Method Engineering using OPFRO

Donald Firesmith
Chair, OPF Repository Organization
http://www.opfro.org

Sponsored by the U.S. Department of Defense
© 2006 by Carnegie Mellon University
Topics

• Basic Concepts and Terminology
• State of the Practice
• Current Challenges
• Description of the OPFRO
• Method Engineering using OPFRO
• Current Limitations
• Future Directions
• Conclusion
Basic Concepts & Terminology

Method & Method Component
Process & Process Improvement
Process Metamodel
Method Component Repository
OPEN, OPF, and OPFRO
Method Tailoring & Method Engineering
Basic Concepts & Terminology

Method (a.k.a., methodology)
- A standardized way of describing a process consisting of a cohesive and consistent collection of integrated method components
- A model of a set of similar processes

Method Component (a.k.a., method element or fragment)
- A cohesive part of a method

Method should contain all types of method components:
- *Work products* to be produced or modified
- *Work units* to be performed on work products
- *Producers* who perform work units on work products
- *Stages* during which work units are performed
- *Endeavors* staffed by producers and organized by stages
Types of OPF Method Components

Method Component (Process Component Class) <<abstract>>

- Endeavor
- Language
- Producer
- Stage
- Work Product
- Work Unit

Note that method components (process component classes) actually need to be clabjects having both class and object characteristics. Clabjects are used to implement the Powertype pattern and are needed to ensure that core framework “classes” are properly connected through concrete method components to process instances.

Most developers should probably just consider method components to be classes of process components related by inheritance (generalization). Clabjects and Powertype pattern are formalisms of primary interest to professional methodologists and process tool vendors.
Relationships between Core OPF Method Components

- Producer
  - produces
  - performs
  - manipulates
- Work Product
  - is documented or implemented using
- Language
- Work Unit
- Stage
- Endeavor
  - is organized and staffed by
  - provides timing to the performance of
  - is timboxed using
  - is temporally organized by
Basic Concepts & Terminology

Process

• How real people and tools actually perform real work to produce or update real work products during the stages of real endeavors
  For example, Mary Brown’s use of I-Logix’s Rhapsody to create her UML design for the Fire Detection subsystem of her company’s Home Control system.
• The enactment (“instance”) of a method

Process Improvement

• Work performed to improve the processes actually used on endeavors, typically by improving the associated method and its usage
Basic Concepts & Terminology

Process Metamodel
- A metamodel for modeling processes
- A model for modeling methods
  (defines modeling language)

Examples include:
- OPF Metamodelling (OPEN Consortium)
- AS4651-2004 (Australia)
- ISO 24744 (draft)
- SPEM (OMG)

Method Repository
- A repository for storing reusable:
  - Method components
  - Methods
Traditional OMG Process Metamodel

- **M2 Level**: Process Metamodel
  - **M1 Level**: Process Model (Method)
    - **M0 Level**: Process (as enacted)
      - Process metaclasses
      - Process classes
      - Process class objects

- is an instance of
OPF Process Metamodel

- **Process Component Class Framework**
  - **Framework Level**
  - Generalization and instantiation

- **Process Component Class Libraries**
  - **Class Library Level**
  - Selection and integration

- **Process Models (Methods)**
  - **Method Level**
  - Instantiation

- **Processes (as enacted)**
  - **Process Level**
  - Core process clabjects (method components)
    - Are powertypes of process clabjects (method components)
      - Are aggregations of process clabjects (method components)
        - Are instances of process clabject objects (method component instances)
Framework Layer

Class Library Layer

- OPF Repository
- OFP-compliant Organizational Repository

Process Metamodel (Framework & Others)

{consistent, abstract and concrete, core classes & subclasses}

Method Layer

- OPF-compliant Industry (RUP, XP), Organizational, or Endeavor-specific Method

Process Model (Method)

{selected, integrated, concrete, tailored classes & subclasses}

Process (Enacted Method)

{integrated}

Process Component (Instance)

Method Layer

- Opf-compliant Endeavor-specific Process

Method Component (Process Component Class)

Process Layer

- Endeavor
- Language
- Producer
- Stage
- Work Product
- Work Unit
Repository Users
Basic Concepts & Terminology

Open Process, Environments, and Notation (OPEN) Consortium

OPEN Process Framework (OPF)
- The process framework developed and maintained by the OPEN Consortium

