

CD1 Figure1

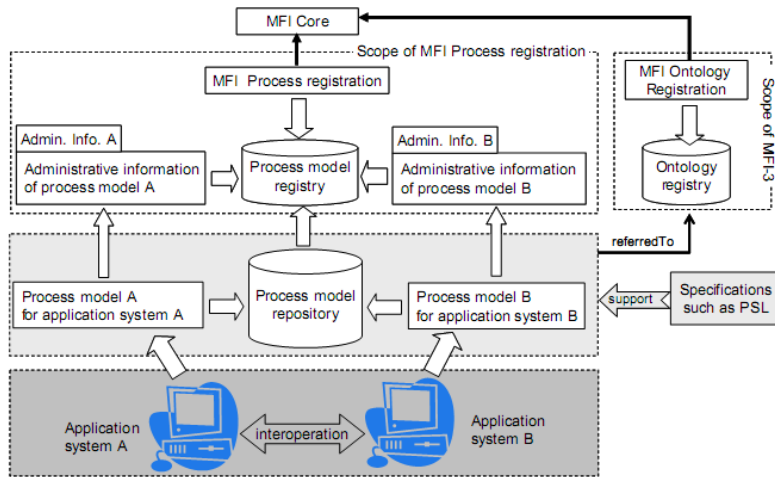
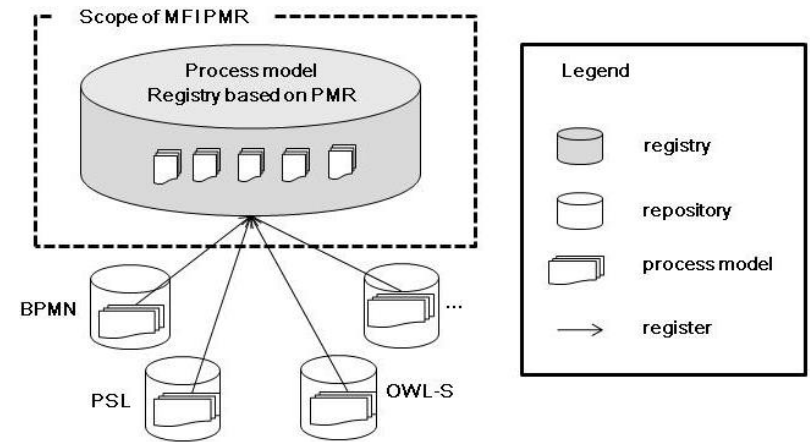


Figure 1 – Scope of MFI Process registration

Fig1(CD1)-Fig1(CD2)

CD2 Figure 1 – The scope of MFI PMR



<p>Scope</p>	<p>Made corresponding changes to the scope statement.</p> <p>The scope of CD2 means that any process that is modeled with a specified language can be registered into a process registry based on part 5.</p> <p>In addition, The “practical value added” by applying “process on physical things” and “process by human” may not be great.</p>	<p>AU04</p> <p>AU05</p> <p>CA06</p> <p>JP002</p> <p>JP003</p> <p>GB05</p>
<p>Figure1</p>	<p>Figure 1 in CD1 has been totally changed.</p> <p>Figure 1 in CD2 shows the scope of part 5. And the corresponding legend is provided.</p> <p>In Fig 1 in CD2, the reference to MFI-3 is deleted.</p> <p>The relationship between part5 and MFI-3 is explained in Clause 5.2 and Figure2.</p>	<p>CA08</p> <p>CA10</p> <p>CA29</p> <p>CA30</p> <p>US02</p>

