

SOA Enabled Workflow Modernization

Vitaly Khusidman

Abstract

Workflow Modernization is a case of Architecture Driven Modernization (ADM) and follows ADM Horseshoe Lifecycle. This paper explains how workflow modernization fits into the ADM paradigm and discusses different perspectives for the executable workflow and business process model as well as workflow system's inherent services orientation. The purpose of workflow modernization is to devise the target solution supporting new optimized business model and leveraging Business Process Management (BPM) and Service Oriented Architecture (SOA). It involves discovery of knowledge hidden in existing solution and capturing it in the as-is business model. The business model is then upgraded to to-be business model with new business requirements and is optimized based on the defined by business criteria. Then the to-be business model is used to devise the target workflow- and services-enabled solution. The workflow-enabled solution is viewed as a combination of workflow enactment service (WES) and invoked services. This approach enables definition of target business processes and business services and their optimization. It also enables mapping business services to business use cases and invoked service to system use cases in the forward engineering part of the ADM horseshoe lifecycle. Finally, this paper discusses major workflow modernization scenarios following ADM horseshoe lifecycle and leveraging SOA.

Background

Architecture Driven Modernization (ADM) is a discipline concerned with understanding existing software and other IT assets, preserving investments in existing systems (including proven business logic and expertise of current staff), ensuring they meet the enterprise's current requirements, and evolving those systems to meet future needs [1].

Traditionally ADM is associated with modernization of applications written in some programming language (e.g. older languages such as COBOL or C) and accessing data residing in the file system or database (e.g. IMS, ADABAS). However, a workflow enabled solution fits the ADM definition as well. Therefore, one can talk about Workflow Modernization as a case of ADM that follows ADM Horseshoe Lifecycle.

This paper explains how workflow modernization fits into the ADM paradigm and discusses its distinguishing features, such as the different levels of perspectives (i.e. executable workflow model vs. business process model) and inherent services orientation. The purpose of workflow modernization is to devise the target solution supporting new optimized business model and leveraging Business Process Management (BPM) and Service Oriented Architecture (SOA).

This is accomplished by discovering knowledge hidden in existing workflow enabled solutions and capturing the knowledge in a business model. Business models then undergo analysis, improvement, re-engineering, and change based on changed business goals, competitive landscape, internationalization, and a myriad of other reasons, for the purpose of optimization. The target workflow- and services-enabled solution is developed based on these business models by applying forward engineering methodologies and tools.

Architecture driven modernization overview

The ADM lifecycle involves reverse engineering existing solutions, adding new business goals, competitive threats, business requirements, and other artifacts, to create and optimize a "to-be" process model, and, finally, feeding the "to-be" model into the forward engineering process that follow the SDLC and workflow development methodologies.. The existing and target solutions (both workflow and services) belong to *IT domain*. In many cases the perspective for as-is and to-be models belongs to *business domain*. In these cases we are referring to as-is and to-be

business models. The described above approach is often called as ADM Horseshoe Lifecycle and is illustrated in the Figure 1 below:

