Cross-Component Business Process Management (ccBPM) within SAP NetWeaver Process Integration 7.1

SAP NetWeaver Product Management
SAP NetWeaver Regional Implementation Group

March 2008
1. ccBPM Overview
2. New Design Features
3. New Runtime Features
4. ccBPM in a Nutshell
Model-Based Business Process Management on a Business Process Platform

End-to-end business processes are seamless combinations of different types of processes, as:

- **Core application or platform processes** that provide proven standard business practices and are delivered by SAP’s applications. These packaged processes are built to satisfy a high demand on integration, integrity and legal compliance, and are typically mission-critical. They are designed as integral part of a business process platform.

- **System-centric integration processes** that include A2A and B2B interactions, legacy system or third part system integration. These processes are technically delivered through SAP NetWeaver Process Integration (PI).

- **Human-centric composite processes** focusing on human interactions and process collaboration. These processes are delivered through a new building block of BPM capabilities that is integrated into SAP NetWeaver Composition Environment (CE).

- All these types of business processes interact seamlessly through **services and events**.

**Building Blocks of SAP NetWeaver BPM Capabilities**

The **SAP NetWeaver Composition Environment** will support a standards-based modeling environment (BPMN, Business Process Modeling Notation), process design collaboration, semantic integration with SAP’s application core processes, human interaction management that provides task management, rule and responsibility assignments and business event resolution mechanisms.

The **Enterprise Services Repository** (ES Repository) is shipped in 2007 with SAP NetWeaver Process Integration 7.1 and SAP NetWeaver Composition Environment 7.1. As an evolution of the SAP NetWeaver Exchange Infrastructure Integration Builder, it does not only contain enterprise services (service interfaces, service operations) and data types but also tools to cover the integration needs of a SOA middleware (e.g., a mapping editor, a BPEL integration process editor) Apart from this, the ESR is used as the central place for **process component architecture** modeling with SAP modeling methods and content for enterprise SOA.
Cross-Component BPM (delivered with SAP NetWeaver Process Integration) handles processes where the message flow between different business applications is dependent on several messages or on time and business actions or reactions. Interdependencies can be defined using an internal state derived from content of incoming messages. Messages belonging to one process instance are identified by correlations as common denominators on the basis of message content (e.g. a Purchase Order, an ASN, a Confirmation and an Invoice in a procurement process via the Order ID in combination with the business partner ID and/ or the company code). ccBPM also supports handling of system exceptions and alerting.

SAP Business Workflow has been embedded in the heart of the SAP solutions, so that SAP applications (including CRM, SRM and ERP) are built on it to incorporate workflow features directly in their application. Integration with Organizational Management and with standard SAP reporting tools allow to reuse the investment that customers have made in SAP solutions.

Universal Worklist (UWL): a work-item inbox/ task list is the watering hole where users access their workflow “to-do” lists. Work items that appear in these inboxes can span a range of business activities, from administrative processes like a vacation request to more in-depth processes like the evaluation of a sales opportunity.
Enterprise Services Builder is the central tool for the design and configuration of integration processes. It contains the Integration Directory for configuration purposes as well as the Enterprise Services Repository, an evolution of the Integration Directory, for business process definition. Within Enterprise Services Repository all information regarding interfaces, mappings, data types, integration scenarios and integration processes are stored.

In the Integration Directory the configuration of the cross-system processes for an existing system landscape takes place. The configuration describes how the Integration Server is to process inbound messages and to which receiver or receivers messages must be sent.

Various engines are involved in message processing on the Integration Server:

- Integration Engine, the central runtime component
- Adapter Engine, runtime component for adapter communication
The BPEL process editor within the Enterprise Services Builder consists of:

- Header – contains several information about the object
- Edit Area – to display and change objects, several view available (Graphical Definition, BPEL Display, Correlation Editor)
- Output Area – to check whether an object is complete and works properly together with objects referenced
- Object Area – provides several views (Containers, Correlation List, Process Signature, Configurable Parameters)
- Property Area – to edit properties of an object
- Birds view – to show alternative representations of the business process
An integration process comprises individual steps. There are various different step types available for the definition of an integration process:

