Fourth Revised SPEM 2.0 Submission

ad/06-08-01 and Errata ad/06-11-01 (RFP ad/04-11-04)
OMG Meeting, Washington, December 6th, 2006)

Peter Haumer (phaumer@us.ibm.com)

Outline

- Submission history
- Summary of Errata
- General overview and compliance points
- Working with Osellus and Conclusions

Submission team

- Adaptive Ltd.
  (pete.rivett@adaptive.com)
- Fujitsu Ltd. & Fujitsu Consulting
  (miyazaki.hir-02@jp.fujitsu.com & Serge.Deschamps@ca.fujitsu.com)
- Fundacion European Software Institute
  (Xabier.Larrucea@esi.es)
- International Business Machines
  (phaumer@us.ibm.com)
- Softeam
  (Philippe.Desfray@softeam.fr)

- Supporters: Alcatel, Armstrong Process Group, Aubry Conseil, BAE Systems, Boeing, Borland, Capgemini, EDS, HP, Kabira, Laboratoire d'Informatique Paris 6, Lockheed Martin, MEGA, MetaMatrix, Mitre, Number Six Software, Sierra Systems, SINTEF ICT, Telelogic, Unisys

Specification history

- RFP for SPEM 2.0 issued 04/11
- First specification drafts reviewed at Athens meeting 05/04
  - drafts and presentations from Fujitsu, ESI, IBM, Softeam
- Initial submission for Boston meeting 05/06
  - incorporated content from all submission drafts
  - feedback from Borland: RUP-centric, not fit for Agile
- First revised submission for Atlanta meeting 05/10
  - refactored packaging based on feedback from Borland and Adaptive
  - accepted request from Osellus and moved for 6m submission extension
- Second revised submission for St. Louis 06/04
  - incorporated minor feedback received
- Third revised submission for Boston 06/06
  - polishing and eleven case studies added
- Fourth revised submission for Anaheim 06/09
  - incorporated detailed feedback and requirements from Borland
  - updated model and profile for UML 2.1 and created XMI
- Errata for fourth revised submission for Washington 06/12
  - addressed minor reviewer issues

Outline

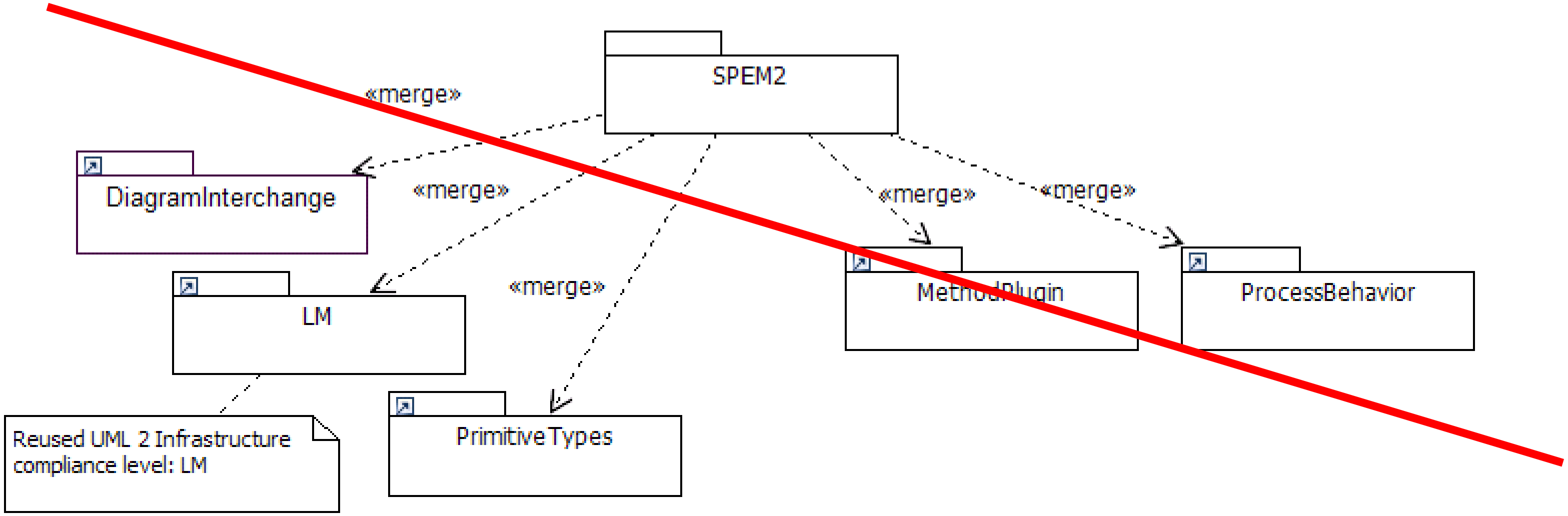
- RFP and submission history
- Summary of Errata
- General overview and compliance points
- Conclusions

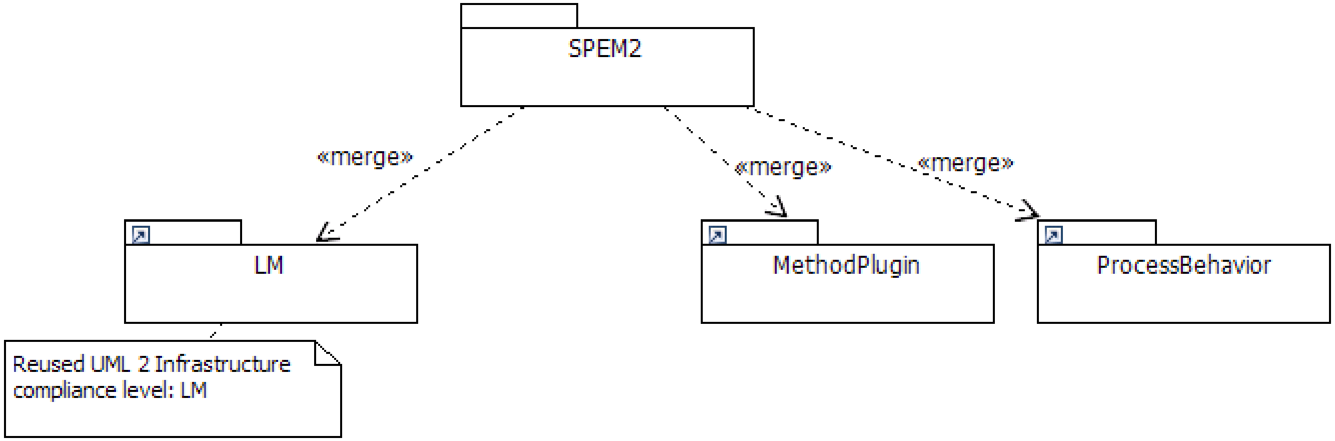
ad2006-11-01-Errata for ad2006-08-01: Summary