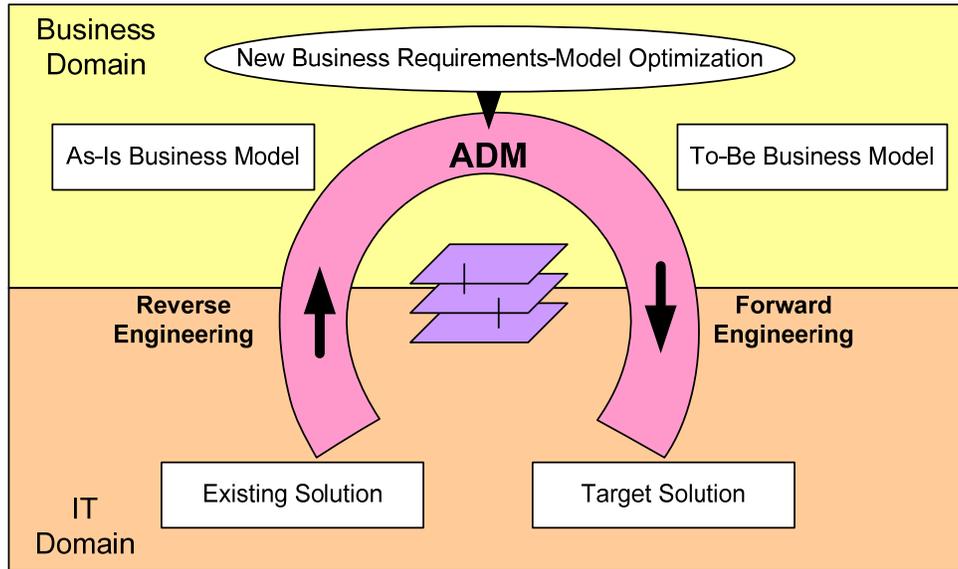


Figure 1. ADM Horseshoe Lifecycle

ADM horseshoe lifecycle consists of three parts: *reverse engineering*, *model upgrading and optimization* and *forward engineering*. A workflow-enabled solution is viewed as a combination of a workflow engine a.k.a. *workflow enactment service (WES)* and *invoked services*.

In the context of workflow modernization the reverse engineering part of the lifecycle includes mapping of workflow definition executed by WES and the invoked services to the as-is business model capturing the business description of the existing solution. This reverse engineering requires human analysis and assistance to change the perspectives from the IT domain to business domain.

The model upgrading and optimization part of the lifecycle includes analysis of the as-is business model, its enhancement with new business requirements and optimization based on criteria which addressing business objectives. This part of the lifecycle produces the to-be business model capturing the business definition of the target solution.

The forward engineering part of the lifecycle uses the to-be business model to devise the target workflow- and services-enabled solution following the SDLC and workflow development methodologies. This approach enables definition of potentially new boundary between business processes and services as well as definition and optimization of business services at the business model level. It also enables mapping the business services to business use cases and WES-invoked service to system use cases in the forward engineering part of the lifecycle.

Workflow reference model overview

Workflow Management Coalition (WfMC) has published the Workflow Reference Model [2] shown in Figure 2 below:

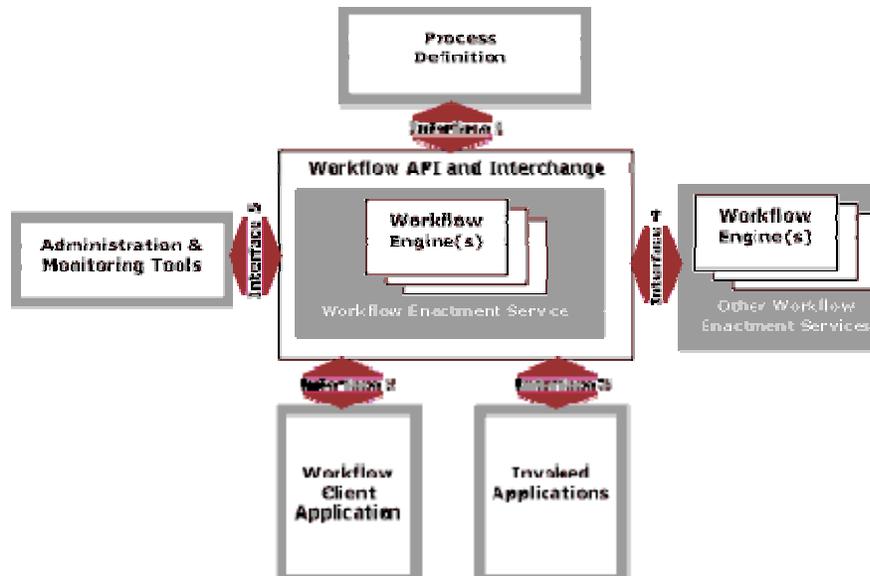


Figure 2. WfMC Workflow Reference Model

The WfMC Workflow Reference Model defines the following five interfaces associated with any workflow:

- **Interface 1** – between WES and a process definition tool. This interface is important in the context of Workflow Modernization since it facilitates exchange of workflow definition model (script) which is used in both existing and target workflow system.
- **Interface 2** – between WES and a Workflow Client Application. This interface is relevant to Workflow Modernization since it facilitates interaction with a workflow client application which itself is subject for knowledge discovery.
- **Interface 3** – between WES and Invoked Application. This interface is also relevant to Workflow Modernization since it facilitates interaction with an invoked application, which itself is subject for knowledge discovery.
- **Interface 4** - between WES and other WESes, This interface is not addressed in this paper.
- **Interface 5** – between WES and Administration & Monitoring Tools. This interface also is not addressed in this paper.

