Process Metamodels

- Results of applying abstraction to software development processes

- Highlight the high-level features of a process or family of processes

- Can be instantiated in order to produce concrete processes

- The two most well-known object-oriented process metamodels:
  - OPEN Consortium’s *OPEN Process Framework (OPF)*
  - OMG’s *Software and Systems Process Engineering Metamodel (SPEM 2.0)*
Software Process Engineering Metamodel (SPEM 1.0)

- Similar in essence to OPF yet much simpler

- Primarily based on Rational Corporation’s *Unified Software Process Metamodel (USPM)*, which was chiefly intended as a metamodel for the RUP process

- Mainly supports the modeling of UML-based processes similar to RUP

- Unlike OPF, SPEM 1.0
  - does not include a process component library.
  - does not offer a specific procedure for instantiating a software development process using the metamodel (only well-formedness rules are provided).
SPEM 1.0: Core Structure

- Regards the core structure of a software development process as consisting of:
  - process roles
  - work products
  - activities

- Regards a software development process as
  - a collaboration of active entities (process roles)
    - aimed at performing specific operations (activities)
      - performed on a set of tangible artefacts (work products)
      - continued until the artefacts are brought to a well-defined state, and declared as complete.
SPEM 1.0: Core Structure

Diagram showing relationships between Role, WorkProduct, Activity, input, output, Uses, Performs, Produces.
Software Development Methodologies – Lecture 16

SPEM 1.0: Detailed Structure

- **Work products:**
  - may be composed of other work products;
  - can be associated with a state machine.

- **Activities:**
  - can be partitioned into *disciplines* based on their common structural and functional themes;
  - may consist of atomic sub-activities called *steps*;
  - can have a *precondition* and a *goal* as constraints on its enactment;
  - may be associated with an *activity graph*, which shows the flow of steps in the activity.
SPEM 1.0: Lifecycle Definition

- SPEM incorporates definitions for
  - Iteration
  - Phase
  - Lifecycle

- Intended to constrain the order in which the activities are performed, and to define the lifecycle structure of the process

- Very similar to their corresponding definitions in RUP
SPEM 1.0: Strengths and Weaknesses

**Strengths**

- Flexibility and configurability due to the generality of the metamodel (albeit limited, because of dependence on RUP as a metamodel basis)
- Well-defined general framework
- Provision of well-formedness rules to be observed when instantiating processes
SPEM 1.0: Strengths and Weaknesses

**Weaknesses**

- Lack of a specific instantiation procedure
- Lack of a detailed specification document: the specification document adopted by the OMG is a very general description of the metamodel.
- Mainly targets the modeling of processes similar to RUP, hence limiting applicability and generality (even the terminology is that used in RUP).
- The developer is responsible for constructing the methodology, and well-formedness rules are not enough to prevent bad instantiations.
Weaknesses (Contd.)

- Lack of subtyping for important process components (let alone a component library), which makes the metamodel of very little practical use. Consequently:
  - Poor coverage of lifecycle activities
  - Lack of explicit support for umbrella activities
  - Modeling and artefact production issues not explicitly addressed
Software and Systems Process Engineering Metamodel (SPEM 2.0)

- Adopted by OMG in December 2006
- Addresses the weaknesses of SPEM 1.0
- Provides necessary concepts for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes
  - Provides standardized representation and managed libraries of reusable method content
  - Supports systematic development, management, and growth of development processes
  - Supports deployment of method content and process needed by defining configurations of processes and method content
  - Supports enactment of process for development projects
SPEM 2.0: Conceptual Usage Framework

- 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
  - Lessons learnt from previous project and iteration
  - Corporate guidelines on compliance
  - Process assets patterns
  - Standard or reference processes
  - Project plan templates

**Configure** a cohesive process framework customized for my project needs

**Enactment** of process in the context of my project

[OMG 2007]
SPEM 2.0: Separation of Method Content from Development Process (1)

[OMG 2007] Department of Computer Engineering

Sharif University of Technology
SPEM 2.0: Separation of Method Content from Development Process (2)

Method Framework

Method Content
- Work Product Definition
- Role Definition
- Task Definition
- Category

Process
- Task Use
- Role Use
- Work Product Use
- Activity
- Process

[OMG 2007]
SPERM 2.0: Separation of Method Content from Development Process (3)

Method Content Element “Task Definition” referenced in more than one Process.

Underlying technical concept to support reuse and smart customization: “Task Use”

Individual customization of a “Task Use” by selecting steps, providing additional documentation, etc.

[OMG 2007]
SPEM 2.0: Method Content - Elements

- Roles are responsible for work products
  - Each work product is the responsibility of a single role
- Process roles perform tasks
  - Each task is only performed by a single role
- Work products used as inputs to tasks and outputs from tasks
  - “Somebody does something that changes something”
SPEM 2.0: Method Content - Guidance

- Can be associated with any process model element to provide more detailed information about the element to the practitioner.
- Can standalone – does not have to be associated.
- Most often associated with activities and work products.
- SPEM comes with a set of built-in guidance types:
  - Checklist
  - Template
  - Example
  - Tool mentor
  - Guideline
SPEM 2.0: Process Components

- Allow the user to treat the actual definition of the work that produces the outputs as a "black box."
- Allow different styles or techniques of doing work to be replaced with others.

[OMG 2007]

Department of Computer Engineering

Sharif University of Technology
SPEM 2.0: Process Patterns

Revisiting design work based on same underlying pattern

Dynamic linking of patterns increases maintainability

Changes in patterns require zero updates
SPEM 2.0: Modeling Enactable Processes

Role, Tasks, and Work Products

Activity Diagram

Work Breakdown Structures

[OMG 2007]
References
