorrow...

- It is possible to design “for the future”
- Think “re-usable component”
 - If something is perhaps not appropriate for being a service, it might be in the future
 - Putting service interfaces onto existing libraries isn’t necessarily a chore
- Build in a modular way

eFramework and Submissions



- What does the e-Framework do?
- How does content get into it?
- Why bother?

What the e-Framework documents:



- Collects information about services into one place
- Knowledge Base provides intelligence on services in different ways
 - Non technically – “what an <x> service does”
 - Technically – “how to write an <x> service”
 - Combinations – “if an <x> and a <y> are used together, one can do...”
 - Details actual services
 - Details models and workflows

Some Benefits



- Promotes work to wider international audiences
- Reduces re-invention and re-implementation
- Provides help by showcasing solutions to the problems encountered by institutions
- Fosters further sustainability

- **<http://www.e-framework.org>**
- **Step 1** Download and complete the relevant template. Include as much detailed technical information as possible. Include diagrams!
- **Step 2** On the e-framework website, use the upload facility on the submission.
- **Step 3** Provide any extra information requested on the online form.
- Get Published

- Check the website for guidance
 - Example content
 - White papers
 - Presentations
- Contact the editors:
 - Jon Mason (jon@intercog.net)
 - Phil Nicholls (pjn@psydev.com)

Any Questions...?