- **Receive Step** – starts an integration process, receives messages, define sync/async communication
- **Send Step** – send messages synchronously / asynchronously, send acknowledgements, define sync/async communication
- **Transformation Step** – convert message into another message, bundles or splits messages
- **Receiver Determination Step** – get a list of receivers (configured in Integration Directory) for a subsequent send step
- **Container Operation** – set value for a target container element at runtime
- **Control Step** – terminate current process, trigger an exception, trigger an alert
- **Loop** – repeat the execution of steps within the loop, runs while end condition returns “true”
- **Fork** – to continue process in branches, independently from each other
- **Block** – to combine steps which should be executed one after the other and which use the same local data, or have the same deadline, exception handler
- **Undefined Step** – has no influence to the process flow, acts like a placeholder
- **Wait Step** – to incorporate a delay in a process, can be defined as either a point in time or a period of time
- **Switch Step** – to define different processing branches for a process, based on a condition for each branch
Correlation Handling

- Routing of messages (Business Documents) to process instances
- Dependencies rely on business data (e.g. Message ID, Order ID, Business Partner ID, Company Code)
- Correlations define these dependencies

You use a correlation to assign messages that belong together to the same process instance. An integration process is to process a purchase order and the corresponding sales order, for example. The purchase order and the sales order have the same purchase order number. To ensure that the purchase order and the corresponding sales order are processed by the same process instance, you define a correlation using the purchase order number.

Receive and send steps can activate or use correlations. First you must define a correlation before you can enter it in a receive or send step.
Usage of ccBPM

Integrate processes within and across systems
- Enable service-based integration end-to-end today
- Endorse composite applications
- Provide central orchestration – loose coupling of legacy systems

Enable model-driven process flexibility
- Model-driven process automation
- Usage of pre-defined content
- Reduce adaptation time by changing models (zero coding approach)

Typical use cases
- B2B scenarios with request/response interaction of business partners
- Service Orchestration for composite applications
- Synchronous-asynchronous mainframe integration

Business Process Management
- Links models with execution
- Provides configuration
- Controls process execution
- Facilitates model-driven monitoring
- Supports governance & standardization

… for Service Orchestration
- Decouples process logic from services
- Supports A2A, B2B integration and human workflow in a unified manner
- Applicable for inter-SAP applications & composites
- Applies open standards
1. ccBPM Overview
2. New Design Features
3. New Runtime Features
4. ccBPM in a Nutshell
With SAP NetWeaver Process Integration 7.1, a lot of new modeling concepts are introduced to ccBPM (cross-component Business Process Management)

Basic extensions
- Besides the Integration Process, new Process Types are offered to allow fine granular check services depending on the Process Type (Integration, Monitoring, …)
- Language dependent text for all modeling elements allow better documentation for all SAP delivered processes
- Configurable Parameters allow new concepts such as parameterization in general (see conditions) and user decision

Step Groups for speeding up process modeling

User Interaction
- User Decisions
- Alert Categories

Conditions and expressions
- Extended capabilities
- Advanced User Interface

Standards
- Process Definition Standards
  - BPEL4WS 1.1 (specification/already adopted)
  - WS-BPEL 2.0 (specification / preview / implementation)
Language Dependent Texts

- Definition of language-specific texts in Integration Processes
- Define alerts or describe steps and translate these texts

Language dependent texts support describing processes in native language.

Define the original language in the software component version and also the target language for the translations.

Edit the language specific text and translate the text of the object -> “Display Translation for Object”

After designing the process, change the display language and get the description of process elements in the defined language.
Configurable Parameters

- Configure the value of a parameter
- No need to change the process definition if value must be changed later
- Enables multiple processes using the same process definition, but with different parameters

Configurable Parameters enable the configuring of the values of a parameter in the Integration Process component and in the Integration Directory.

Therefore, if the value must be changed later on, the process definition does not have to be changed.

Also, it is possible to create multiple configurations for one process and define different values for a configurable parameter in each process.

Define configurable parameters in the same way as container elements.