OPFRO
- The organization that develops and maintains the OPF repository [http://www.opfro.org](http://www.opfro.org)
OPF Process Framework (OPF)
Basic Concepts & Terminology

Method Tailoring
• Modifying an existing method to make it better fit the needs of a single endeavor

[Situational] Method Engineering
• Creating endeavor-specific methods by reusing (e.g., selecting, tailoring, and integrating) reusable method components
Method Engineering Tasks
Topics

• Basic Concepts and Terminology
• **State of the Practice**
  • Current Challenges
  • Description of the OPFRO
  • Method Engineering using OPFRO
• Current Limitations
• Future Directions
• Conclusion
State of the Practice

Method Source and Documentation:
• Ad hoc
• Popular book
• Organizational standards and procedures
• Consultant training materials
• Internet articles
• Process Tool (e.g., RUP)

Support Availability:
• Lack of local trained method/ process engineer
• Short term consultant
• Web websites and articles

Method tailoring and engineering:
• Some initial tailoring (often inadequate)
• Method engineering is rare and tools are mostly academic proof-of-concept prototypes
State of the Practice

Results:

- Shelfware
- Methods that are inappropriate:
  - Incomplete
    (missing needed roles, teams, disciplines, tasks, techniques, and work products)
  - Too heavy for project
    (e.g., waterfall and excessively document driven)
  - Too light for system size, business criticality, safety, and security, etc.
    (e.g., too agile)
- Poor Quality Systems and Software Applications
- Inconsistent:
  - Usage
  - Outcomes
Topics

• Basic Concepts and Terminology
• State of the Practice
• **Current Challenges**
  • Description of the OPFRO
  • Method Engineering using OPFRO
  • Current Limitations
  • Future Directions
• Conclusion
Current Challenges

Every Endeavor is Unique.

Process Engineers must contend with different:
  • Product Characteristics
  • Endeavor Characteristics
  • Organizational Characteristics
  • Method Characteristics

Methods Used do not Match Process Needs
Different product characteristics include different:

- **Product Type:**
  - System vs. Software Application

- **Product Number:**
  - Single System
  - Initial Production
  - Mass Production

- **Product Newness:**
  - “Green Field” vs. Enhancement

- **Product Variants:**
  - Single Product vs. Product Line
Product Characteristics

More different product characteristics include different:

- Product Size
  - Number of Requirements
  - Function Points
  - Subsystems

- Product Complexity

- Business Criticality

- Requirements Stability

- Technology Maturity

- Relevant Disciplines:
  - Such as Content Management & Digital Branding

- Quality Factor Criticality:
  - Such as Reliability, Performance, Safety, & Security
Endeavor Characteristics

Different endeavor characteristics include different:

• **Endeavor Type:**
  - Single Project
  - Program of Projects
  - Enterprise

• **Endeavor Contracting:**
  - Formally Specified and Binding
  - Informal Contract
  - No Contract
Endeavor Characteristics

More different endeavor characteristics include different:

- **Endeavor Scope:**
  - Business Reengineering Phases
  - System Development Phases
  - Operation / Usage Phases
  - Retirement Phase

- **Endeavor Schedule:**
  - Extremely Short to Generous (incredibly rare)

- **Endeavor Funding:**
  - Under Funded through Over Funded (also rare)
Organizational Characteristics

Different organizational characteristics include different:

- **Management Culture:**
  - Innovator through Laggard
  - Risk Taker through Risk Avoider

- **Developer Culture:**
  - Innovator (e.g., Agile)
  - Laggard (e.g., document-driven waterfall)
  - Web UI designers vs. web technical developers

- **Staff Localization:**
  - Everyone Co-Located
  - Locally Distributed
  - Geographically Distributed
Organizational Characteristics

More different organizational characteristics include different:

• Staff Organization:
  - Same Customer and Developer Organization
  - Separate Customer and Development Organizations
  - Separate Prime and Subcontractor
  - In-House and Outsourced

• Staff Expertise, Experience, and Skill Level:
  - High, Medium, and Low
  - Generalists vs. Specialists
  - Management vs. Technologists

• Methodological Maturity:
  - No Process
  - Shelf-ware Process
  - CMMI-Level
Method Characteristics

Different Method Scopes:
- Single Person
- Team
- Discipline
- Phase
- Development Cycle

Different Life Phases:
- Business Reengineering
- Product Development
- Operation
- Retirement
Method Characteristics