Continuous collaboration with OMG members: Addressed reviewer feedback and issues:

1. Removed Diagram Interchange as directly merged or imported package
2. Removed Primitive Types package and made Date and Double attributes Strings
3. Activity Extension stereotype now extends generalization, not dependency
4. Suppressed stereotype now extends dependency, not association
5. Added an Activity Use UML2 profile example to its section
6. Predecessor stereotype now extends dependency, not association
7. Metric now references Value Specification and no longer specializes guidance
8. Removed the class Process as a meta-model class
   (only use Activity with stereotypes now)
9. External Reference: made URI an association to Element

1. Removed Diagram Interchange as directly merged or imported package

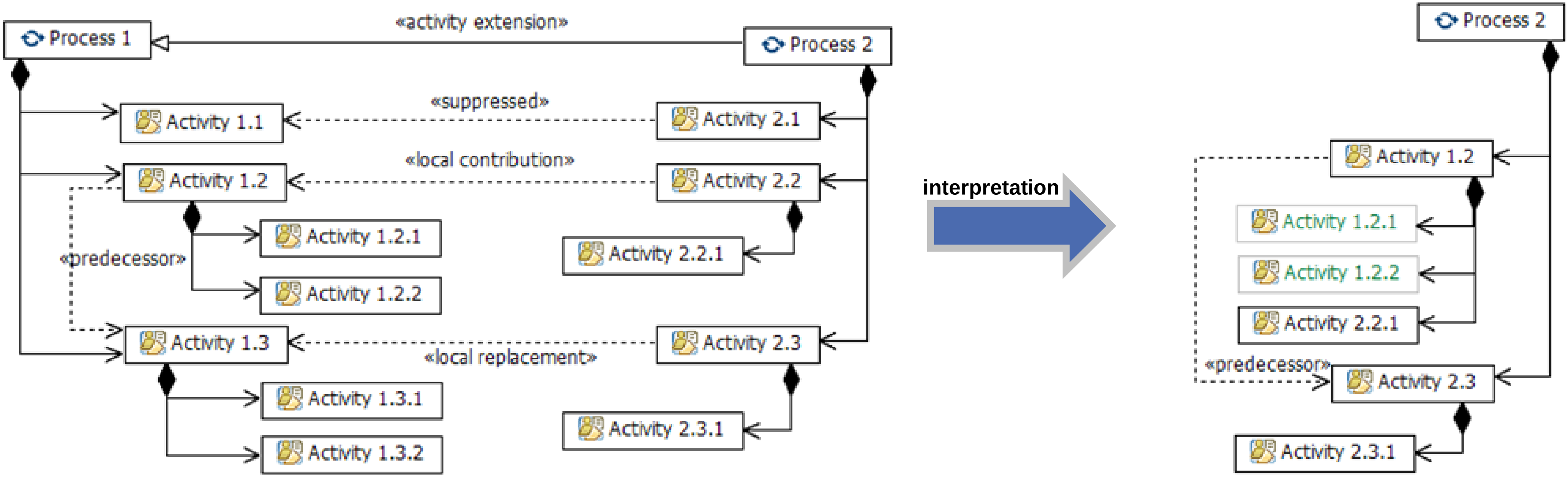




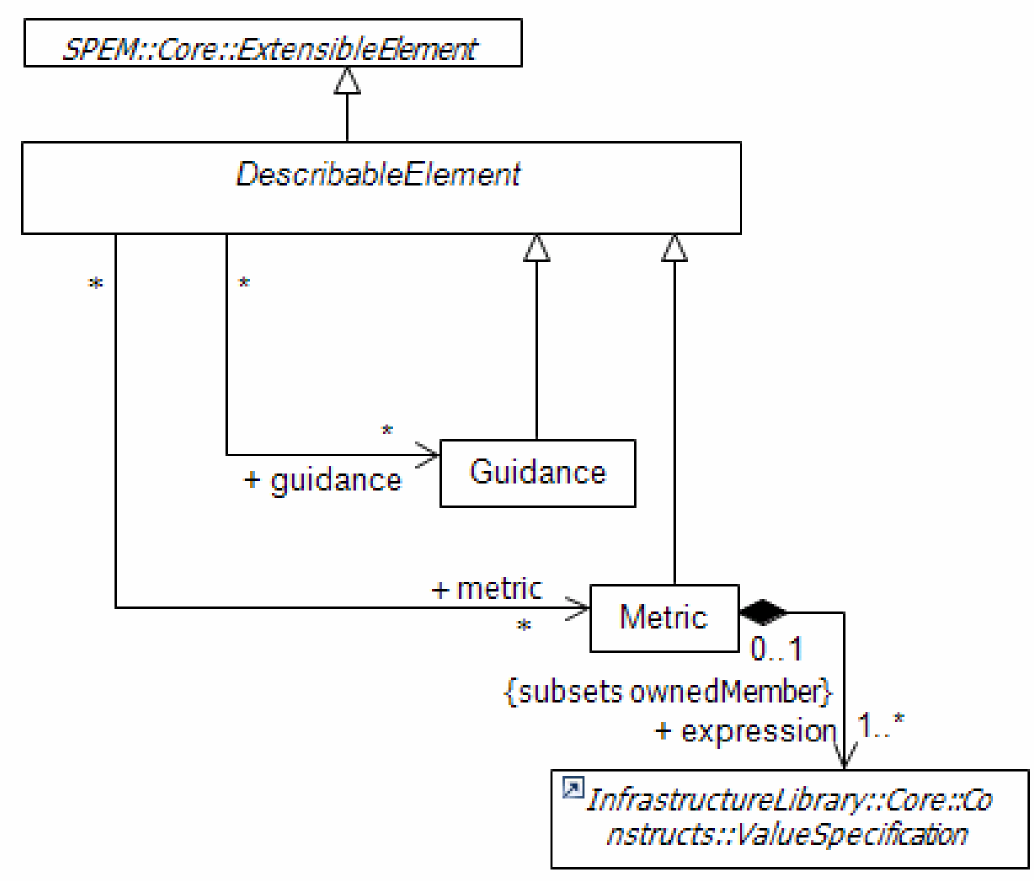
3. Activity Extension stereotype now extends generalization, not dependency
4. Suppressed stereotype now extends dependency, not association