The WfMC Workflow Reference Model and interfaces and services it defines belong to IT domain.

Workflow modernization as a case of ADM

Workflow Modernization follows the ADM horseshoe lifecycle as illustrated in Figure 3 below.

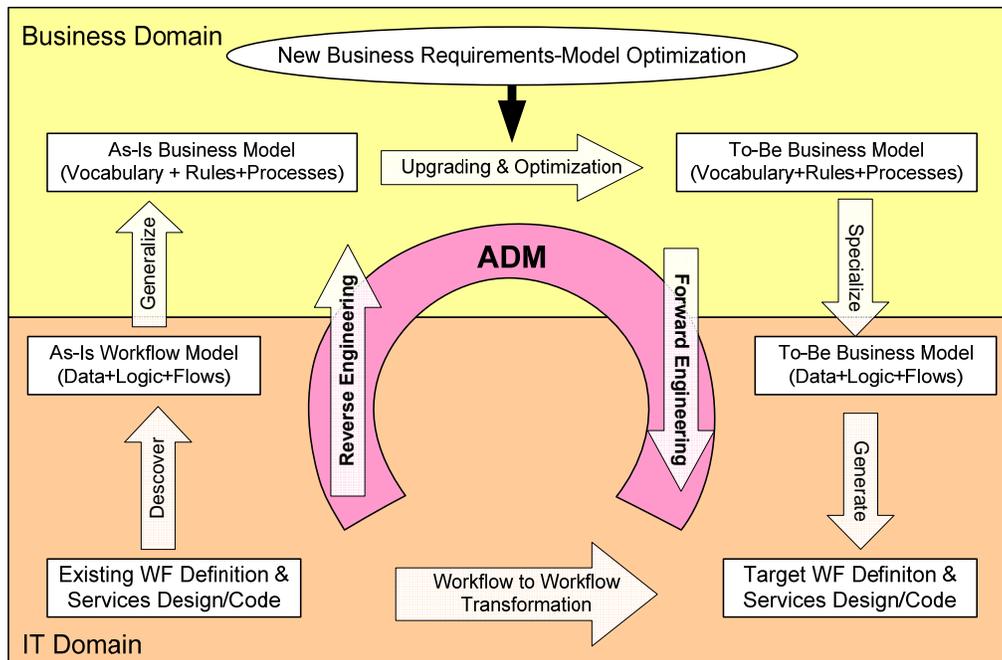


Figure 3. ADM Horseshoe lifecycle for workflow modernization.

The ADM horseshoe lifecycle for workflow modernization consists of three parts:

- **Reverse engineering** or Knowledge Mining and Abstraction (KMA) to discover knowledge hidden in the existing solution and generalize it to the business model.
- **Upgrading and optimization** of the business model to better satisfy changing business objectives.
- **Forward engineering** to develop the target solution.

The **reverse engineering** part of the ADM horseshoe lifecycle for workflow modernization includes two phases:

- **Discovery** of the knowledge hidden in the workflow definition and invoked services' code to build the as-is workflow model. Both sets of artifacts (i.e. existing workflow definition with invoked services and workflow model) belong to IT domain. This phase is highly automated – most transformations are done without human intervention.
- **Generalization** of the as-is workflow model to the as-is business model involves crossing IT/business domain boundary. The term *generalization* is used here in the sense of dropping implementation details and other information pertained to technology, and highlighting essential business-significant aspects of the solution. This phase of moving from one perspective to another inherently involves human assistance. However, all routine operations are highly automated to increase human productivity and reduce cost.

The **upgrading and optimization** part of the ADM horseshoe lifecycle for workflow modernization involves upgrading of the as-is business model with new business requirements and/or optimization the resultant model based on the criteria addressing business objectives. The ultimate result of the upgrading and optimization is the to-be business model.

The **forward engineering** part of the ADM horseshoe lifecycle for workflow modernization includes two phases:

- **Specialization** of the to-be business model into the to-be workflow model to cross the business/IT domain boundary and change perspective back to the IT perspective. The

term *specialization* is used here in the sense of enriching the model with details enabling implementation of the solution in the IT domain. This phase inherently involves human assistance. However, all routine operations are highly automated to increase human productivity and reduce cost.

