Model-Driven Business Process Platforms

David S. Frankel
Lead Standards Architect – Model Driven Systems
SAP Labs
Economic Drivers

The Evolution of the Software Platform

Why This is Difficult

Meeting the Challenges: Metadata

Meeting the Challenges: Product Lines
Economic Driver: The Innovation Economy

Last-generation value proposition
- Applications embody well-validated, accepted business processes

Next-generation business reality
- Innovative business model and associated processes define an enterprise’s competitive advantage
  - Not enough to have a great product
  - Must also have a great business model
- Not attractive to simply follow a business process defined elsewhere
- Outsourcing all but core business processes
- Need flexibility in designing and executing innovative business processes
  - Core business processes
  - Value Chains / Value Networks for accessing mission-critical, non-core processes
Economic Drivers

The Evolution of the Software Platform

Why This is Difficult

Meeting the Challenges: Metadata

Meeting the Challenges: Product Lines
The Technical Platform Stack

Means A depends on B

Level of Abstraction

Middleware

Transaction Systems

DBMS

Network Systems

Operating System / VM

Machine
Composite App: Procure-to-Pay, Order-to-Cash, Manufacture-to-Inventory

David Burdett, SAP Labs

- - - > = Invoke
Model Compilers and the Abstraction Level

Application Model

Model Compilation Abstraction Gap

Middleware

Transaction Systems

DBMS

Network Systems

Operating System / VM

Machine

Context Preferred Cheking Inv:

\[ \text{Balance} \geq \text{minBalance} \]

© SAP AG and David S Frankel 2005, MDA / 9
Raising the Level of Abstraction

Composite Application Models

Model Compilation

Composite Applications

Direct Model Execution

Business Process Models

Abstraction Gap

Reusable, Executable Enterprise Application Services

Application Platform

Middleware

Transaction Systems

DBMS

Network Systems

Operating System / VM

Technical Software Platform

Machine

Business Process Platform

Composite Application Models

Abstraction Gap

Composite Applications

Direct Model Execution

Business Process Models

Reusable, Executable Enterprise Application Services

Application Platform

Middleware

Transaction Systems

DBMS

Network Systems

Operating System / VM

Technical Software Platform

Machine
Economic Drivers

The Evolution of the Software Platform

Why This is Difficult

Meeting the Challenges: Metadata

Meeting the Challenges: Product Lines
This Will Not be an Easy Transition

This jump in the platform abstraction level is more difficult than the last jump (middleware)

- Just as raising the abstraction level for development languages above 3GLs is more difficult than the last jump to 3GLs

Crawl, Walk, Run

- Provide business value at every step
Semantically thin specifications reach their limits

- How do you achieve semantic interoperability on top of syntactic interoperability?
  - Do collaborating parties have a common understanding the contract of a service?
  - You can’t rely on informal conversations among people
  - The parties might have different human languages as native tongues

- How do you find suitable services to compose?
  - Suitable functional behavior
  - Suitable quality of service?

- Same for reusable business processes

How do you build reusable components?

- Component-based development has proven hard in practice
- How do you anticipate requirements of composite applications?

Configuration/version/dependency management problems do not go away

- They can even get worse
Economic Drivers

The Evolution of the Software Platform

Why This is Difficult

Meeting the Challenges: Metadata

Meeting the Challenges: Product Lines
Semantically Rich Specifications

Need a metadata-rich environment to assist humans using the business process platform

- **Design by Contract™**
  - Service message types specified as precisely as possible
    - Invariants
  - Service operations functional contract specified as precisely as possible
    - Preconditions and postconditions (more numerous than invariants)
  - Using machine-readable, declarative constraint languages
  - We’ve know how to do this for decades
  - Also improves quality
  - Also need to learn to specify QoS requirements and capabilities as precisely as possible

- **Capture dependencies in metadata**
  - Don’t bury them in code

- **Formal grounding of languages**

- **Inferences identify candidates or flag potential problem combinations**
  - Let the human decide what to do
  - Record what the human decides
  - Show the next human what the others decided
  - Learn
Invariant rule expressed in UML’s Object Constraint Language (OCL). This invariant is independent of implementation technology.

context PreferredChecking inv:
--Cannot go below the minBalance
balance >= minBalance
Formal Abstract Model of a Business Service
With Pre-conditions and Post-Conditions

Signature

<<BusinessService>>
FundsXFer

XFerFromChecking(in fromAcct : CheckingAccount, in toAcct : SavingsAccount, in amount : Money)

context FundsXfer::XFerFromChecking (fromAcct : CheckingAccount, toAcct : SavingsAccount amount : Money) : void
pre:
    --There must be sufficient funds in the checking account to support the transfer
    fromAcct.balance >= amount
pre:
    --The checking account and the savings account must belong to the same customer
    fromAcct.customer = toAcct.customer
post:
    --The balance of the checking account is reduced from its original amount by the amount of the transfer
    fromAcct.balance = fromAcct.balance@pre - amount
post:
    --The balance of the savings account is increased from its original amount by the amount of the transfer
    toAcct.balance = toAcct.balance@pre + amount

Pre/post conditions—Independent of implementation technology
Bridging the Semantic Web and MDA Worlds

Means tool A outputs B, and B serves as input to tool C.

Generic UML Modeling Tool

UML Model Using UML Profile

Generic UML Model

MOF/XMI Based Tooling (e.g. Semantic Web Eclipse Plug-In)

XMI OWL Document

ER Tool

ER Model

Native Semantic Web Ontology Development Tool

Native OWL Document

UML-OWL Bridge

OWL-XMI Bridge

ER-OWL Bridge

OWL Document

ER Tool

ER Model
Metadata Across the Lifecycle
Model-Driven = Metadata-Driven (With Traceability)

Business Process Models
Service-Level Agreements
Composite Application Models
Warehouse Star Schemas
Operations Metadata

Service Definitions
Platform-Specific Component Descriptors
Operational RDB Schemas
Deployment Metadata

Common Metadata Management Infrastructure
Economic Drivers

The Evolution of the Software Platform

Why This is Difficult

Meeting the Challenges: Metadata

Meeting the Challenges: Product Lines
Product Line Practices

Individual systems produced via *product development*

The Sims “Water Line”

Reusable assets for the product line
Created via *core asset development*

Architecture

Components

Specialized Compiler(s)

Domain-Specific Language(s)
Applying Product Line Practices

Level of Abstraction

Business Process Platform

Asset Bases for Product Lines

Composite Applications

Asset Bases for Product Lines

Composite Applications
Desktop Application Platforms and BPPs

MS Office is a *desktop application platform*

- Has hundreds of components (has had them for 15 years)
  - Statless: e.g. a thesaurus component
  - Stateful: CRUD operations on office documents and calendars
- Powerful tools for rapid assembly of innovative desktop applications
  - Visual basic
  - Code wizards

A BPP has an *enterprise application platform*

- Will have hundreds (or more) components
  - Stateless, e.g. calculations
  - Stateful: CRUD operations on systems of record
- Model-driven tools
  - Application construction tools
  - Business Process Management tools

Synergy

- Integrating desktop and enterprise application platforms
  - Opens up another order of magnitude of possibilities for innovative composite applications
Summary

Business process platforms are coming

- Transition will be gradual, but powerful
- Model-driven tools are important for making the platforms usable

Configuration management has to be faced square-on

Metadata-rich environment, formal grounding, and product line practices needed to manage the complexity

Vigorous competition for a growing pie—if we do this right
Economic Drivers
The Evolution of the Software Platform
Why This is Difficult
Meeting the Challenges: Metadata
Meeting the Challenges: Product Lines