Stereotype	*Meta-/Superclass*	*Keyword*	*Properties*	*Abstract*	*Icon*
UsedActivityExtension	Generalization	activity extension	n/a	no	n/a
UsedActivityLocalContribution	Dependency	local contribution	n/a	no	n/a
UsedActivityLocalReplacement	Dependency	local replacement	n/a	no	n/a
Suppressed	Dependency	suppressed	n/a	no	n/a

5. Added an Activity Use UML2 profile example to its section

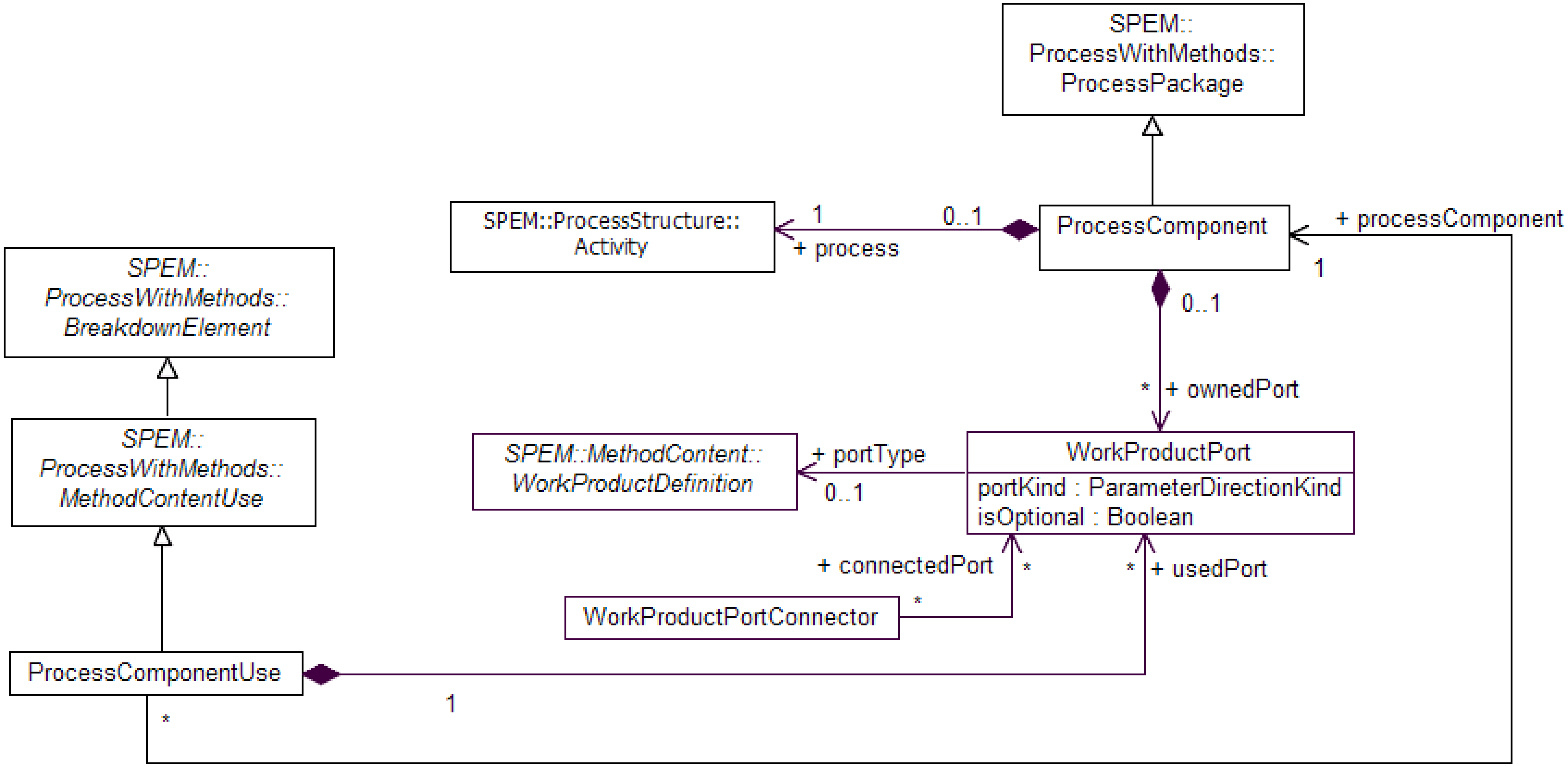


7. Metric now references Value Specification and no longer specializes Guidance



8. Removed the class Process as a meta-model class

Substituted all uses of Process with its generalization Activity



Outline

- RFP and submission history
- Summary of Errata
- General overview and compliance points
- Conclusions

General approach

Standardize representation and manage libraries of reusable **Method Content**

Content on agile development

Content on managing iterative development

Guidance on serialized java beans

JUnit user guidance

Content on J2EE

Configuration mgmt guidelines

Develop and manage **Processes** for performing projects

Process for Custom Application Development with J2EE

Process for Embedded System Development

Process for SOA Governance

Process assets patterns

Standard or reference processes

Enactable project plan templates

Corporate guidelines on compliance

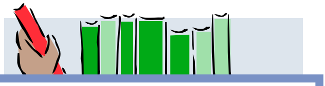
Configure a cohesive process framework customized for my project needs

Create project plan templates for **Enactment** of process in the context of my project

General approach and goals

Standardize representation and manage libraries of reusable **Method Content**

Content on agile



JUnit user guidance

Content on J2EE

Configuration mgmt guidelines

Goals:
- Management of reusable content.
- Basic concepts for expressing development methods and best practices.
- Free-form content and metrics of various types.
- Categorization for indexing.

