

Essential Aspects of Compliance Management with Focus on Business Process Automation

3rd International Conference on
Business Process and Services Computing
September 27, 2010

David Schumm,
Steve Strauch



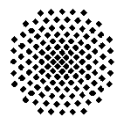
Tobias Anstett,
Daniel Schleicher



Frank Leymann



Institute of Architecture
of Application Systems



Universität Stuttgart
Germany

Overview

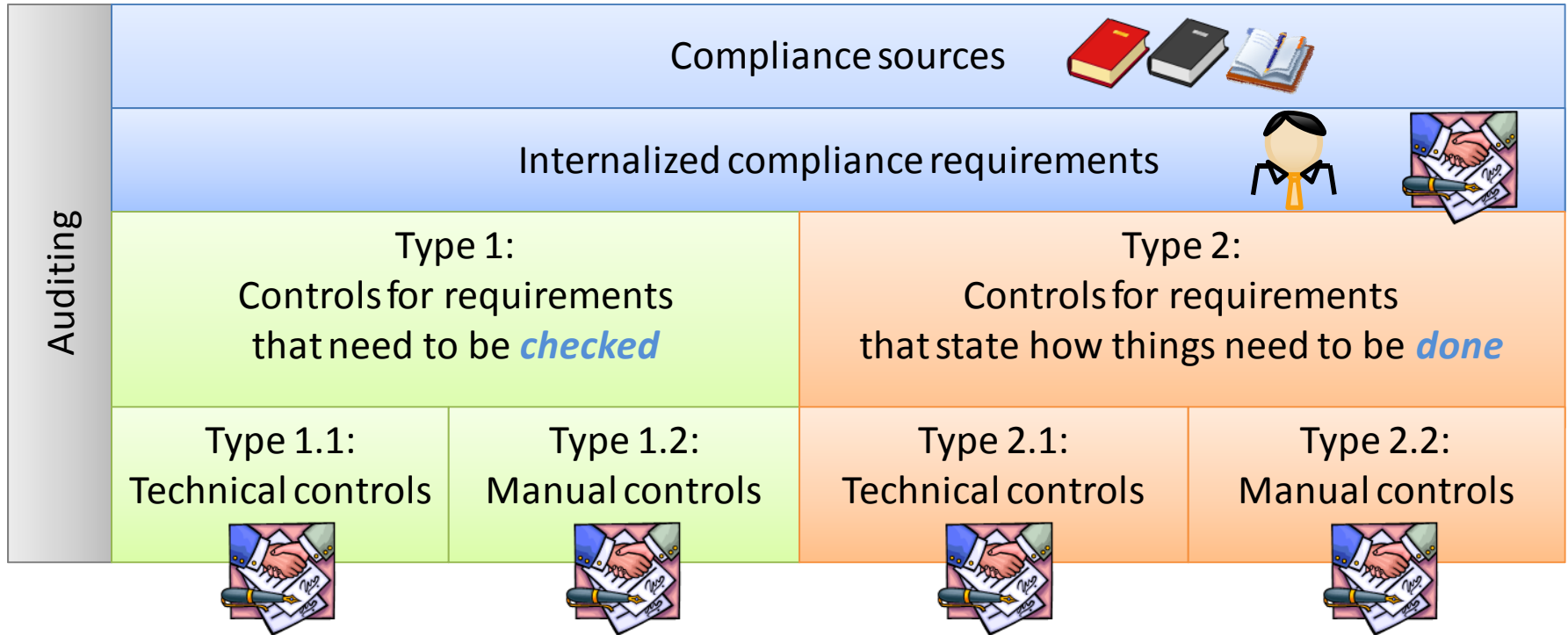
- Compliance Management
 - What is Compliance?
 - How can Compliance be managed?
- Focus on Business Process Automation
 - Impact of Compliance on the BPM lifecycle
 - Compliance Fragments
 - BPEL Extensions for Compliance Fragments
 - Integrating Compliance Fragments into Workflows
- Conclusion and Outlook