- **Generation** of the target solution which includes workflow definition and invoked services. This phase is highly automated – most transformations are done without human intervention.

Workflow to workflow transformation shown as an arrow in the bottom of the Figure 3 denotes a shortcut for the ADM horseshoe lifecycle. In this scenario the workflow definition for one kind of WES is translated into another and no reengineering is performed. Invoked services are also reused without significant modifications.

Modernizing Workflow

From the ADM point of view, a workflow enabled solution is one in which the *activities* within a given process are managed, coordinated, queued, and executed, according to the *workflow definition*. Workflow definition may include data and decision points' logic. Activities invoke services which in turn may contain code representing data, logic and flow.

Workflow modernization involves improvement and/or replacement of flows of activities and services (as well as relevant data and logic) with the new ones that better support business objectives and possibly using different WES. The most important workflow modernization scenarios involve reverse engineering of the existing workflow definition and services to the business model. As a result of modernization the boundaries between activities and services, as well as the packaging of services, may change.

Workflow modernization also includes an important scenario when the existing solution is not workflow enabled (e.g. all process logic is implemented as application code). However, the target solution in this scenario is modeled as collaborating business processes and services within business domain and implemented as workflow- and services-enabled solution within IT domain.

Key workflow aspects and relevant standards

The IT domain workflow model specifies data, logic and flow as shown in Figure 3 above. These three key aspects can be found in any IT domain workflow definition that specifies the *data* (e.g. variables, messages payload, etc.), *logic* (e.g. decision points conditions, operations on the data perform within activity or on its behalf, etc.) and *flow* (a sequence of activities executed either conditionally or unconditionally). The IT services invoked by activities also deal with data, logic and flow.

There are several standards available which represent IT domain workflow model. The most successful standards specifying xml schema for exchange of workflow model between tools are BPEL (a.k.a. WS BPEL) from OASIS [3] and XPDML from WfMC [4]. Both of these standards capture data, logic and flow. The flow visualization is supported by OMG/BPMI BPMN standard [5].

Same three key aspects are relevant for business model. In business domain they are referred to as business vocabulary, business rules and business processes (see Figure 3 above) which are serving the same purpose as data, logic and flow but from different perspective. Currently, there are no standards available which cover all three key aspects for the business domain. However, the new OMG Semantics of Business Vocabulary and Rules (SBVR) standard [6] (adopted by OMG Architecture board in September 2005 but not officially published yet) defines a metamodel for business vocabulary and rules. BPMN supports business process visualization. Another OMG standard – Business Process Definition Metamodel (BPDM) is in the submissions reconciliation phase at the time of publishing this paper. BPDM is concerned with business processes and focusing on orchestration and choreography. SBVR and BPDM combined cover all three key aspects for business model level. However, correspondent OMG task forces have to integrate the respective standard's metamodels. Another possibility is that XPDML/BPMN will play a role in the

business domain but that would require to develop additional capabilities to represent vocabulary and rules in a business friendly format.

Orchestrating services by IT domain workflow

IT domain workflow activities can be executed by the WES or by invocation of a service external to the WES. In the first scenario the WES interprets the workflow definition script itself or by calling an appropriate workflow client (using WfMC Interface 2) to do this work on WES behalf. This scenario is the *invocation of internal service*. In the second scenario the WES calls an external application or service (using WfMC Interface 3), which executes its own program or interprets its own script. This scenario is the *invocation of external service*. Both internal and external services can interact with human or system actors or be autonomous meaning that they do all the work without interaction with any actors. An example of a simple workflow calling internal and external services is shown in the Figure 4 below.