Develop and manage **Processes** for performing projects

Process for Custom Application Development with J2

Process Embedded Syst Developm

Process SOA Governar

Process assets patterns

Goals:
- Represent processes close to best practices for development project
- Support reuse and tailoring of process
- Enable reuse from other process standards as much as possible.

Configure a cohesive process framework customized for my project needs

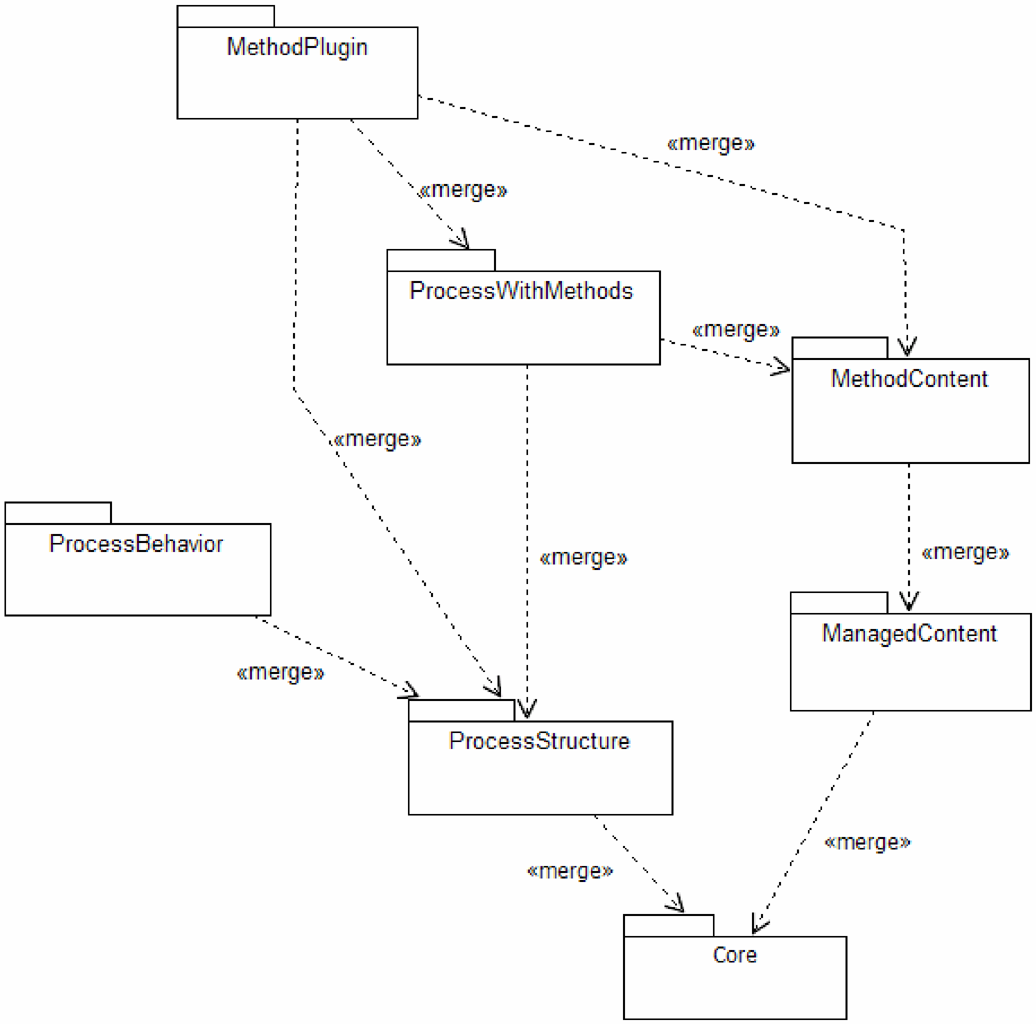
Goals:
- Support scaling, but also definition of views and sub-sets.

Create project plan templates for **Enactment** of process in the context of my pr

Goals:
- Support today's best practices of enacting development projects such as plans and work items..

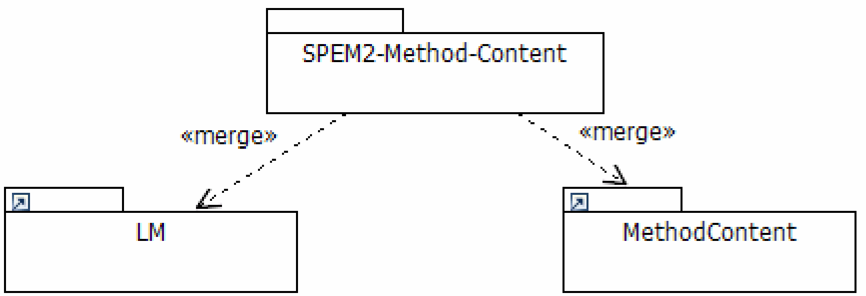
Meta-Model: Packaging supports implementing only parts of terminology

e.g., just creating an agile knowledge base of best practices (no process) or just model a processes without documentation



- Three Compliance Points
  (with supported XMI schemas)
  - Method Content
  - Process Behavior and Managed Content
  - SPEM 2 Complete
- Many other combinations possible for adopters and implementers
  - Process Structure and Behavior: Pure modeling; no documentation
  - Process Structure with Managed Content: to document and publish your processes.
  - Process Structure with Method Plugin: to manage collections of processes that extend each other.
  - Everything but Plug-ins: small scale of fully documented processes with method content