Compliance Management

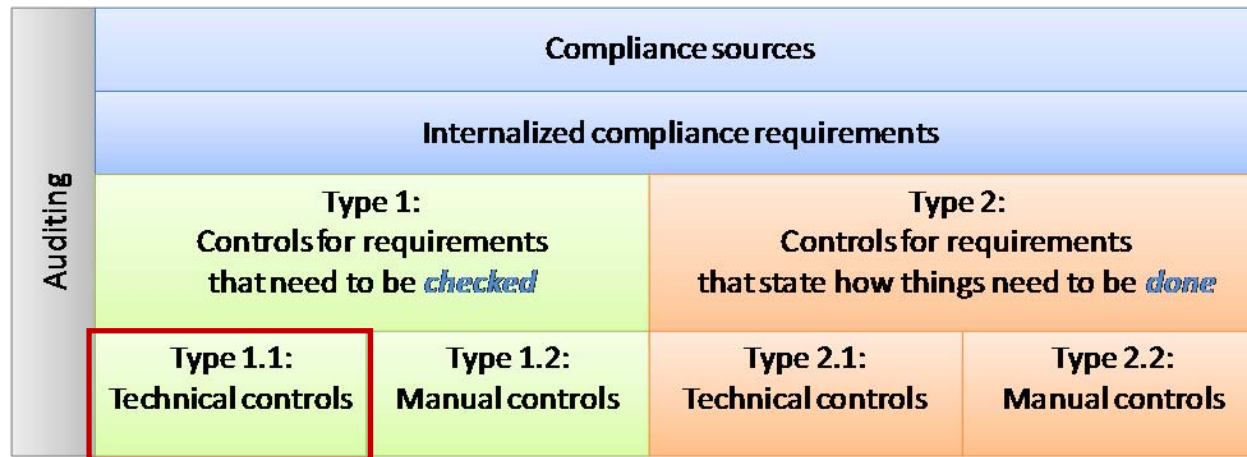
Compliance

- Compliance requirements coming from laws, regulations etc. constrain, how a company may carry out its business
- Various different actions have to be taken to prevent and detect compliance violations
- Compliance has an impact *on all components* in the IT infrastructure of a company
- Compliance also has an impact on the business processes which run *outside of IT systems*
- Compliance management ensures that business processes are in accordance with a set of prescribed requirements

Compliance Management Model



Controls for Requirements that need to be checked



- Functionality for *checking compliance within IT systems*
 - Time of checking (Design time, Runtime, Offline)
 - Specification of constraints (logical languages, DSLs, ...)
 - Appropriate software need to be installed (model checkers, CEP, ...)
- Impact of these controls
 - In general they do not change the behavior of the IT system
 - But: the information which is required for checking has to be available

Controls for Requirements that need to be checked

Auditing	Compliance sources			
	Internalized compliance requirements			
	Type 1: Controls for requirements that need to be <i>checked</i>		Type 2: Controls for requirements that state how things need to be <i>done</i>	
	Type 1.1: Technical controls	Type 1.2: Manual controls	Type 2.1: Technical controls	Type 2.2: Manual controls

- Functionality for *checking compliance outside IT systems*
 - Interviews
 - Questionnaires
 - Anonymous complaint boxes
 - “Whistleblowing hotlines”
 - Several forms of auditing
 - Sample checks by a compliance officer
 - Technical audit of a software framework
 - Financial audit

Controls for Requirements that state how things need to be done

Auditing	Compliance sources			
	Internalized compliance requirements			
	Type 1: Controls for requirements that need to be <i>checked</i>		Type 2: Controls for requirements that state how things need to be <i>done</i>	
	Type 1.1: Technical controls	Type 1.2: Manual controls	Type 2.1: Technical controls	Type 2.2: Manual controls

- Measures for *enabling compliance within IT systems*
 - General requirements: Security, privacy and trust
 - Hardening the environment and installation of security components
 - Ensuring particular service levels and policies
 - Custom settings of applications and data bases
 - Augment applications with functions related to compliance
 - *Augment processes* with process structures related to compliance