Different Method Sources:
- Popular Book
- Internet
- Consultant
- Organization Documentation
  (e.g., Standards and Procedures)
Method Characteristics

Different Method Constraints:
- International Standards (e.g., ISO and ANSI)
- National Standards (e.g., Military)
- De facto Industry Standards (e.g., RUP)
- Assessment Methods:
  - CMMI
  - SPICE or OOSPICE
- Method Types:
  - Heavy vs. Agile
- Cycle:
  - Waterfall vs. Spiral vs. Iterative/Incremental/Parallel
- Incremental and Iterative Build Length:
  - Short (days) vs. Medium (weeks) vs. Long (months)
Method Used vs. Process Needs

Methods used do not meet process needs:
• Low Process Maturity (e.g., CMMI)
• Method Size and Complexity
• Inappropriate Methods:
  - Engineering mismatch
  - Functional Decomposition vs. OO Decomposition
  - Waterfall vs. OO Lifecycle
  - Traditional Milestones vs. Incremental Milestones and Inch Pebbles
  - Traditional Milestones Reviews vs. Incremental Reviews
• Shelf-Ware
Topics

• Basic Concepts and Terminology
• State of the Practice
• Current Challenges
• **Description of the OPFRO**
  • Method Engineering using OPFRO
  • Current Limitations
  • Future Directions
• Conclusion
Description of the OPFRO

World’s largest collection of free, open source, and reusable method components

Over 1,100 method components

Currently in Website http://www.opfro.org and http://www.donald-firesmith.com (original mirror site)

Available for 5 years with over 1 million visits and roughly 20 million hits (very high Google ratings)

Written in standardized XHTML format

Moving to:
  • XML
  • Relational database
  • Eclipse epf toolset
  • Vendors
Description of the OPFRO

One primary webpage per method component

Organized according to OPF Metamodel

Easy, Standardized Navigation by:
- Navigation Tree Browser (left side of webpage)
- Website-Internal Search Engine
- Website-Internal Index
- Webpage Topics (top of webpage)
- Relatives (bottom of webpage)
- Internal Links (to referenced method components)
Activity

Definition

Activity

the highest-level work unit that models a cohesive collection of one or more tasks that are performed by one or more collaborating producers when producing a set of one or more related work products or providing one or more related services related to a single management or engineering discipline

Also known as a ‘discipline’

An activity is cohesive in the following senses:

- An activity models a single, functionally-cohesive discipline that maps a single exhibit
OPEN Process Framework (OPF)

- On small, short, and simple endeavors, an activity might be composed of only a single scaled down task performed by only a single person (perhaps playing multiple roles on multiple teams).

- To provide process engineers with maximum flexibility during the instantiation of the OPEN Process Framework (OPF), any activity can theoretically be composed out of any set of tasks. However, activities should be cohesive, and the relationships between activities and their component tasks are therefore typically much more constrained. For example, requirements engineering is usually composed of requirements tasks such as requirements elicitation, requirements analysis, and requirements specification.

For full navigation, enter website at: www.opfro.org

Last updated on 4 March 2006.

Website Statistics
Visits: 1,163,688
Hits: 18,796,263

Method Components: Endavors Languages Producers Stages Work Products Work Units

Work Units: Activities Tasks Techniques Work Flows

Contact Us

© 2006 by Carnegie Mellon University
Description of the OPFRO
**Endeavors**

Diagram: `Method Component` is temporally organized into `Endeavor` <<abstract>>. `Endeavor` is staffed by `Producer` and is temporally organized into stages. `Endeavor` is connected to `Stage`.

- `Enterprise` and `Program` are connected to `Project`.
Stages

Endeavor

Stage

Stage With Duration

Cycle

Phase

Build

Stage Without Duration

Milestone

Inch Pebble

Work Unit

Method Component

is temporally organized into

is typically punctuated by

is timeboxed using

provides timing to the performance of
Producers

Method Component

Producer <<abstract>>

Work Unit performs

Work Product produces

Indirect Producer

Direct Producer

Organization

Team

Role

Person

Tool

Enterprise

Program

Project

Endeavor

is played by

uses

is staffed by

© 2006 by Carnegie Mellon University

Method Engineering Using OPFRO - page 41
Work Units

Method Component

Work Unit "<abstract>"

Stage

Convention

Producer

Work Product

Activity (Discipline)