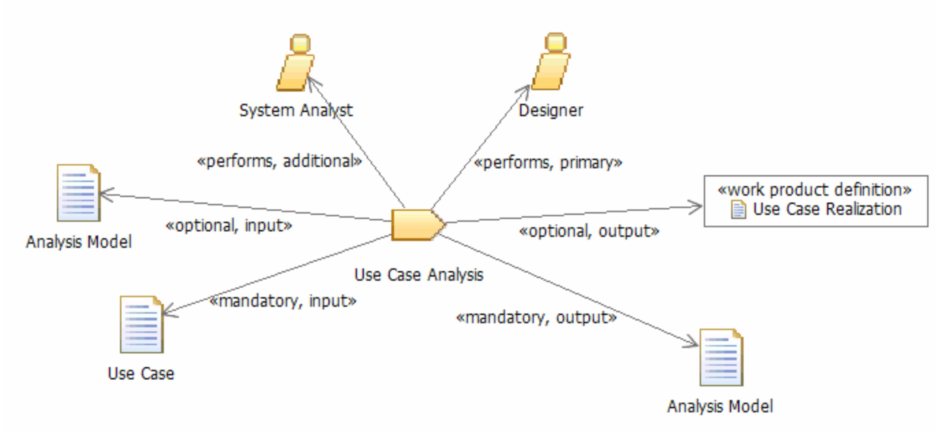
SPEM 2 Method Content



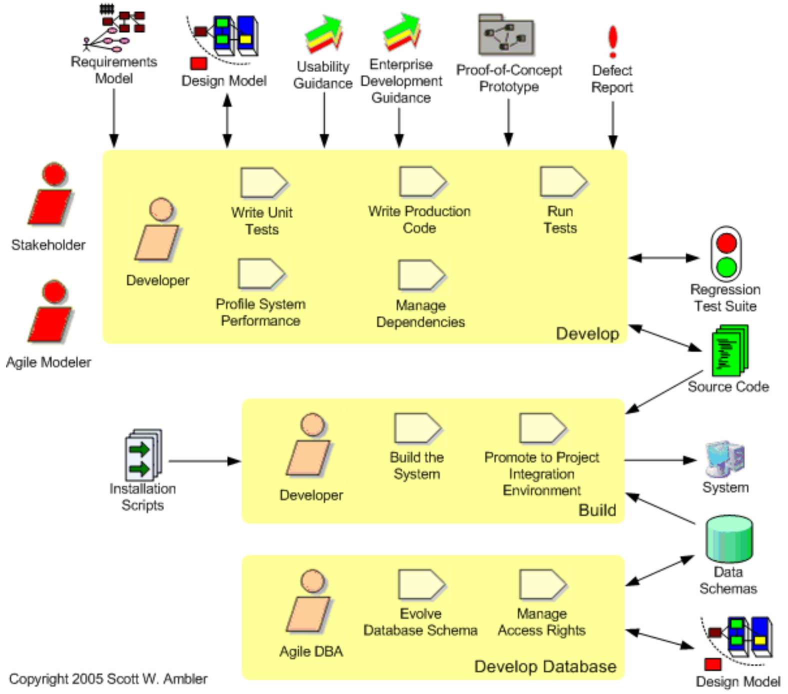
- Transitively merges: Managed Content and Core
- Focus: documentation of development methods, techniques, and best practices
- Audience:
  - Agile teams who do not want to see development processes expressed as flow charts
  - Book authors publishing development methods
  - Educators, Trainers and Mentors teaching people the principles of development
- Models provide low level of formality to express only basic relationships amongst roles, tasks, work products, and guidance
  - Which roles are responsible for which work products? Which tasks are performed by which roles with which input/outputs? What additional guidance gives me more background information?
- Can also be used to implement development knowledge bases or Wikis

SPEM 2 Method Content: Examples

Simple Method Content structure using SPEM 2 UML Profile

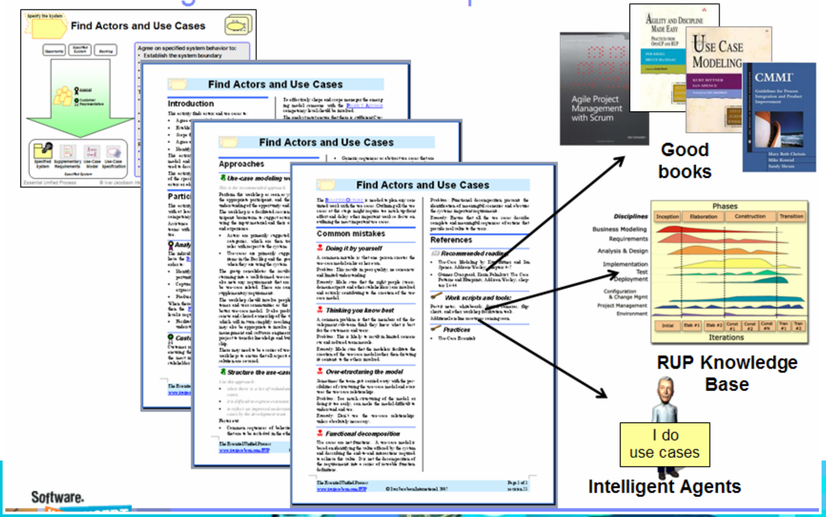


Scott Ambler's Agile Unified Process

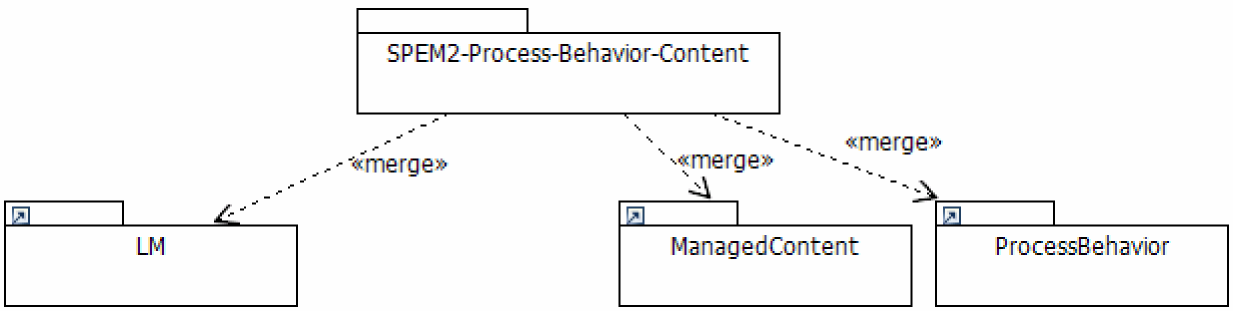


Cards and textual Guidance representation from Ivar Jacobson Intl's Essential Unified Process

Consult the guideline if more help is needed


Find Actors and Use Cases
Find Actors and Use Cases
Introduction
Find Actors and Use Cases
Approaches
Find Actors and Use Cases
Common mistakes
References
Agile Project Management with Scrum
Agility and Discipline Made Easy
Use Case Modeling
CMMI
Good books
Phases
Disciplines
Iterations
RUP Knowledge Base
I do use cases
Intelligent Agents
Software.