Controls for Requirements that state how things need to be done

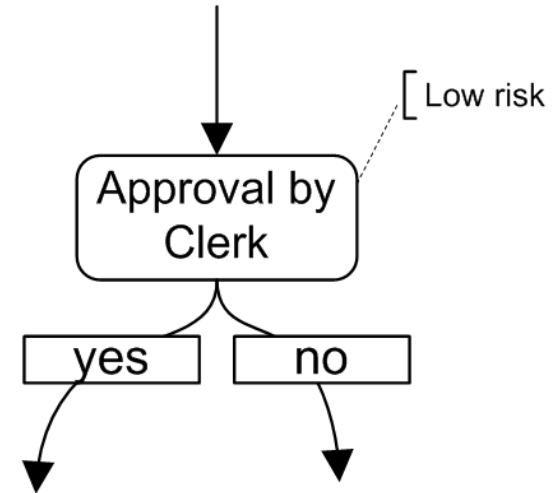
Auditing	Compliance sources			
	Internalized compliance requirements			
	Type 1: Controls for requirements that need to be <i>checked</i>		Type 2: Controls for requirements that state how things need to be <i>done</i>	
	Type 1.1: Technical controls	Type 1.2: Manual controls	Type 2.1: Technical controls	Type 2.2: Manual controls

- Measures for *enabling compliance outside IT systems*
 - Management of the facility security
 - Training of employees to adhere to guidelines
 - Changes in the organization
 - Increasing number of employees concerned with compliance
 - Installation of a compliance office
 - Inter-organizational compliance management
 - Ensuring compliance of business partners
 - Business contracts

Focus on Business Process Automation

Compliance Fragments: Concept

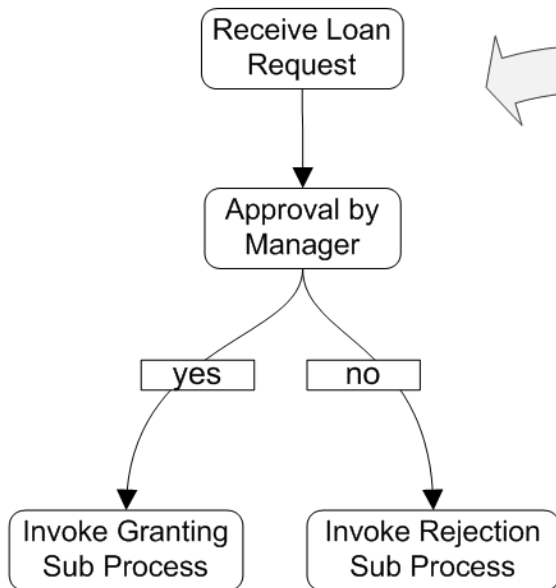
- A compliance fragment is a connected, possibly incomplete process graph
- A compliance fragment can be used as a building block to realize compliance requirements in terms of process logic
- It may be partially undefined
- A compliance fragment is made up of
 - Activities
 - Activity placeholders (so-called regions)
 - Multiple incoming and outgoing control edges
- A compliance fragment contains a context
 - Variables, handlers, ...



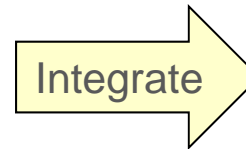
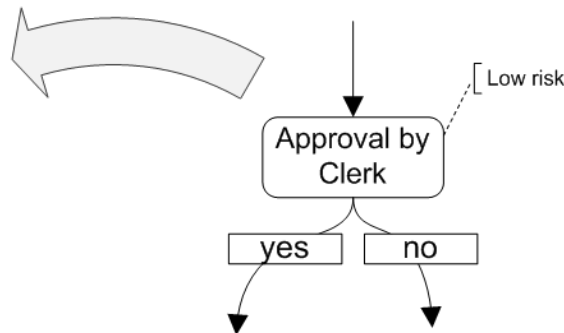
Compliance Fragments: Usage Example

- Augmentation of a loan approval process with a compliance fragment for an additional decision

