Unambiguous, Non-Binding Requirements for MDA

-David Hansz, Requirements Analyst
-David Fado, Software Architect

Why Discuss MDA and Requirements

- Experience! Efficient requirements management is a practical barrier in the use of models to deliver software. MDA implementation more efficient with a proper requirements “interface”.
- UML 2 offers exciting options to integrate requirement “specifications” with the rest of the model.
- MDA standards can apply to many types of software projects: MDA is about standards, not about development approaches.
- An illustration of how MDA standards are preconditions for advanced requirements strategies.

To MDA via UML

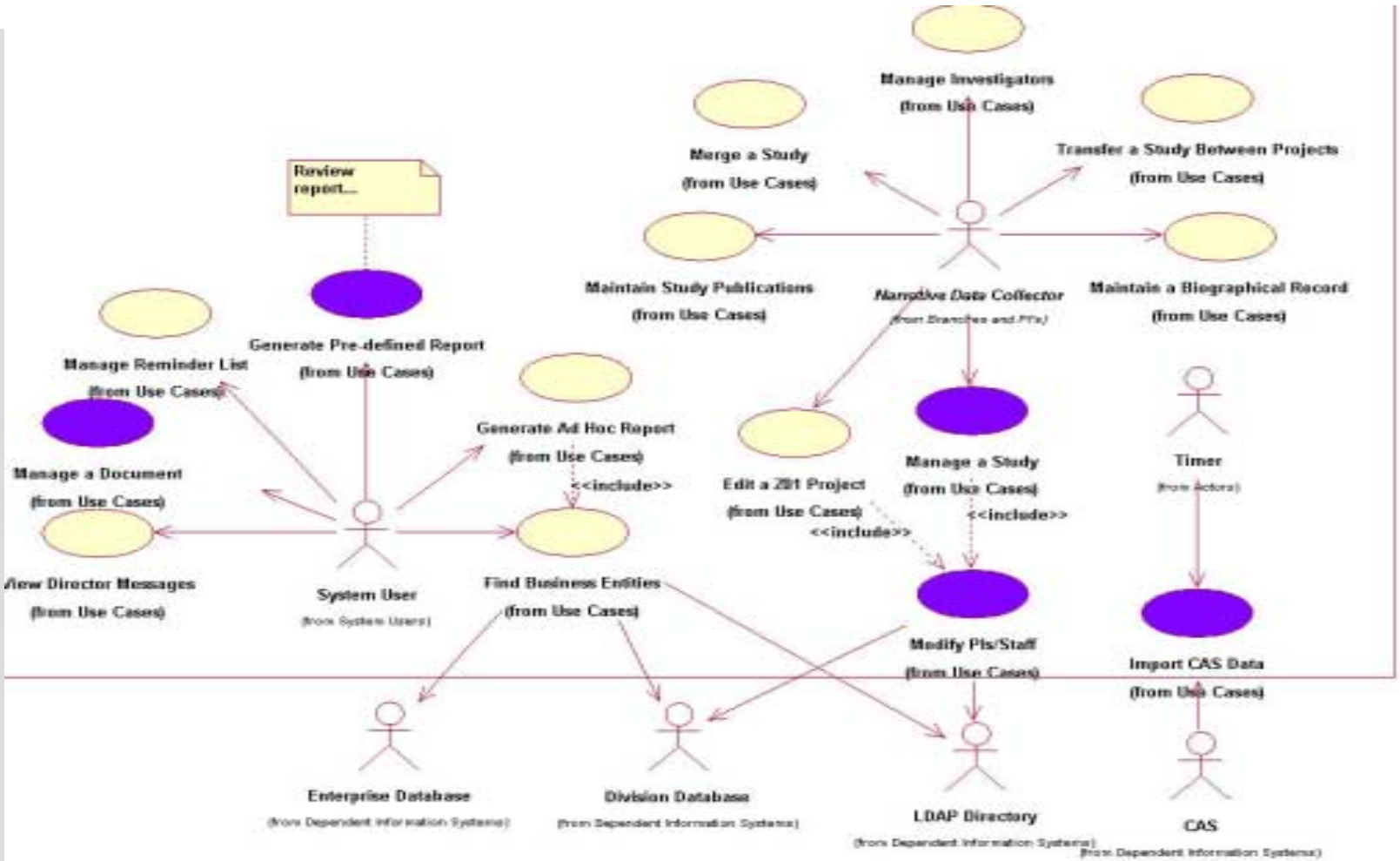
- The main goal: make models more effective (executable) for software delivery.
- MDA/UML 2 answers problems we faced making models effective using UML 1.x
  - Proliferation of Profiles
  - Domain models that cross projects
  - Communication between different UML tools
  - Integration of requirements artifacts with the model
  - Tendency for many artifacts to be ambiguous
- MDA a big tent, but those coming from high requirements churn projects are likely more covered in mud than others.

