

OPM – PML2 Mapping Issues

James Michaelis

5 / 22 / 2009

RPI/TWC

Provenance Example

- Necessary to illustrate OPM-PML2 mapping issues
- Based off OPM Primer (v. 1.01, at openprovenance.org) example on page 9, Figure 2:
 - John bakes a cake, using a bowl, flour and eggs.

Object Mappings

OPM	PML	Mapping	Example
Artifact	pmlp:Information	Intuitive	Bowl
Process	execution of InfRule?	---	---
Function (the execution of a Process)	InfStep + InfRule	Not necessarily intuitive	Baking the cake
Agent	pmlp:InferenceEngine	Intuitive	John

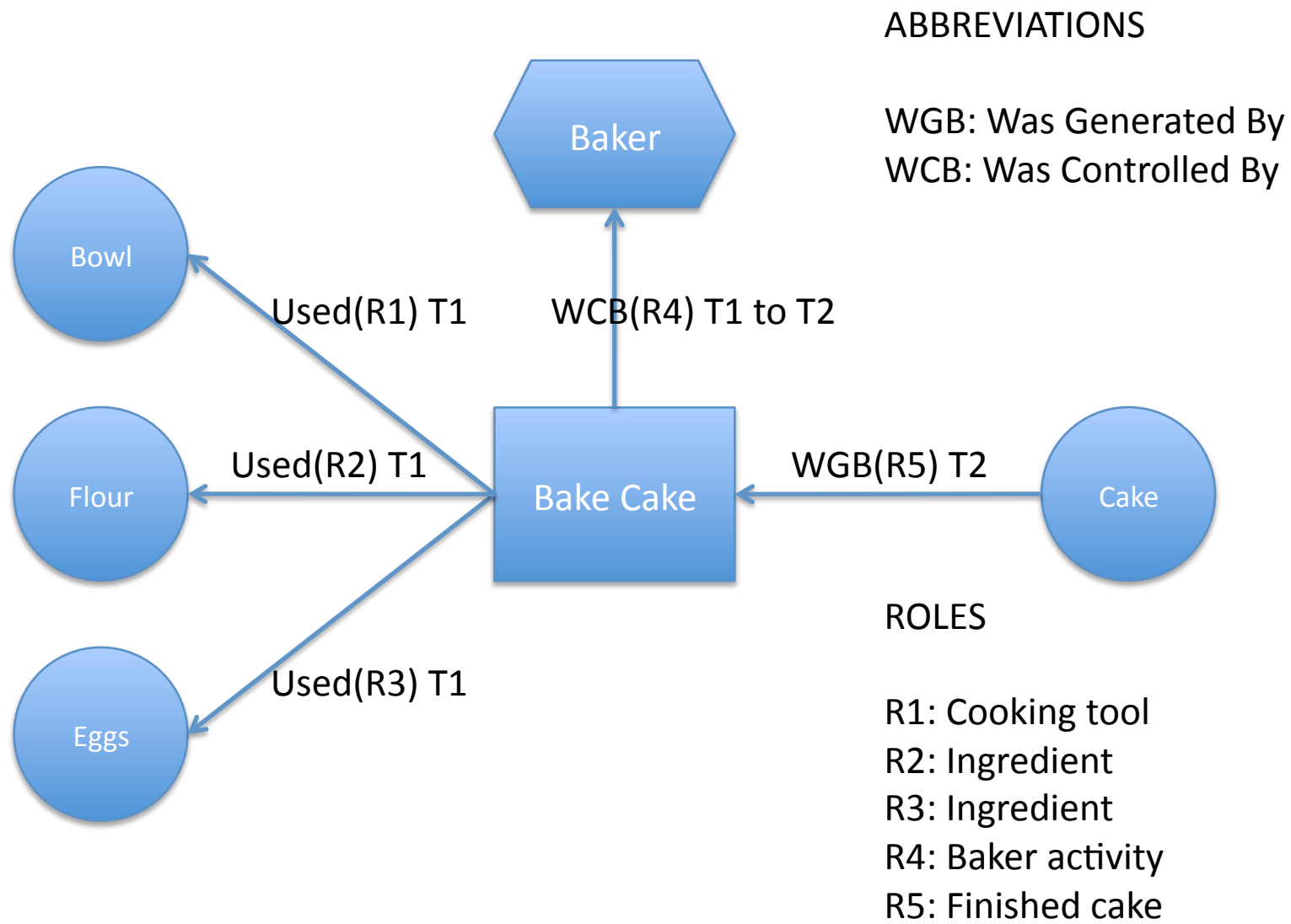
Relation Mappings

OPM	PML	Mapping	Example
Used	hasAntecedentList +	Intuitive	John used a bowl to produce the cake
WasGeneratedBy	isConsequentOf -	Intuitive	Cake was generated by baking done by John
WasControlledBy	hasInferenceEngine	Intuitive	John did the baking
WasDerivedFrom	?	NONE FOUND	---
WasTriggeredBy	?	NONE FOUND	---
Overlaps	?	NONE FOUND	---

Annotation Mappings

OPM	PML	Mapping	Example
OTime	pmlp:hasCreationDateTime + xmls:dateTime	Intuitive	John uses the bowl at time T1, and finishes the cake at T2
Role	?	NONE FOUND	---
Account	?	NONE FOUND	---