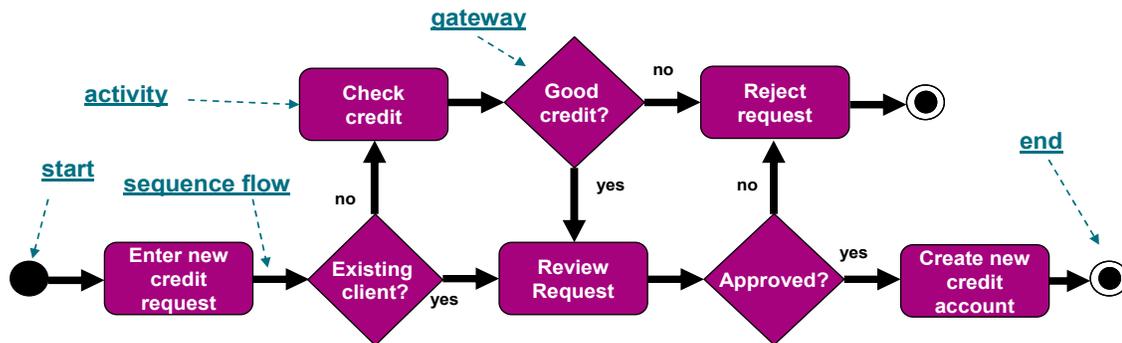


Figure 4. Example of a simple workflow

The example in Figure 4 above is represented using BPMN graphical notation. According to BPMN the workflow in the Figure 4 is represented with the *start and end events, sequence flows, activities and gateways*.

From the ADM point of view all activities in the workflow shown in Figure 4 are performed by calling services - internal or external as following:

- **Activity “Enter new credit request”** – For this activity the WES calls a client application using WfMC Interface 2. Workflow definition specifies a request entry form and the client application manages the interactions with a human actor for data entry. ***This is a case of internal service with human interactions.***
- **Activity “Check credit”** - For this activity the WES calls an external application using WfMC Interface 3. This external service interacts with a Credit Check Subsystem (system actor), which in turn performs a credit check for a new client. ***This is a case of external service with system interactions.***
- **Activity “Review request”** - For this activity the WES calls an external application using WfMC Interface 3. This external service interacts with a human actor according to its own program logic and/or interprets its own script to review the credit request. ***This is a case of external service with human interactions.***
- **Activity “Reject request”** - For this activity the WES calls a client application using WfMC Interface 2. The workflow definition specifies the logic which is executed by the client application. It consists of filling in a predefined credit request rejection form and calling an email subsystem with the instructions to send the rejection form to a predefined distribution list. ***This is a case of internal service with system interactions.***
- **Activity “Create new credit account”** - For this activity, the WES calls an external application using WfMC Interface 3. This is external service does not interact with any

actor (human or system), but rather executes a transaction against a database and creates a new credit account ***This is a case of autonomous external service.***

This relatively simple workflow example is very representative of real-life situations and results in the following observations:

1. ***Each activity represents a system use case for the correspondent service.*** For example, the activity “Check credit” represents a system use case for an external Credit Check service (or application). The functional requirements to this service are captured in the system use case specification artifact and the logic is further elaborated in the analysis and design models. In the context of ADM, these system use cases are the requirements artifacts which will be fed into the forward engineering portion of the ADM Horseshoe lifecycle in case the services have to be replaced with new implementations.
2. ***Some activities represent an embedded workflow (sub-process) that can be modeled as a separate process diagram.*** For example the activity “Approve request” can be represented as a sub-process depicting workflow managed by the external Request Approval service (or application). The sub-process will describe the service’s internal business logic for managing interaction with a human to approve the credit request. The important implication of this, is that the logic of the sub-process is not accessible to the WES, since it is outside of the scope of the workflow definition. In the context of ADM, the only way to reverse-engineer this business logic into the business process model is to extract it from the external service code. Therefore, the knowledge required to build the complete workflow model comes from two independent sources – the workflow definition script and the source code of the external service.

Mapping between workflow and business process

The workflow and services related artifacts (i.e. workflow descriptions, services code, workflow and services models and services code inventory) belong to the IT domain because they define only the automated processes/activities and supporting IT processes/activities that have no business significance. The business models including the ones for workflow and for services belong to business domain because they define processes/activities that have business significance.

The diagram in the Figure 5 below shows the reverse engineering of the existing workflow definition and the invoked services implementation to the as-is business model which is accomplished in three steps:

Step 1. The existing workflow definition is automatically transformed to the as-is workflow model. The existing services code is automatically transformed to the as-is services *code inventory*. Code inventory is a code digest produced by a mining tool. Code inventory shall follow the upcoming OMG Knowledge Discovery Metamodel (KDM) standard [7] to achieve higher level of tool independence.

Step 2. The as-is workflow model is generalized into the as-is business model for workflow. The as-is services code inventory is generalized into the as-is business model for services using KMA approach [8]. Both generalizations produce business models expressed in terms of business vocabulary, rules and processes and require manual effort since they involve abstraction. However, all routine operations are highly automated to reduce the cost.