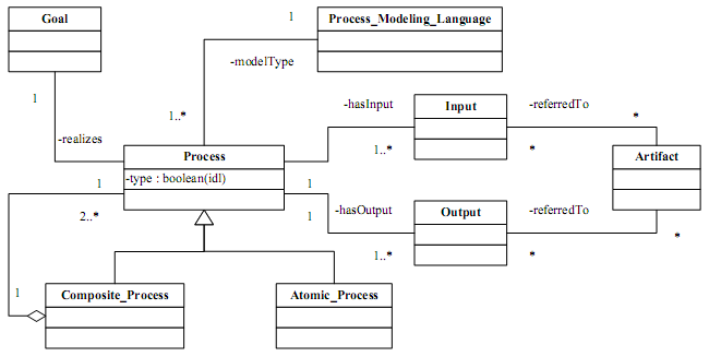


Figure 2 – Base Model of MFI Process registration

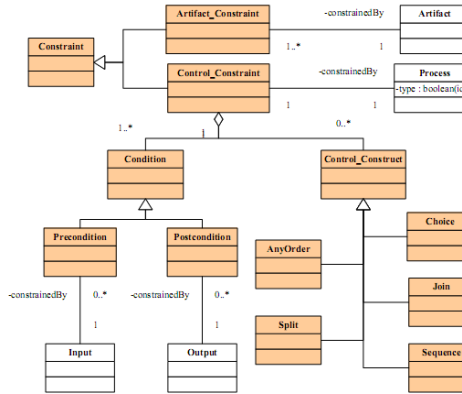


Figure 3 – Process Control Model of MFI Process registration

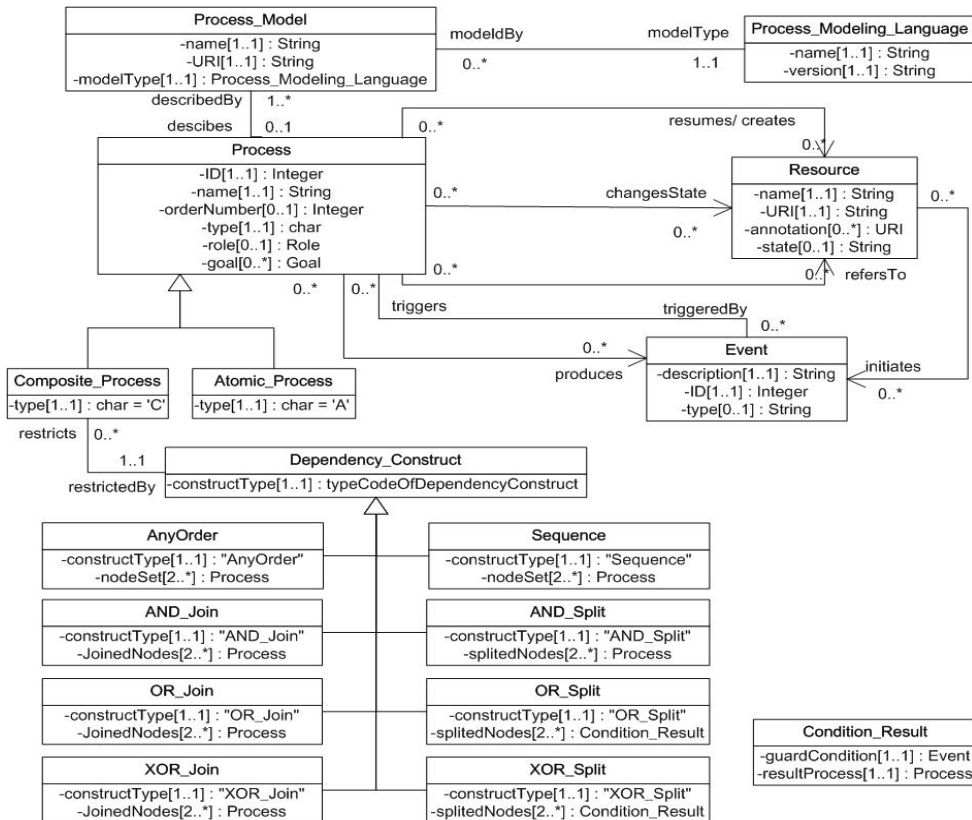


Figure 2 – The metamodel for process model registration

Fig2-Fig3 ->Fig2	In CD1, Base Model and Process Control Model were defined to record basic structural and constraints of processes. But in CD2, there is only one metamodel by merging key metaclasses from CD1.	AU06
	Figure 2 in CD2 provides the metamodel for process model registration	CA10 CA22 GB13 GB14 US04