Task

Work Flow

Technique

performs

provides timing to the performance of

is performed according to

provides ways of performing

performs

manipulates

is timeboxed using

produces related

produces individual
Languages

Method Component

Language <<abstract>>

Constraint Language <<concrete>>
Implementation Language <<abstract>>
Modeling Language <<concrete>>
Natural Language <<concrete>>
Specification Language <<concrete>>

Database Language <<concrete>>
Interface Language <<concrete>>
Programming Language <<concrete>>
Protocol Language <<concrete>>

Scripting Language <<concrete>>
Topics

• Basic Concepts and Terminology
• State of the Practice
• Current Challenges
• Description of the OPFRO
• **Method Engineering using OPFRO**
• Current Limitations
• Future Directions
• Conclusion
Method Engineering with OPF

Major Tasks (highly iterative and incremental):
- Method Needs Assessment
- Method Construction
- Method Documentation
- Method Mandating
- Method Training
- Method Consulting
Method Needs Assessment

Major Method Needs Influenced By:
• System vs. Hardware vs. Software
• Product Size, Criticality, and Lifespan
• Relevant Disciplines (a.k.a., Activities)
• Resulting Method Size and Formality
• Methodology Principles

Get Help From:
• Process Engineers
• Process Consultants
• Methodologists
Method Construction Task

Method Component Selection and Tailoring:
- Activities (a.k.a., Disciplines)
- Work Products (and Languages)
- Tasks (and Steps)
- Producers (Teams, Roles, and Tools)
- Stages (Cycle, Phases, and Milestones)

Method Component Integration
- Integration
- Consistency Checking (and Fixing)
- Publication

Method Repository Extension
Topics

• Basic Concepts and Terminology
• State of the Practice
• Current Challenges
• Description of the OPFRO
• Method Engineering using OPFRO
• **Current Limitations**
• Future Directions
• Conclusion
Current Limitations

Manual process is too labor intensive and error prone.

Need XML and database versions

Need for automated help to:
• Find relevant and appropriate method components
• Determine appropriate method characteristics
• Tailor method components
• Integrate method components
• Ensure method consistency:
  - Dangling hyperlinks (pointers)
  - Orphaned method components (objects)
• Publish methods
• Maintain method components and methods
• Provide security (e.g., access control)
• Provide configuration management
Topics

- Basic Concepts and Terminology
- State of the Practice
- Current Challenges
- Description of the OPFRO
- Method Engineering using OPFRO
- Current Limitations

Future Directions

- Conclusion
Future Directions

We seek vendors and others who will collaborate with us to:
  • Incorporate the OPFR method components into their tool sets (e.g., Osellus and Eclipse epf)
  • Help us build tools for the OPFR (e.g., volunteers and Cesar Gonzales)

We seek active volunteers for:
  • Eclipse epf translation
  • Tool Development
  • Method Component Development & Maintenance
Future Directions – Planned Tools

- User Interface
  - Component Browser
  - Component Editor
  - Process Consultant
  - Method Builder
  - Method Browser
  - Method Editor
  - Method Simulator
  - Consistency Checker

- OPF Repository
  - OPF Metadata
  - OPF Metamodel
  - OPF Repository
  - Organizational Repository
  - Endeavor Repository
  - Reusable Method Components
  - Reusable Methods
  - Endeavor Method Components
  - Endeavor Method

- Security
  - CM
Topics

• Basic Concepts and Terminology
• State of the Practice
• Current Challenges
• Description of the OPFRO
• Method Engineering using OPFRO
• Current Limitations
• Future Directions

• Conclusion
Conclusion

Method engineering enables a method engineer to create an endeavor-specific method by selecting, tailoring, and integrating reusable method components stored in a method repository.

Method engineering enables method engineers to produce methods that are more endeavor-specific than tailoring generic tailorable methods.
Conclusion

Lego Effect:
• Standardization is in the repository of free, open source, reusable method components.
• Flexibility is in their selection and integration.

Method engineering is more practical if based on a:
• “Complete” repository of reusable method components
• Set of tools for selecting, tailoring, and integrating relevant method components
Contact Information

OPEN
http://www.open.org.au

OPEN Process Framework Repository Organization (OPFRO)
http://www.opfro.org
http://www.donald-firesmith.com (old mirror Web site)

Donald Firesmith (OPFRO Chair):
+1 (412) 268-6874 (SEI work phone)
dg@sei.cmu.edu (SEI email)
opfwebsite@hotmail.com (OPFRO email)