Step 3. The as-is business models for workflow and for services are merged into as-is business model. In the resultant model the business processes discovered from services are included as sub-processes into the main business process derived from the existing workflow definition.

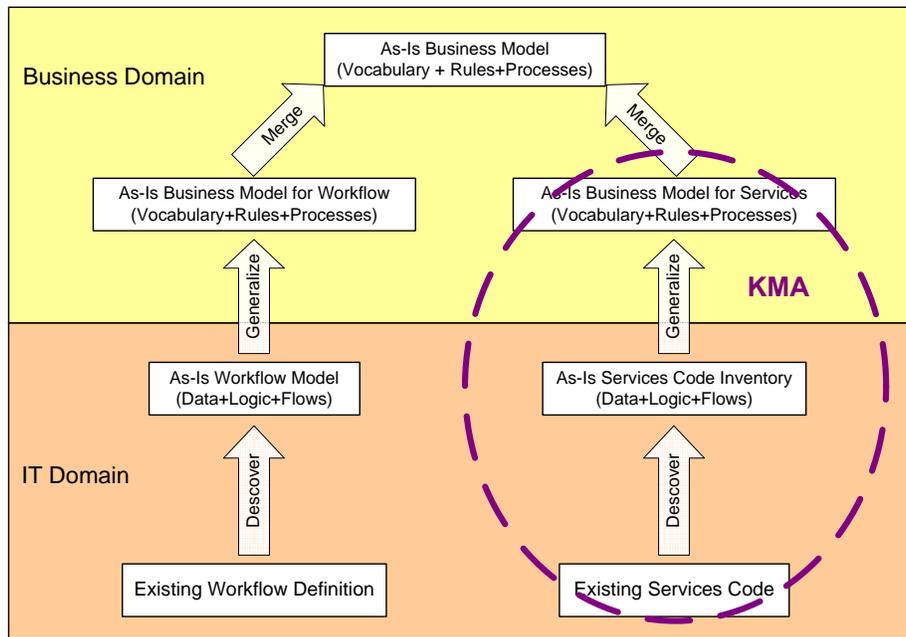


Figure 5. Reverse engineering: discovery and generalization

The diagram in the Figure 6 below shows the forward engineering of the existing workflow definition and the invoked services implementation to the as-is business model which is accomplished in three steps:

Step 1. The to-be business model is split into the to-be business model for workflow and the to-be business model for services based on the newly developed and optimized workflow-services boundary definition that includes services orchestration and choreography.

Step 2. The to-be business model for workflow is specialized into the to-be workflow model. The to-be business model for services is specialized into to-be services model. Both specializations produce IT domain models expressed in terms of data, logic and flows and require manual effort because it involves crossing the business and IT domain boundary.

Step 3. The to-be workflow model is automatically transformed to the to-be workflow definitions. The target service code is generated from the to-be services model automatically (using MDA) or manually (using traditional forward engineering SDLC).