Select in the object area of the process editor the Configurable Parameters view. Then give the parameter a name and choose the required category and type:

- Agent – Specifies the agent of a user decision
- Simple Type – Configurable parameter to use in conditions
- Adapter – Specifies the communication channel when programming a mapping lookup that will be used in a transformation step
Step Groups

- Step groups enable creation of reusable templates
- Design time artifact – speeds up modeling
- Pre-setting of some simple properties
- Global availability
- Set to changeable / not changeable

Step groups speed up process modeling, as it is possible to define templates or typical process patterns which can be reused at design time.

While defining a step group, some simple properties can be predefined and then later on adapted to the current process design.

The change of the properties can also be prohibited if necessary and only the preset properties can be used in the process.

An advantage of the step groups is their global availability. This means that once defined, a step group can be used in the whole repository.

When using a defined step group at process design, the group itself can be expanded or collapsed.
Another new feature is the creation of alert categories directly in the Integration Process.

Previously, it was necessary to use transaction code ALRTCATDEF and define the alert, then use this definition in the Integration Process.

Now create an object of type “Alert Category” and use this in the Integration Process. After creating the alert, recipients are determined in ALRTCATDEF.
Extended Conditions and Expressions

- Define conditions to control processing depending on the result of the condition
- The condition editor supports the definition of the condition
- Insert comments to ensure that conditions remain clear and easy to understand

With the help of conditions, you are able to control the flow of a process. Depending on the result of the defined condition, the process will be executed in the defined way.

Use the step type “Switch” and use “Condition” in the properties.

In the integrated Condition Editor, you can formulate logical conditions.
User Decisions

Generic user decision during the execution of an Integration Process

- Deadline occurs, or an alert is thrown, option needed to directly interact
- Provides decision gateway and outcomes of the user decision, and uses the Integration Directory for responsibility determination
- Uses container elements type “agent” from the configurable parameters
- Assigned user receives a dialog work item in the workflow inbox at runtime

Use the new step type “User Decision” to enable a user to decide which branch of a process flow should be executed.

For each decision option, a branch is entered.

At runtime, the user will be notified by a dialog work item in the workflow inbox.

To define the text to be displayed for the user, it is possible to use local variables. At runtime, these variables will be replaced with the current data from the process.

Within “User Decision”, you will use the introduced “Configurable Parameters” to determine which user should be notified at runtime. Therefore, you use the configurable parameter type “Agent”.

© SAP AG 2008
**BPEL 2.0 and BPEL4People**

**Standard Support**
- BPEL4WS 1.1 (specification / already adopted)
- WS-BPEL 2.0 (specification / preview / implementation)
- Plans to support BPEL4People

**SAP is**
- A leader in BPEL-Standard adoption
- A driver of the BPEL-Standard
  - SAP was one of the proposers of the OASIS WS-BPEL Technical Committee
  - SAP works together with IBM on BPEL standard extensions

SAP joined the BPEL initiative in March 2003 and it is
- Co-author of the BPEL4WS 1.1 specification
- Early adopter of BPEL4WS 1.1
- SAP NetWeaver customers are live and in production with BPEL4WS 1.1
- Co-author of the WS-BPEL 2.0 proposal

SAP was one of the proposers of the OASIS WS-BPEL Technical Committee

SAP actively participates in the OASIS WS-BPEL Technical Committee

SAP works together with IBM on two significant BPEL Extensions:
- How people interacts with BPEL processes (people interaction patterns go beyond simple Web services calls)
- Modularization and reuse in WS-BPEL
1. ccBPM Overview
2. New Design Features
3. New Runtime Features
4. ccBPM in a Nutshell
New Runtime Behavior of the BPE

Delivery Mode
- Delivery of messages to receive steps to running process instances
- Available with
  - SAP NetWeaver 2004 SPS 17
  - SAP NetWeaver 7.0 SPS 8

Queue Assignment
- Parallelization of process execution
- Available with
  - SAP NetWeaver 2004 SPS 18 and SAP NetWeaver 7.0 SPS 09 (configurable queue)
  - SAP NetWeaver 2004 SPS 19 and SAP NetWeaver 7.0 SPS 10 (multiple queues)