Artifact One: Use Cases

How Use Cases Can Help MDA

- Use cases are a part of UML and offer a foundation or starting point for using models.
- Use cases provide a high level organization focused on functions that benefit users.
- Facilitates communication about the application at the proper level of analysis for the client.
- A way to show the relationship between the users and the application.

Artifact One: Use Case Example



Artifact One: Use Cases

How MDA changes Use Cases

- More strict model creation. Disciplined in scope definition.
- Documentation will be kept within the model.
- Use Case specification artifact will now hold less importance as a requirements artifact.
- Use Case models will be composed of fewer, higher level, use cases focused on proposed system features
- Precisely defined requirements are a must. User interaction can no longer be explained in page after page of a specification.

UML 2 Opportunities

- Leveraging features in the new metamodel to allow the model to represent the requirements.
- Link to other model elements more efficiently to provide the “unambiguous” and “non-binding” features.

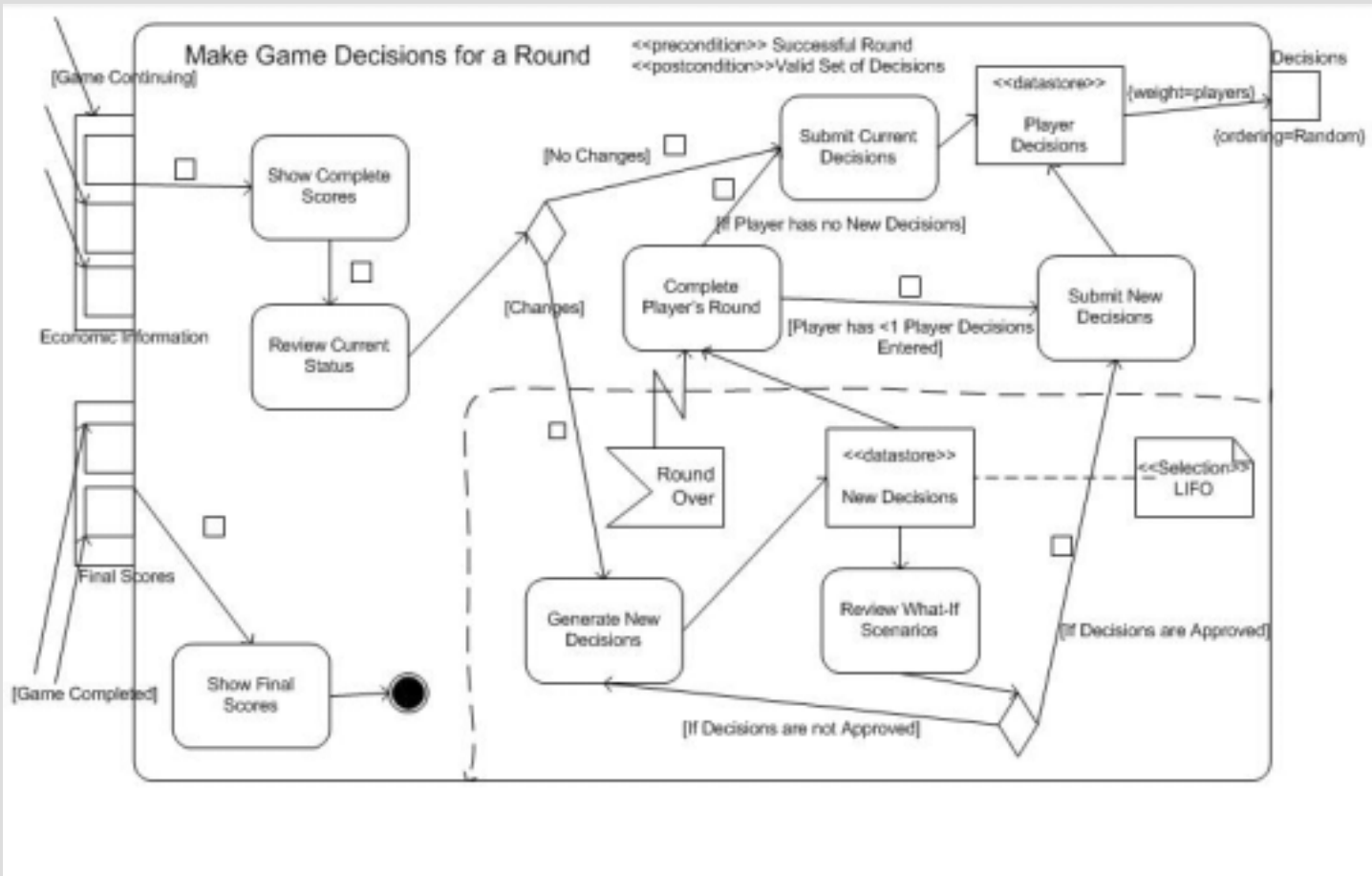
Artifact II: Activity Diagrams

Useful Features of Activity Diagrams

- Activity diagrams clarify application flow and constraints, independent of platform.
- Activity diagrams provide a clear way to elaborate an actor's role in an application.
- By clearly reflecting behavior, they can provide coherence to the functional testing more efficiently than the Use Case diagrams/specification.
- They can link to the CIM/domain model as well as the PSM, providing an excellent tool for change management.

Artifact II: Activity Diagrams

Example derived from www.ProBanker.com



Artifact II: Activity Diagrams

Changes for MDA

- Activity diagrams bear much of the weight for clarity.
- Carefully managed constraints using a standard constraint language.
- Rigorous change management and use of CIM objects.

Opportunities with UML 2

- Activity diagrams now reflect a token flow model better suited to clear specification and timing issues.
- Pins allow for the use of other classifiers in the system flow.
- And more….

Artifact 3: Documentation and Testing

How the Artifact will Help MDA

- Goal: integrate documentation and testing material with the rest of the model, avoiding duplication and maintenance problems.
