Software Factories in the Enterprise

May 23, 2005

Gunther Lenz
Program Manager,
Siemens Corporate Research

Lenz.Gunther@siemens.com

Overview

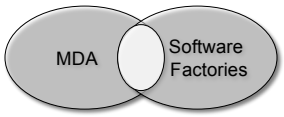
- **Developing Software with Models**
- **Software Factories**
- **Current Status of Software Factories**
- **Outlook**
- **Conclusion**
- **Questions**

Developing Software with Models

- **Motivation**
  - Craftsmanship
  - Many menial and error prone tasks
  - Increased system complexity
  - Little productivity gain over last decade
  - Poor Quality
  - Increasing demand for Software
  - …
- **Goals**
  - Become engineering discipline
  - Increase automation
  - Raise level of abstraction
  - Faster time to market
  - More reuse
  - Incorporate best practices (patterns etc.)
  - …

Popular Approaches for Developing Software with Models

- **Model Driven Architecture (MDA®)**
  - Owned by Object Management Group (OMG®)
  - Building Software using platform independent models.
- **Software Factories**
  - Driven by Microsoft® and other companies/groups
  - Building Product Lines with emphasis on automation.



Software Factories

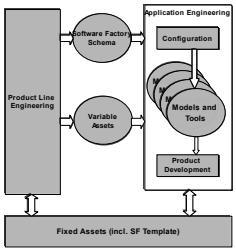
- **Why Software Factories?**
  - Mass customization, prevent one-off development
  - Economies of scope
  - Methodology (not tied to specific technology or implementation)
- **Organizational Aspects**
  - Focus on product line development
  - Integrates and supports development process
  - Reuse of existing assets
  - Feature modeling for commonalities and variabilities
  - Frameworks, patterns, models
- **Automation and Tooling**
  - Process guidance and help Integration
  - Domain Specific Languages (DSLs), templates, model transformations, recipes (Wizards), …

What are Software Factories?

- A set of integrated tool, process and content assets delivered to users in context to accelerate life cycle tasks for a specific type of software deliverable.
- At a minimum, a factory contains a schema and a template.
- Factories are designed to be composable:
  - Composing factories involves manipulating their schemas and templates.
- Involved technologies and methodologies are mostly well-known (e.g. DSLs, Patterns, Process, Orchestration), new is the holistic approach combining all of them.

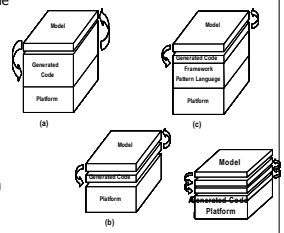
Software Factories Building Blocks

- **Product Line Development**
  - Define problem- and solution domain
  - Identify commonalities and variabilities
  - Reuse existing assets
  - Explicitly develop reusable assets
- **Software Factory Schema**
  - Defines models, viewpoints and artifacts
  - Describes mapping of artifacts
  - Recipe to build product family
- **Software Factory Template**
  - Set of installed assets required to use the factory in an IDE or other development environment
- **Domain Specific Languages (DSLs)**
  - Integrated into Visual Studio, Eclipse or other editors
  - Precisely model concepts of a specific domain
  - Enable to generate other models or code
  - Source artifacts, not just documentation



Models Today

- High level models
  - Large amount of generated code
  - Less efficient
- Low level models
  - Assisted pattern authoring application
  - Evaluation and assisted refactoring
  - Less powerful
- Raise the Level of Abstraction
  - Frameworks
  - Model fills in extension points



Software Development Today

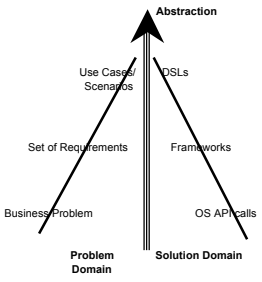
- Models
  - Mainly for documentation
  - Limited code generation capabilities
- Wizards
  - Guidance for developers
  - Automation and enforcing constraints
- Frameworks
  - Incorporate best practices and patterns
  - Raise level of abstraction
- Tools
  - Very little collaboration and integration

Current Status of Software Factories

- Description
  - Book: "Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools"; Jack Greenfield & Keith Short
- DSL tools
  - e.g. SQL, Windows Forms Designer, Class Designer, MetaCase, …
- Feature modeling tools
  - e.g. Krzysztof Czarnecki (University of Waterloo), pure-systems (pure::variants), …
- Guidance Automation Toolkit (GAT)

The Software Factories Tools We Use Today

- Enrich Projects with:
  - Guidance and Automation
  - DSLs
  - Feature Modeling
- Define Software Factories Schema
  - Architecture viewpoints
  - Relationship of viewpoints
  - Mapping requirements to architectural variabilities
  - Template for product family



Outlook

- Microsoft Tool Suite
  - Factory SDK v1 Dec 05, includes:
    - **DSL tools**
    - **Recipes**
    - **Factory life cycle tools**
    - **Factory process description**
    - **Developer guidance**
    - **Document templates**
- Other Tools
  - Software Factories schema
    - **Mapping of models**
    - **Definition of model relationship**
  - Model Exchange
    - **Standardization (W3C)**
  - Model Transformations
- Software Factories Supply Chains

Conclusion

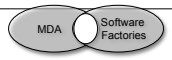
- Actions for Today:
  - Learn about the Software Factories
    - **Understand the methodology**
    - **Recognize applicability for product families**
    - **Do a Cost/Benefit analysis**
  - Implement automation tools:
    - **Bring DSL's into play (e.g. Workflow models)**
    - **Use the Guidance Automation Toolkit**
    - **Make use of DSL's**
  - Use Software Factory Schema to:
    - **Define architecture viewpoints (IEEE 1471) and their relationship**
    - **Identify product communalities and variabilities**
    - **Define the product family**
    - **Map architecture to feature variabilities**
- Actions for the Near Future:
  - Full blown Software Factories approach
    - **Not yet ready for mainstream (wait for SDK release)**
  - Evolution to Software Factories (instead of revolution)
    - **Enrich product development with new tools**
  - Microsoft will supply factories for certain domains
    - **e.g. E-commerce**
  - Other Tool vendors will provide support

Questions?

?

Lenz.Gunther@siemens.com

High-Level Comparison



	MDA	**Software Factories**
Standards	Multiple OMG standards	Not standardized
Platform Independence	PIM/PSM	Not specified
Model	Required	Not specified
Model Language	UML Profiles	Multiple DSLs
Development Process	Not specified	Integrates Process
Product Variations	Not specified	Product Line approach
Model Exchange	XMI	Plan: Standardization through W3C
Transformations	Not specified, QVT?	Not specified
Code Generation	PIM → PSM → Code	From DSL