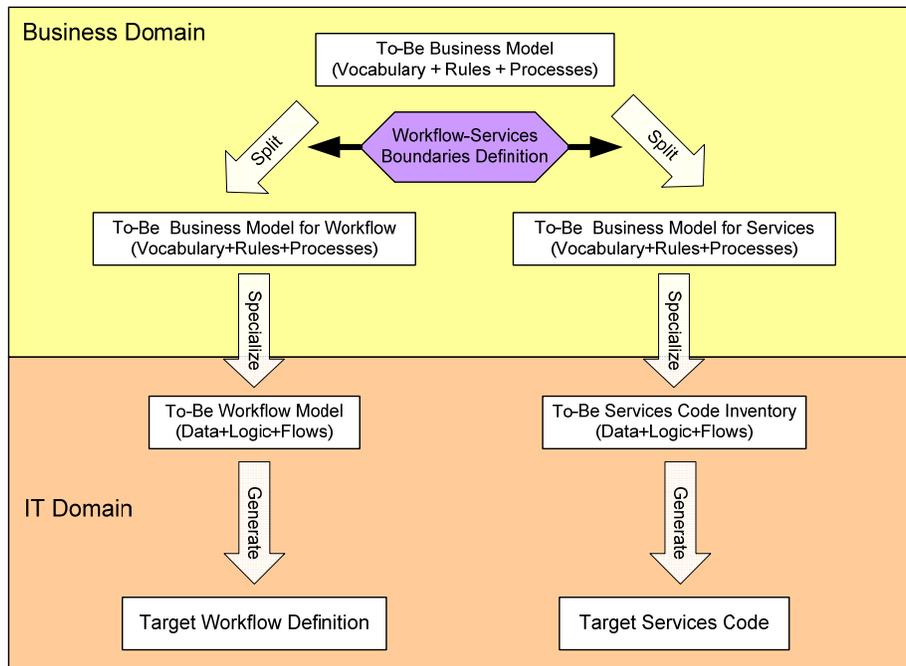


Figure 6. Forward engineering: specialization and generation

Workflow modernization scenarios

Workflow modernization includes but is not limited to the following major scenarios:

- Basic Workflow Modernization
- SOA Workflow Modernization
- Workflow and SOA-enablement

Basic Workflow Modernization. Existing workflow definition for the source WES is transformed to as-is workflow model which is mapped to the as-is business model, the latter is enhanced with the new business requirements and optimized based on the business defined criteria, and then the resultant to-be business model is mapped back to the to-be workflow model which is transformed to the target workflow definition for the target WES.

SOA Workflow Modernization. Same as Basic Workflow Modernization but additionally all internal and external services are mined and abstracted to the business vocabulary, rules and processes and consequently merged with the as-is business model. The mapping from the to-be business model to the to-be workflow model involves definition of the new boundary between workflow and services and correspondent services interfaces. This new boundary may differ from the original boundary between workflow and services as a result of business model optimization and re-definition of the business services.

Workflow and SOA-enablement. The existing solution is not workflow-enabled, but is designed as a system of programs supporting the business processes. The code of the existing solution is mined and abstracted to the as-is business model expressed in terms of business vocabulary, rules and processes. The to-be business model is then mapped to the to-be workflow model and the newly defined services and corresponding services interfaces. This scenario supports modernization of legacy applications (designed to address all aspects of the supported business model using program code) into Service-Oriented Architecture solution where services orchestration is handled by the WES or external BPEL engine.

Conclusion

Workflow modernization is a special case of ADM. It follows the ADM horseshoe lifecycle. The reverse engineering part of workflow modernization involves generalization of workflow model belonging to the IT domain to the business model belonging to the business domain. It also involves KMA for the invoked applications to enhance as-is business model with business vocabulary, rules and processes discovered from these services. The business model upgrading and optimization results in to-be business model which better supports agile business objectives. The forward engineering part of workflow modernization involves definition of new services and orchestrating them workflow.

Organizations may benefit from workflow modernization by applying a number of scenarios including three major ones: Basic Workflow Modernization (workflow to workflow), SOA Workflow Modernization (workflow + services to workflow + services) and Workflow- and SOA-enablement (program system to workflow + services).

References

- [1] Khusidman, Vitaly and Malhotra, Sumeet. *IT Modernization Framework*. Architect Newsletter. Volume 4. Issue 2, page 1
- [2] Workflow Management Coalition. *Workflow Reference Model*. <http://www.wfmc.org/standards/model.htm>
- [3] Organization for the Advancement of Structured Information Standards. *Business Process Execution Language*. <http://www.oasis-open.org/committees/download.php/14616/wsbpel-specification-draft.htm>
- [4] Workflow Management Coalition. *XML Process Definition Language*. <http://www.wfmc.org/standards/XPDL.htm>
- [5] Object Management Group / Business Process Management Initiative. *Business Process Modeling Notation*. <http://www.omg.org/docs/bei/05-08-07.pdf>
- [6] Object Management Group. *Semantics of Business Vocabulary and Rules*. <http://www.omg.org/cgi-bin/apps/doc?dtc/06-03-01.pdf>
- [7] Nikolai Mansurov. *Knowledge Discovery Metamodel*. Tutorial. ADM workshop, 24th October 2005, Washington, DC. http://www.omg.org/news/meetings/workshops/ADM_2005_Proceedings_FINAL/T-2_Mansurov.pdf
- [8] Khusidman, Vitaly and Costello, Scott. *Knowledge Mining and Abstraction for Business Rules*. Architect Newsletter. Volume 4. Issue 4, page 21.

Vitaly Khusidman is the Director of the Unisys Corporation Architecture Driven Modernization Program. He can be reached at vitaly.khusidman@unisys.com