SPEM Process Behavior with Managed Content

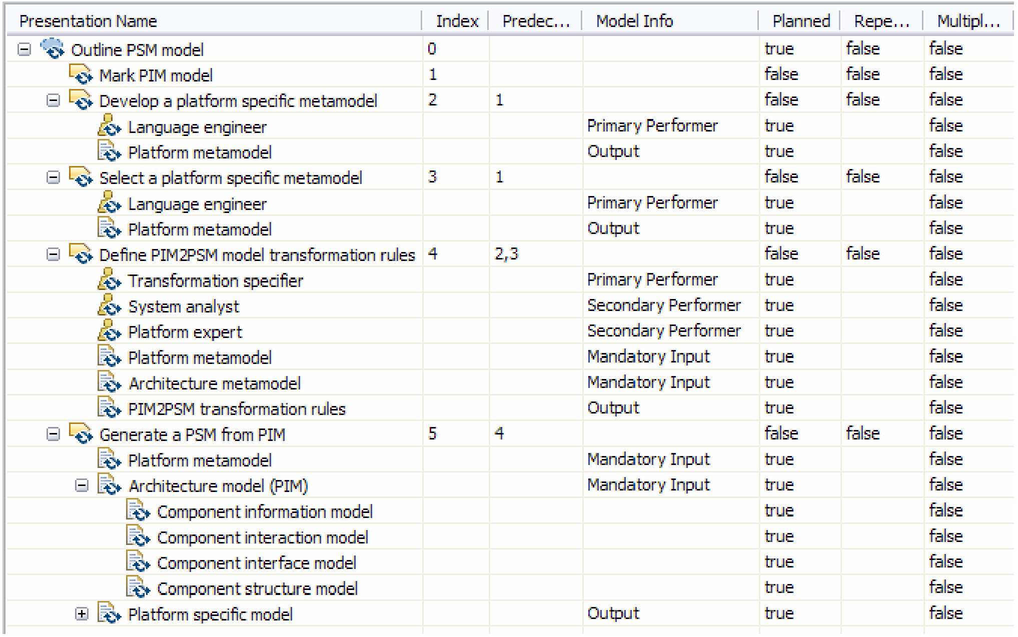


- Transitively merges: Process Structure and Core
- Focus:
  - modeling processes as workflows and breakdown structures
  - one process at the time
    (simple packaging and activity reuse, no scaling, componentization, or variability)
- Audience:
  - Classical SPEM 1.x audience
  - What to represent their process (in contrast to selling large libraries of process families)