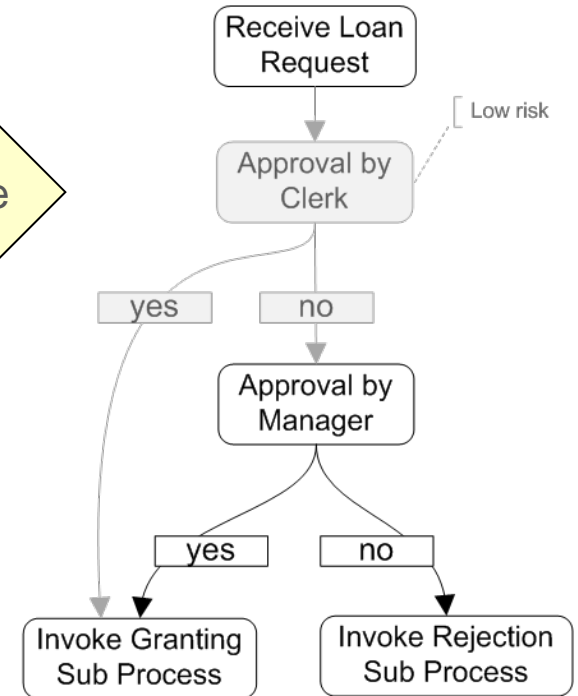
Loan Approval Process



Fragment for Decision



Augmented Loan Approval Process

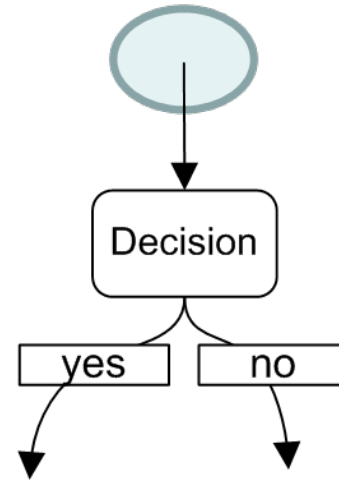


BPEL Extensions for Compliance Fragments

- Some constructs for representing compliance fragments in BPEL are missing
 - Entries
 - Exits
 - Regions (for flexibility)
 - Fragment container (for the context)
 - Unique identifiers (for annotations)
- In the following, we discuss design time BPEL Extensions to represent these constructs

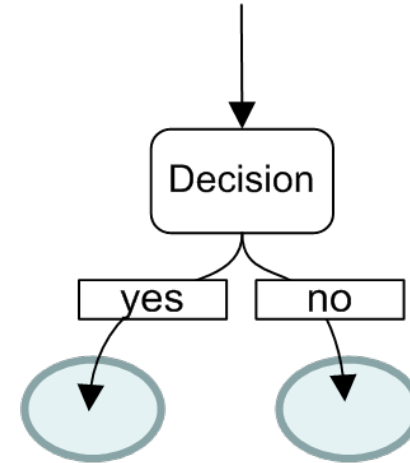
BPEL Extension: Fragment Entry

- Entries need to be declared explicitly for wiring
- They are distinguished into mandatory and optional
- Multiple entries are possible
- An entry needs to be replaced during integration



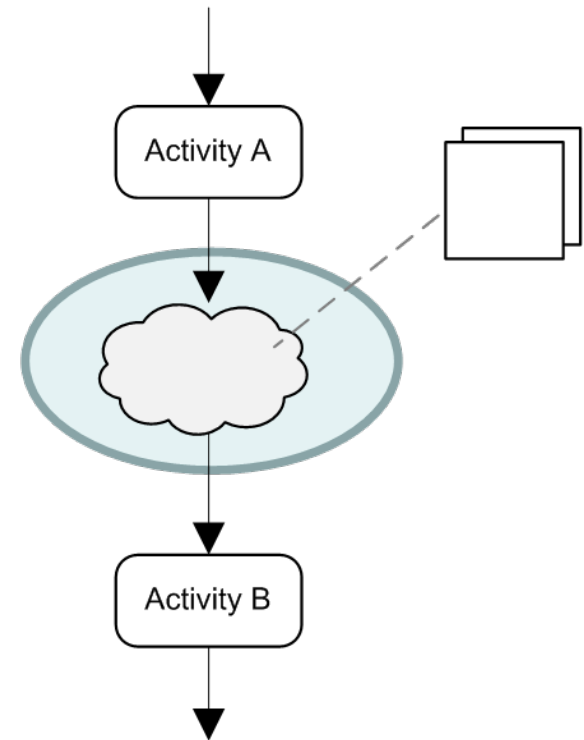
BPEL Extension: Fragment Exit

- Exits also need to be declared explicitly for wiring
- They are also distinguished into mandatory and optional
- Multiple exits are possible
- An exit needs to be replaced during integration, too