- If models can be tested against accurate specifications, then problems are caught well upstream where they can be handled.
- Regardless of development efficiency, meeting changing requirements still a key variable in project success.

Artifact 3: Documentation and Testing

Substantial Modifications for MDA

- Project documentation should be expressed in the form of a model, as with the rest of the project artifacts.
- Profiles for testing will help in the management of these models.
- Making models executable against a modeled test case will require rigorous specifications.
- This changes the working pattern of many requirements documentation and testing documentation experts, who often stay outside the modeling tools. They need to be brought in.

Artifact 3: Documentation and Testing

Opportunities with UML 2

- Improved management of profiles makes requirement/testing profiles a possibility.
- Increase in features on diagrams enhance the possibilities for automated testing inside the model.

Current Strategies

- Focus on activity diagrams helping from a process point of view. This strategy provides the rigor needed in specifications.
- Use case analysis keeps focus on tasks the user needs to perform. Use cases also provide a useful way to discuss large units of functionality.
- Use case “specification document” increasingly integrated with model elements. This is no longer a document outside the model.
- Testing driven from activity diagrams.

Opportunities to Improve

- Testing and requirements documentation far from being integrated with the model.
- Object Constraint Language (OCL) not being adopted. Need a more practical method for expressing constraints. Trade-off between training in OCL and expressions in natural language that the user community can verify. Our clients would never understand OCL.
- A useful executable model verification not in scope of current projects.

Requirements Requirements for MDA Tools

- Provide integration with requirements and testing artifacts within the tool. Avoid the Requisite Pro/Rational Rose dichotomy with a separate requirements management tool outside the tool.
- Implement features to allow easy navigation from use case to related model elements.
- Implement the full features of the activity diagrams focused at the work flow level (less useful for PSM).