SPEM Process with Behavior and Content: Examples

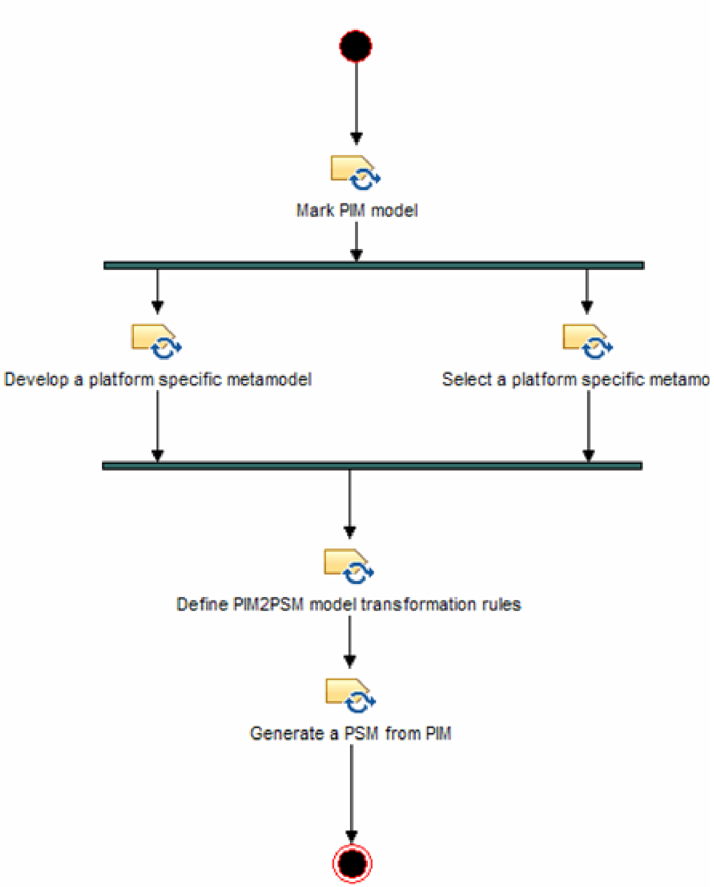
Open UP/MDD: Based on OMG MDA

SPEM 2.0 Specific Breakdown Structure

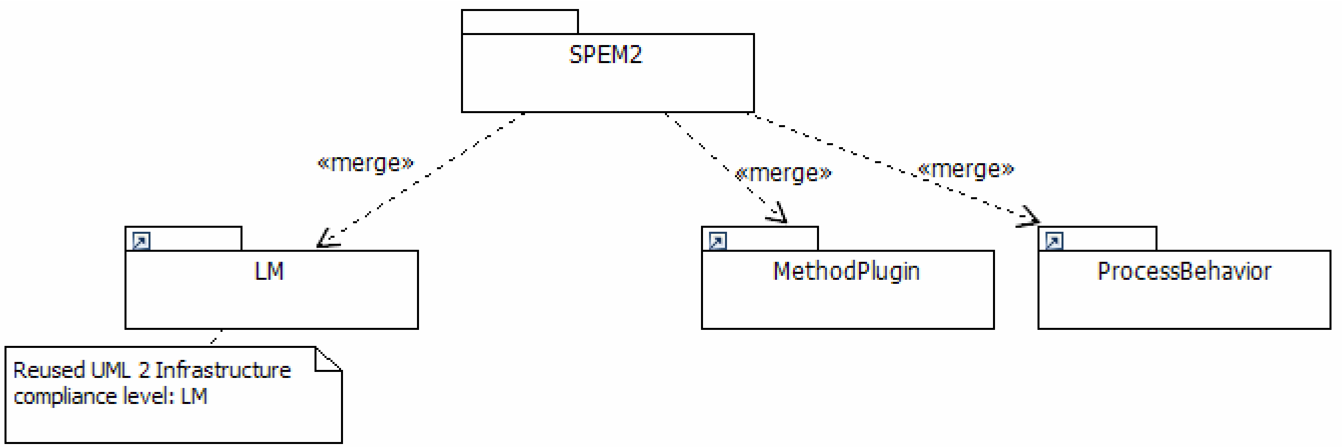


Presentation Name	Index	Predec...	Model Info	Planned	Repe...	Multipl...
Outline PSM model	0			true	false	false
Mark PIM model	1			false	false	false
Develop a platform specific metamodel	2	1		false	false	false
Language engineer			Primary Performer	true		false
Platform metamodel			Output	true		false
Select a platform specific metamodel	3	1		false	false	false
Language engineer			Primary Performer	true		false
Platform metamodel			Output	true		false
Define PIM2PSM model transformation rules	4	2,3		false	false	false
Transformation specifier			Primary Performer	true		false
System analyst			Secondary Performer	true		false
Platform expert			Secondary Performer	true		false
Platform metamodel			Mandatory Input	true		false
Architecture metamodel			Mandatory Input	true		false
PIM2PSM transformation rules			Output	true		false
Generate a PSM from PIM	5	4		false	false	false
Platform metamodel			Mandatory Input	true		false
Architecture model (PIM)			Mandatory Input	true		false
Component information model				true		false
Component interaction model				true		false
Component interface model				true		false
Component structure model				true		false
Platform specific model			Output	true		false

'Linked' UML 2.0 Activity Diagram


Mark PIM model
Develop a platform specific metamodel
Select a platform specific metamo
Define PIM2PSM model transformation rules
Generate a PSM from PIM

SPEM Complete

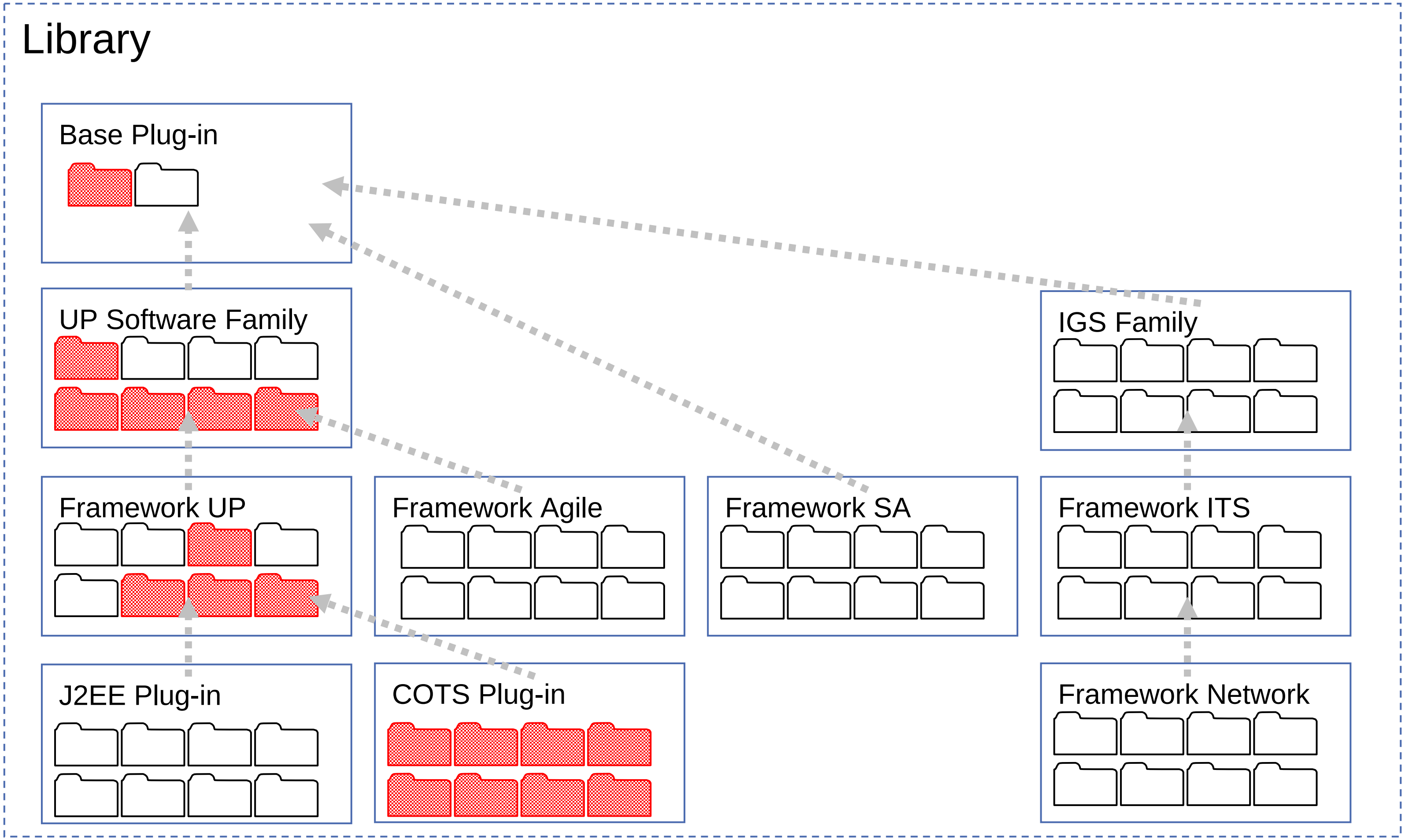


- Transitively merges: everything
- Focus:
  - support linking method content and process (base your processes on best practices)
  - providing additional management structures for scaling, reuse, componentization, and modeling variability
- Audience:
  - Method vendors and framework projects: sell and deploy libraries of and plug-ins for method content processes
  - Enterprise process management users: manage all the organization’s development processes; optimize on layering for hierarchical organizations and reuse