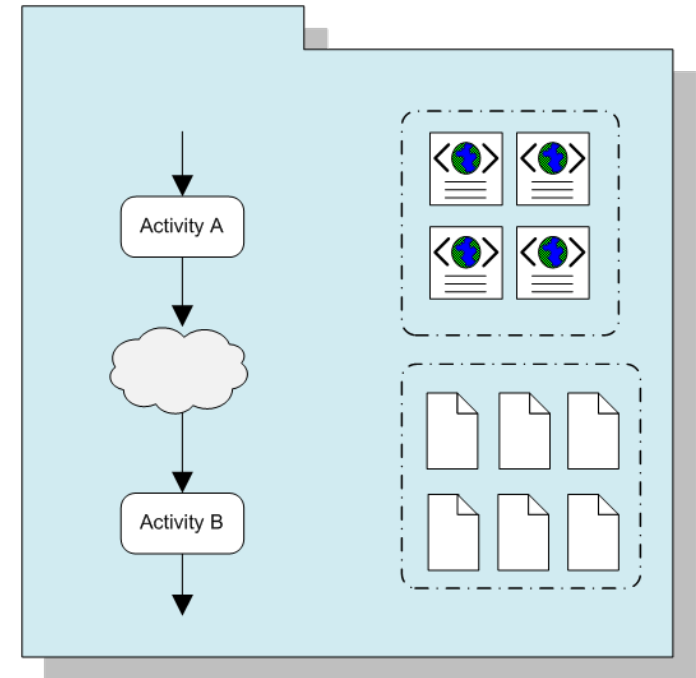
BPEL Extension: Fragment Region

- A region can be used for flexible composition without breaking the fragment design
- Constraints can be imposed from the outside
- A region is replaced during composition



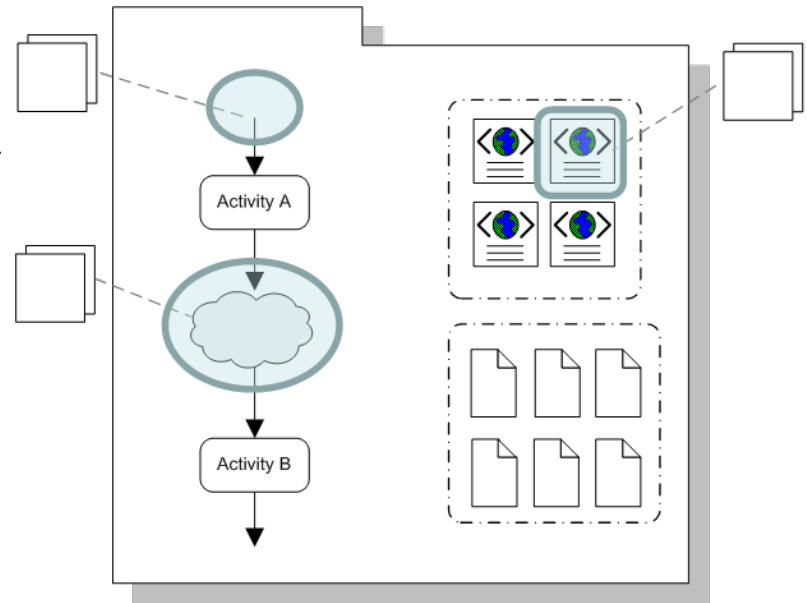
BPEL Extension: Fragment Container

- A fragment defines a *context* like Variables, Correlation Sets, Data Types, PartnerLinks, ...
- We need a container for storing the context: fragment scope
- We need a container for the control structures: fragment flow