Flexible Hibernation / Transaction Handling
- Block-oriented handling of persistency within the transactional concept
- Available with
  - SAP NetWeaver 2004 SPS 19
  - SAP NetWeaver 7.0 SPS 10

BPE Message Bulking
- Available with
  - SAP NetWeaver 7.0 SPS 13
  - SAP NetWeaver Process Integration 7.1 SPS 4

The following parameters to configure the BPE runtime behavior have been introduced:
- Delivery Mode: Governs the delivery of messages to receive steps to running process instances
- Queue Assignment: Permits a parallelization of process execution
- Transaction Handling: Defines the sync-points with the database
- BPE Message Bulking: Provides a mass delivery of messages to receive steps

Those parameters and their impact on the process design and the process runtime (BPE) will discussed in detail.
**Delivery Mode**

Defines whether messages are handed over to process instance directly or with intermediate buffering

**With Buffering**
- Intermediate message storage if process instance does not provide an open receive step
- All messages that fit to an active correlation are taken up by a running process instance
- Message might be captured and stored but is never processed (long running processes, erroneous processes with active correlations)

**Without Buffering**
- Process instance has to provide an open receive step when message is handed over to BPE
- If instance does not provide an open receive step, the process type specific queue (XBQO$PE_WS...) goes to status “SYSFAIL”

The parameter Delivery Mode governs the delivery of messages to receive steps to running process instances.

The parameter can take two values “With Buffering” and “Without Buffering”

**Setting With Buffering**
- Additional storage of messages: If message to be delivered to running process cannot be taken up by a receive step, it is buffered
- The following receive step will take up buffered message
- Drawback: If no further receive step is available, the message is captured by the process and might get lost

**Setting Without Buffering**
- Message is delivered directly to waiting receive step
- Developer has to guarantee that for each delivered message there will be an open receive step
- Drawback: If instance does not provide an open receive step, the process type specific queue (XBQO$PE_WS...) goes to status “SYSFAIL”
Queue Assignment

Defines whether messages are handed over to process instance using single or multiple queues

<table>
<thead>
<tr>
<th>Single Queue</th>
<th>Multiple Queues</th>
</tr>
</thead>
</table>
| - All messages dedicated to a specific process type are handled by a single queue. | - Using multiple queues per process type  
  - Arbitrary distribution to queues: “Multiple Queues (Random)”  
  - Distribution based on correlations: “Multiple Queues (Content-Specific)” |
| Works for all process definitions | Might conflict with correlation handling |

Queue Assignment: Permits a parallelization of process execution

Technically spoken, it defines whether messages are handed over to process instance using single or multiple queues.

Multiple queues may either use
- Arbitrary distribution to queues (and hence process instances). Mainly used for split processes.

Or
- Content-based distribution (required for all process types using correlations). Mainly used for collect processes.
Transaction Handling

Adjustable transaction handling

So far:
- Pessimistic assumption regarding transactional behavior
- Required numerous creation of work items and execution DB actions

Now:
- Developer decides whether
  - A step creates new transaction
  or
  - The steps are executed synchronously
- Configurable DB sync points

- Developer decides whether
  - A step creates new transaction
  or
  - The steps are executed synchronously
- Configurable DB sync points
BPE Message Bulking

Delivery of message bulks to BPE inbound processing

Advantages
- Delivery of multiple messages to process instance in a single transaction
- Raises message throughput (but: latency of single message may increase)
- Reduces persistence effort
- Reduces occupied DB space

Delivery of message bulks to BPE inbound processing in order to increase message throughput.
1. ccBPM Overview
2. New Design Features
3. New Runtime Features
4. ccBPM in a Nutshell
Cross-Component BPM in a Nutshell

- Design, execute and monitor automated processes across applications
  - Provides process control in the central technology layer
  - Graphical Process Editor
  - BPM Runtime: BPE

- Integral part of Process Integration
  - Enterprise Services Repository (Design)
  - Integration Directory (Configuration)
  - Integration Server (Runtime)

- Adheres to open modeling standards
  - Business Process Execution Language (BPEL)

- Technical Process Monitoring
  - Integrated with technical Monitoring of PI
  - Graphical Process Monitoring