SPEM Complete: Configuration example

Concepts for dealing with large scale method libraries

Configurations specify the parts the process you need (i.e. define a visibility scopes within large libraries)



Outline

- RFP and submission history
- Summary of Errata
- General overview and compliance points
- Conclusions

SPEM2 Implementations

- Open source implementation available
  - Eclipse Process Framework, www.eclipse.org/epf/
  - Some key committers/contributors: 2-Pro Mentor, Ambysoft, APG, Bearing Point, Bedarra, CapGemini, Catalysts, Covansys, DSDM, ESI, IBM, Ivar Jacobson Int'l., Number Six, Object Mentor, Softeam, Telelogic, UBC, USC, Xansa
- Commercial implementation available
  - Rational Method Composer on Windows XP and Linux
  - Implementation and process models internationalized into 9 languages
- Implementations will be extended after adoption to support OMG XMI and UML Profile
  - Currently EMF XMI used based on earlier version of meta-model
- Adaptive will provide implementation
- Fujitsu will update their methodologies based on SPEM2
- Armstrong Process Group: create content and education material for SPEM2
- Open Group: Proof of concept for Architecture Development Method of TOGAF using SPEM2 concepts.
- DSDM proof of concept. Discussing adoption.

Many changes based on Osellus feedback incorporated in recent revisions of our submission

Incorporated a lot of great suggestions from the Osellus submission:

- Generic extensibility mechanism ("kind"-classes)
  - Allowed us to remove many controversial specialized meta-model classes
    - Work product specializations (Artifact, Deliverable, Outcome)
    - Activity specialization (Process, Capability Pattern, Delivery Process, Planning Template)
- Modeled many key relationships as extensible classifiers
  (e.g. to support user-defined relationships such as RACI)
- Added Qualification class for Role to provide bridge to enactment
- Revised conformance approach and defined three compliance points only
- Made management of textual representations optional

Remaining differences to Osellus submission

- Handling of workflow
  - Joint team: provide link to any other workflow formalism (recommend UML, provide BPMN example)
  - Osellus: do not provide solution, but would like to provide their own; plus link
- Managed content (managing text)
  - Joint: textual descriptions are key for interchange; textual process descriptions are relevant as the models; provide option for implementer to remove it (package merge)
  - Osellus: do not want to see it included at all
- Activity defining compositions versus components
  - Joint: follow UML approach for activities creating a namespace for sub-activities and related elements (similar to PINs and object nodes in UML owned by Activity)
  - Osellus: want to manage local elements in packages creating two breakdowns: activity breakdown as well as package breakdown
- Activity reuse versus extends
  - Joint: provide three extends relationships with specific semantics that handle complex extension case with very simple models
  - Osellus: provide one relationship with implementation specific semantics (relies on queries) and more complex model structures
- Enactment
  - Agreement: should issue a new RFP that handles this complex subject area in coordination with BMI
  - Joint: do not provide enactment specific classes in SPEM2, but guidance on how to realize enactment as part of an implementation
  - Osellus: provide a partial solution as part of SPEM2

RFP Requirements Addressed – Summary 1

Complete list at the end of this presentation.

- Alignment with UML 2.1
  - UML Infrastructure and optionally Superstructure
- Reuse OMG Specifications
  - Can be ‘linked’ to Superstructure, BPMN, BPDM
- Method Independent
  - Demonstrated with case studies from 11 different methods
  - Improved in this submission by removing special classes from meta-model
- Activity Modeling
  - UML 2 or any other activity modeling approach can be linked
- Support Process Enactment / Project Management Enactment
  - Can link to any “Activity” or “Statemachine”-based enactment system
  - Supports instantiation into project plans

RFP Requirements Addressed – Summary 2

- Revise Process Components
  - Aligned with UML 2
- Metrics
  - Added as describable elements using Infrastructure Expressions
- Guidance for Non-Expert Users
  - Detailed examples, new notations (e.g. breakdown structures), and case studies
- Internationalization
  - Demonstrated with implementation
- SPEM 1.1 to 2.0 migration
  - Mapping of all SPEM 1.1 concepts documented, Rationale provided
- At least two examples
  - Eleven case studies provided

Conclusions

- Documents available
  - ad/06-11-01: Errata to the joint revised SPEM 2.0 submission ad/2006-08-01
  - ad/06-11-02: SPEM 2.0 Convenience specification document with change bars
  - ad/06-11-03: SPEM 2.0 Convenience specification document without change bars
  - ad/06-11-04: Updated SPEM 2.0 XMI schemata for three compliance points
  - ad/06-11-05: Updated SPEM 2.0 UML 2 Profiles
- Specification is mature and solid
  - Used the many extension cycles for detailed reviews and bug hunting
  - Addressed reviewer feedback and issues
  - Models very mature: used UML 2 model checkers and XMI generators
  - Most concepts have been proven with implementations
- Specification is ready for adoption

Motion

- Motion for the ADTF to recommend the adoption of the following specification as the response to the SPEM 2.0 RFP ad/04-11-04

- ad/06-08-01 (SPEM 2 joint revised submission)
- ad/06-11-01 (Errata to the joint revised SPEM 2.0 submission ad/2006-08-01)
- ad/06-11-04 (Joint revised submission - SPEM 2.0 CMOF XMI schema)
- ad/06-11-05 (Joint revised submission - SPEM 2.0 UML 2 Profile XMI)