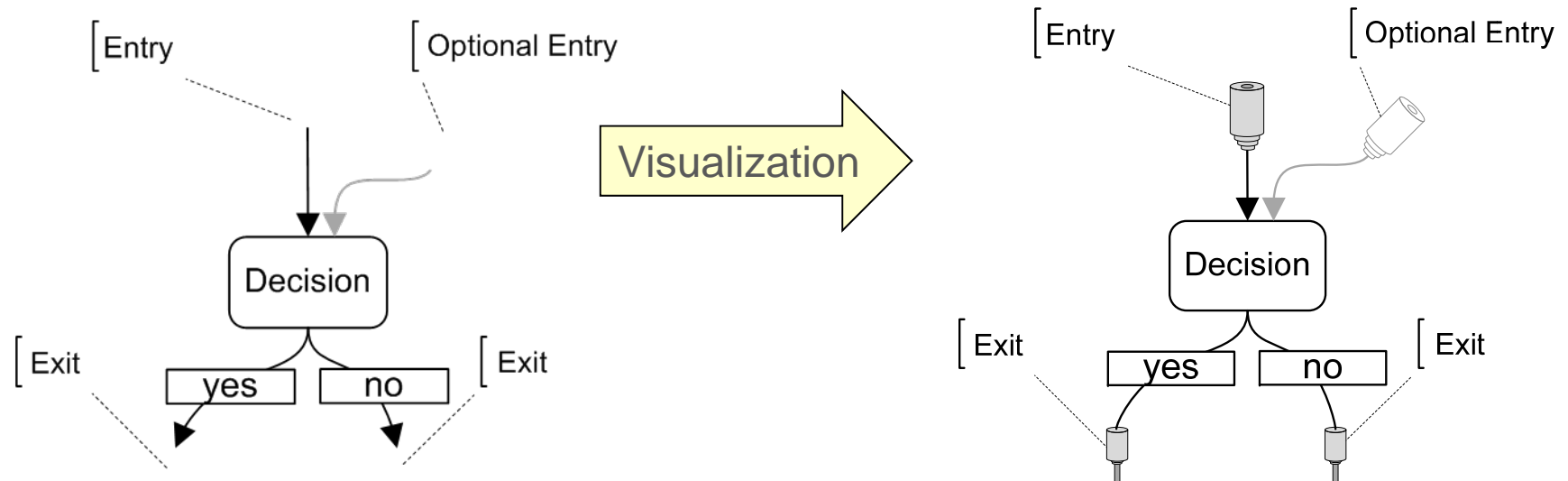
BPEL Extension: Unique Identifiers

- Parameters can be declared externally for keeping the process clean
- Constraints can be imposed from the outside using arbitrary languages (e.g., DSLs)
- Therefore, we need a unique identifier on all constructs



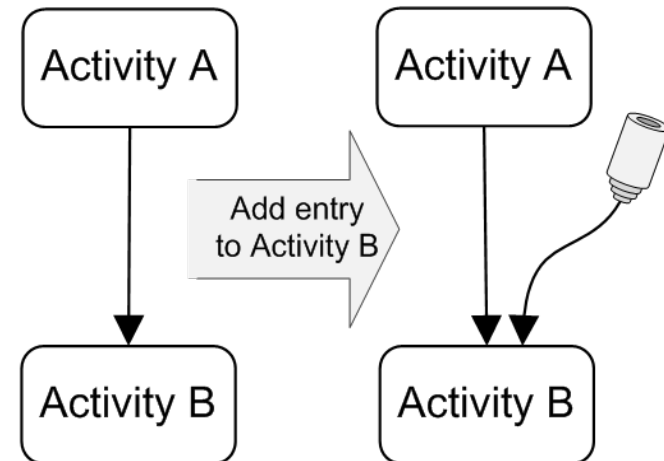
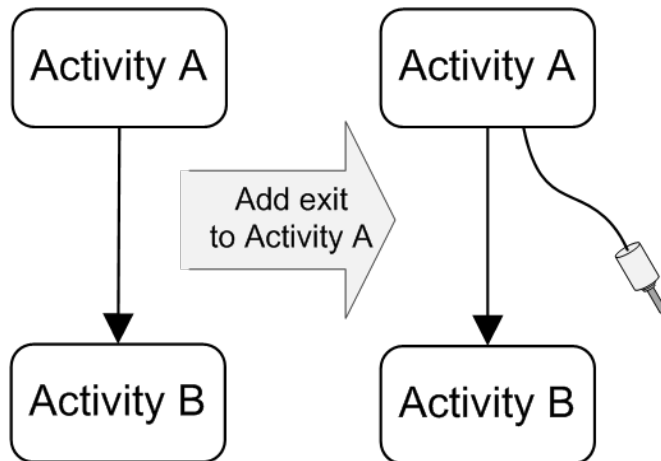
Wiring a Compliance Fragment with a Workflow