The relations between MFI-5 and some other parts in MFI

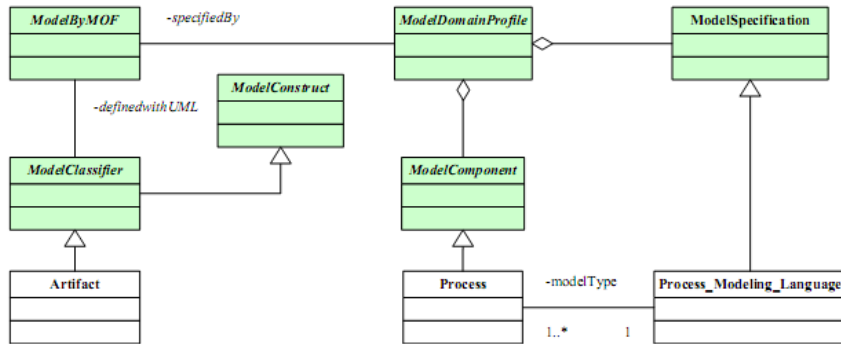


Figure 4 – Relationship between MFI Core and MFI Process registration

Fig in CD1

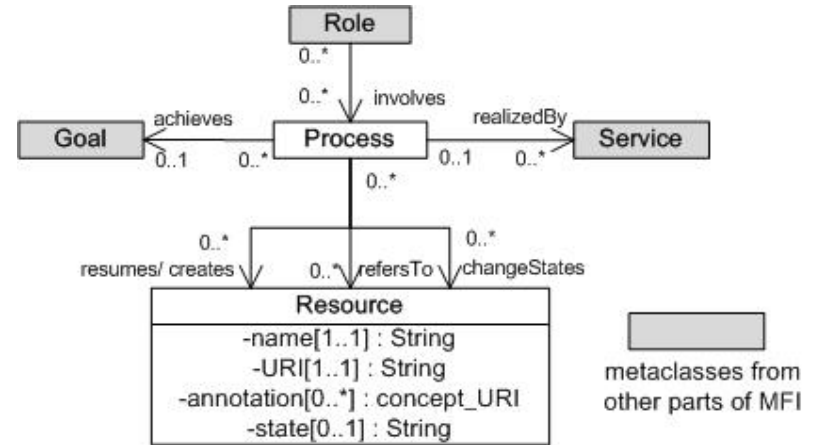


Fig in CD2

<p>Figure4(Figure3& Relationships between MFI-5 and other parts)</p>	<p>Figure 4 in CD1 shows the relationship between MFI Core and MFI Process registration, which has been totally changed. In CD2, the relationship between MFI-5 and MFI Core should be recalled after we get consensus on MFI Core. Figure 3 in CD2 shows the relationship between MFI PMR and other parts in MFI. You can see more details in clause 5.2.</p>	<p>AU04 AU09 AU14 CA07 CA35 JP003 GB16</p>
--	--	--

CD1->CD2		WHY&HOW	comments	
delete	Goal	<p>Since MFI-8 focuses on Role&Goal registration, CD2 removes the definition and text of "Goal".</p> <p>But in 5.2, the relationship between Process and Goal is addressed. A process can achieve zero to one instance of Goal.</p>	<p>AU06</p> <p>CA27 CA41</p> <p>GB14</p>	
	Input/Output	<p>In CD2, "Input" and "Output" are also removed. Instead, we add three relationships named "resumes" "creates" and "refersTo" from Process to Resource. The resource a process consumes or refersTo can be considered as an input, and that a process creates can be considered as an output.</p>	<p>AU09-AU12 JP008</p> <p>GB14 GB30</p>	
	Constraint	Artifact_Constraint	<p>The "Artifact_Constraint" is removed in CD2. But the role it plays in CD1 is addressed by the attribute "annotationType" of Resource, which can be declared as the URI of the registered Ontology_Atomic_Construct based on MFI Ontology Registration. It means that the concepts in ontology can be used to annotate resources participating in a process, so the relationships between ontology concepts imply constraints between resources.</p>	<p>CA31 CA44</p> <p>JP014</p> <p>GB18</p>
		Control_Constraint	<p>The "Control_Constraint" is removed in CD2. But the role it plays in CD1 is addressed by the association between process and dependency construct to handle a control with the order of processes in which the connected processes are executed.</p>	<p>CA44</p> <p>GB18 GB33</p> <p>US10</p>
	Condition	Precondition	<p>"Precondition" and "Postcondition" are removed.</p>	<p>AU09-AU12</p>
Postcondition		<p>In CD2, the precondition that an input has to satisfy can be described as the specific state that the resource resumed by a process should stay. Besides, "Event" is added to record the event that triggers a process. We add a relationship named "produces" from Process to Event and another relationship named "triggers" from Event to Process.</p> <p>The event that triggers a process means the condition that must always be true prior to the execution of a process in a formal specification. While the event produced by a process means the condition that must always be true just after the execution of a process in a formal specification.</p> <p>Event is added to support the registration of event-driven process model.</p>	<p>JP009</p> <p>JP14-JP19</p>	
add	Event	<p>The event that triggers a process means the condition that must always be true prior to the execution of a process in a formal specification. While the event produced by a process means the condition that must always be true just after the execution of a process in a formal specification.</p> <p>Event is added to support the registration of event-driven process model.</p>	<p>JP001</p> <p>GB19</p>	
	Conditional_Process	<p>Conditional_Process is a metaclass designating a splitted node of "OR_Split" or "XOR_Split" which contains a guard condition and the according process to be executed if the guard condition is satisfied.</p>	<p>CA45</p> <p>GB21</p>	
	Process_Model	<p>In 4.1 of CD2, process and process model are separated.</p> <p>Process is defined as a set of activities and resources, organized by a dependency construct, which all participate in fulfilling a given purpose.</p> <p>Process model is defined as a specification that is the result of modelling one or more processes, adopting a specific process modelling language to describe features of a process. It shows what the process does and how it is done.</p> <p>So CD2 adds a new metaclass named "Process_Model", connecting "Process" to "Process_Modeling_Language".</p>	<p>AU07</p> <p>CA14</p> <p>CA24</p> <p>CA25</p> <p>JP005</p> <p>GB27</p>	