- During integration, entries and exits of a fragment have to be wired
- Wiring means, that control links between process activities and fragment activities are being established
- We can illustrate wiring with the *concept of plugs*
 - To ease understanding of fragments on end-user level
 - Different forms of plugs conceivable



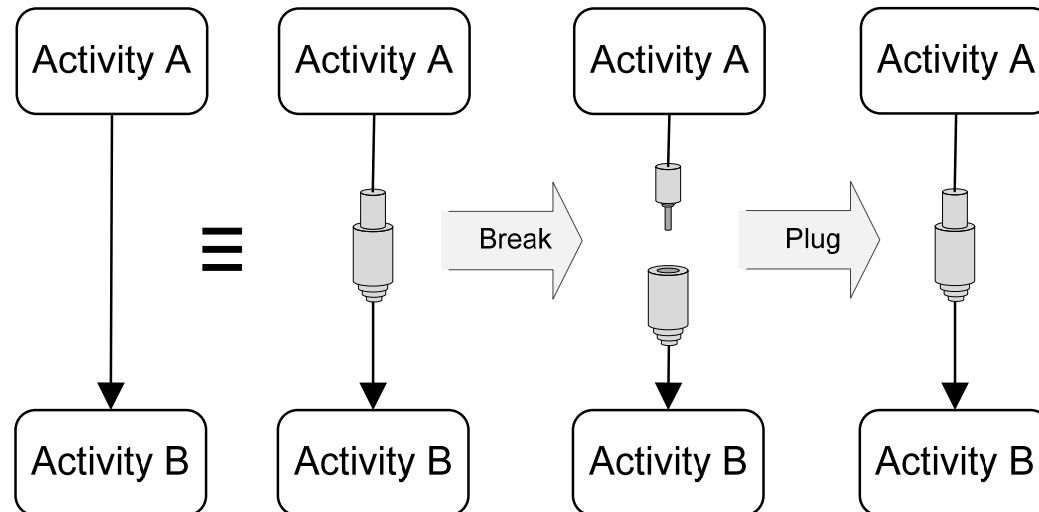
Adding Additional Connector Plugs

- New connector plugs can be created
- Available entries or exits from another incomplete integration operation can also be used



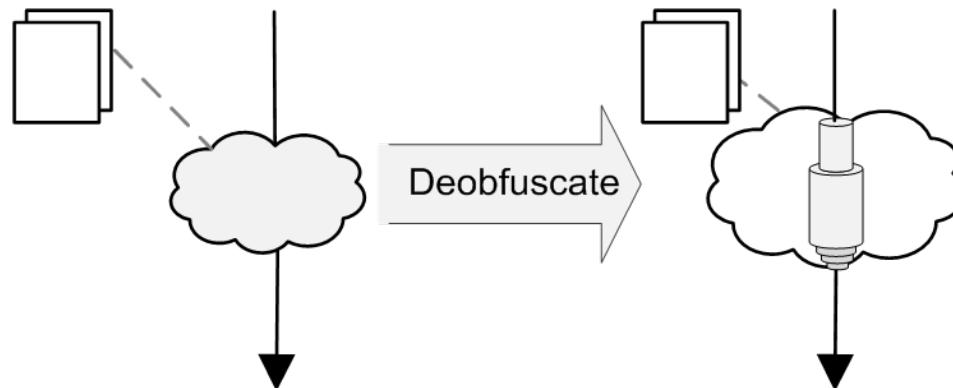
Plugging and Unplugging

- Existing control links can be broken, i.e., unplugged
 - When a fragment exit is plugged into a fragment entry, this equals a regular control link
 - When plugging a fragment exit into an entry, the helper constructs are then no longer needed and can be removed
- A new control link is inserted from the source of the fragment exit to the target of the fragment entry