		<p>Process_Model is a specification, the result of modelling processes. Process_Modeling_Language specifies the modeling language that Process_Model uses to represent processes.</p> <p>CD2 assumes the process model to be registered has been created and stored in a specific process model repository.</p>	<p>US05</p> <p>US06</p>
revise	Artifact(CD1)->Resource(CD2)	<p>The term 'Artifact' has connotations of something that is man-made. A more general term would be 'Resource'. So "Artifact/artefact" is replaced by "Resource" throughout the document.</p> <p>And in order to execute a state transition, a resource needs an attribute named state to specify the current state of resource. So In 5.3.6 of CD2, attribute "state" is added into Resource. Besides, we provide a relationship "changesState" from Process to Resource to describe the change of state.</p>	<p>CA23 CA33</p> <p>CA42 CA43</p> <p>JP001</p> <p>GB02</p> <p>US02</p>
	Composite_Process	<p>In CD2, the definition of Composite_Process is provided. Clause 4.1.6 defines composite process as "process that consists of other processes and is organized by only one type of dependency construct."</p> <p>And the aggregation relation (consists of) between Process and Composite_Process was removed.</p>	<p>CA15</p> <p>US07</p>
	Control_Construct (Dependency_Construct)	<p>"Control_Construct" in CD1 is changed into "Dependency_Construct", in which there are 8 kinds of constructs, such as "AnyOrder", "Sequence", "AND_Split", "AND_Join", "OR_Split", "OR_Join", "XOR_split" and "XOR_Join".</p> <p>Dependency_Construct is used to handle a control with the order of processes in which the connected processes are executed.</p>	<p>AU09 AU15</p> <p>CA45</p> <p>JP020 GB21</p> <p>US07</p> <p>US09 US10</p>
	Introduction	<p>The first paragraph in CD1 has been deleted.</p> <p>Introduction in CD2 is rewritten, see document of the draft text of CD2 19763-5.</p> <p>In CD2, References to "workflow" are removed.</p>	<p>CA02-</p> <p>CA05</p> <p>GB04</p>

1. We combine the event-driven and state-transition in a uniformed way to handle a control by Dependency_Construct.
 - For example, suppose Process 1 has a post condition named "condition1", and condition1 is also the precondition of Process2. In CD2, we can define an event whose description is "condition1", and then Process 1 produces it, which also triggers Process2.
 - The event-driven languages can be registered as the BPMN example in Annex A.
 - As to state transition languages, CD2 introduces a "state" attribute in "Resource". And we add another association named "changesState" from "Process" to "Resource" by means that the state of a resource may get changed in a process. If a resource staying in a state would result in the execution of one or more processes, we say the specific resource initiates an event that can trigger those processes. So process models that are described in a state-transition can also be registered.

QUESTION:

1. In CD2, Conditional_Process is a kind of data type designating a split node of "OR_Split" or "XOR_Split" which contains a guard condition and the according process to be executed if the guard condition is satisfied. "OR_Split" or "XOR_Split" can have two or more Conditional_Processes as split nodes.
So, there is a question: Do we need to add Conditional_Process into clause 5 as a metaclass.
2. We are thinking whether it is acceptable to merge "AND_Split" and "AND_Join" into one construct named "AND", merge "OR_Split" and "OR_Join" into "OR", and merge "XOR_split" and "XOR_Join" into "XOR" because split and join are always appear in pair. If they are not merged, the split nodes will appear twice so that it is not suitable to register resources as inputs or outputs.
What do you think?