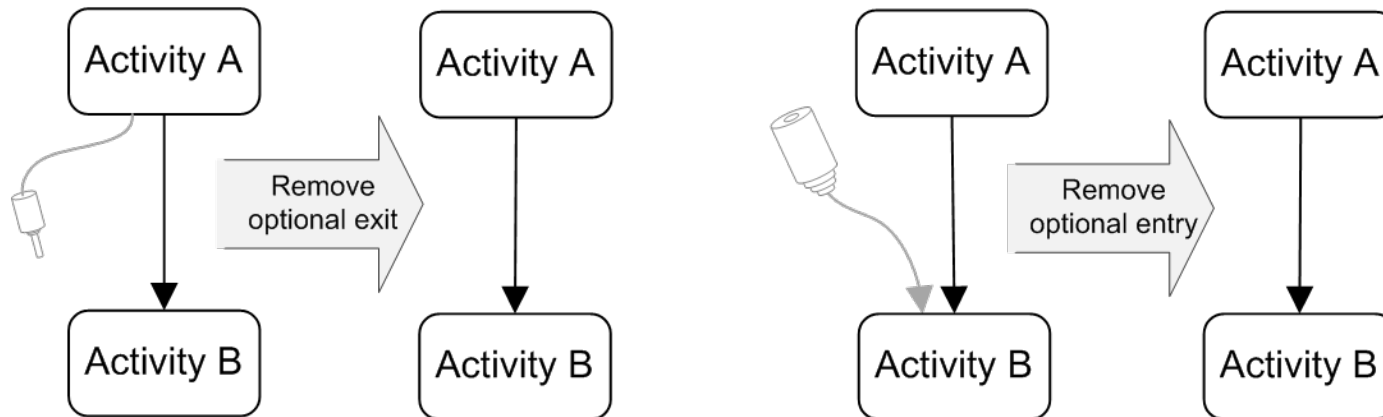
Usage of Regions

- Regions can be filled with life, i.e. with process structures
- A region can be split into a fragment entry and a fragment exit



Removing Unused Optional Entries and Exits

- Fragment entries and fragment exits which are optional and not used during composition can be removed



Completing the Integration: Merging the Contexts

- Matching and merging of the fragment artifacts with artifacts which already exist in the process context
 - Fragment artifacts: variables, imports, partnerLinks etc.
 - Matching based on the name, identifier or type
 - Possibly required: adjustment of data types
- Merging of handlers
 - Fragment handlers: fault, compensation, termination, events
 - Wiring with the handlers located in the ancestor scope
- Merging fragment metadata (WSDL, policies)
- Ongoing work: Design of an easy-to-use wizard that assists the user in wiring and merging of the contexts

Conclusion and Outlook

Limitations of the Approach to Manage Compliance

- Compliance fragments can address requirements that concern the activities *within* a workflow
- A compliance fragment cannot ensure that people who are involved in the process will behave compliant
- A compliance fragment cannot control the applications which are external to the workflow engine
- Further concepts (technical *and* non-technical) are needed for an overall approach to compliant business process automation and to a compliant business

Conclusion and Outlook

- Compliance is an important topic in research and industry
- Compliance fragments enable consistent augmentation of automated business processes with compliance
- Ongoing work regarding compliance fragments
 - Development of a wizard for integration of compliance fragments into processes
 - Managing change of compliance requirements
- Final results of the COMPAS research project
 - Available at the beginning of 2011
 - Downloads at www.compas-ict.eu
- Final results of the MASTER research project
 - Available at the beginning of 2011
 - Downloads at www.master-fp7.eu

End of Document