OPM Representation



PML2 Translation

ABBREVIATIONS

hCon:
pmlj:hasConclusion

iCof:
pmlj:isConsequentOf

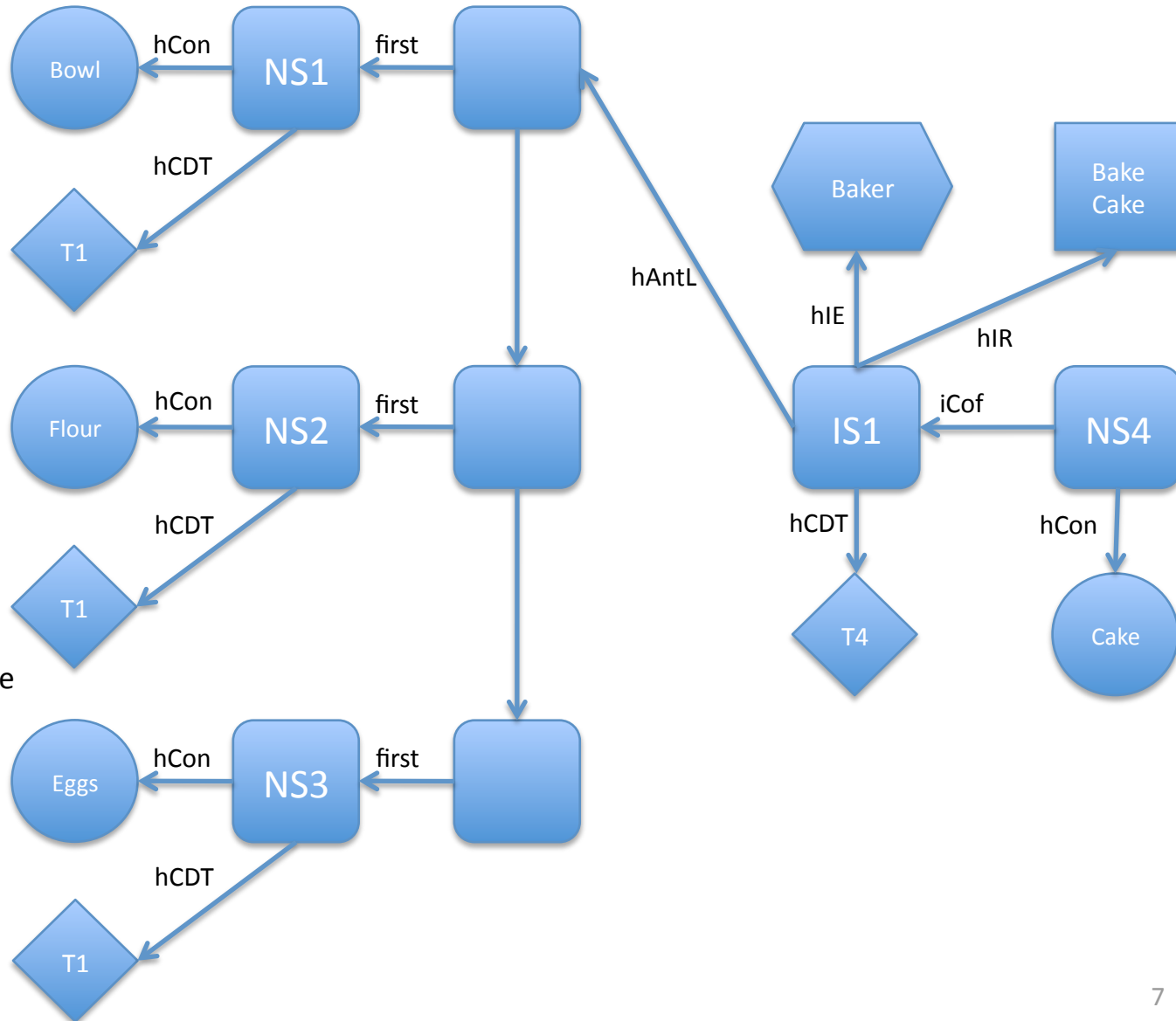
hIR:
pmlj:hasInferenceRule

hIE:
pmlj:hasInferenceEngine

hCDT:
pmlp:hasCreationDateTime

hAntL:
pmlj:hasAntecedentList

first:
ds:first



Our General Strategy

- Record provenance information in a format known as ProtoProv
- Designed for three purposes
 - Generation of OPM
 - RDF
 - PML

Our generation of OPM

- Handled via ontology for taking down basic annotations

Generation of PML

Appendix: OPM Model

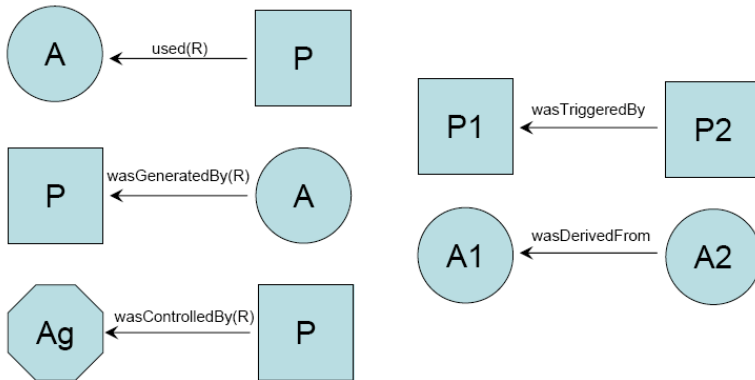


Figure 1: Edges in the Provenance Model

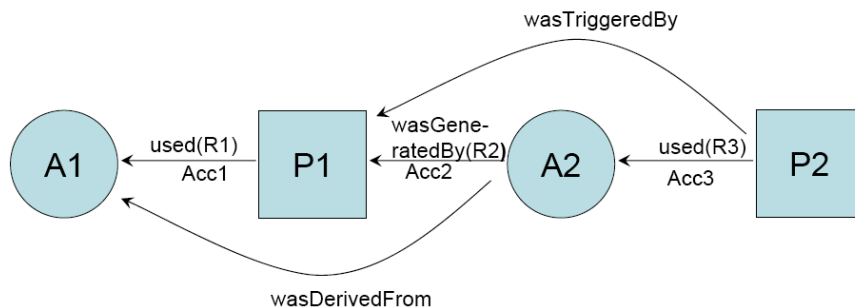


Figure 7: One Step Inference in the Provenance Model

- Three main concepts
 - Artifact
 - Process
 - Agent
- Five Main Relations
 - (Artifact) was generated by (Process)
 - (Process) used (Artifact)
 - (Process) was controlled by (Agent)
 - (Process) was triggered by (Process)
 - (Artifact) was derived from (Artifact)
- Three auxiliary concepts
 - Account (artifact, process, agent)
 - OTime (relation)
 - Role (relation)
- Computation
 - Transitive inference (WTB, WDF)
 - Temporal inference (OTime)

Appendix: OPM Syntax

<i>ProcessId</i>	: primitive set	(Process Identifiers)
<i>ArtifactId</i>	: primitive set	(Artifact Identifiers)
<i>AgentId</i>	: primitive set	(Agent Identifiers)
<i>Role</i>	: primitive set	(Roles)
<i>Account</i>	: primitive set	(Accounts)
<i>Value</i>	: application specific set	(Values)
<i>Time</i>	: primitive set	(Time)
<i>Process</i>	= $ProcessId \rightarrow Value \times \mathbb{P}(Account)$	
<i>Artifact</i>	= $ArtifactId \rightarrow Value \times \mathbb{P}(Account)$	
<i>Agent</i>	= $AgentId \rightarrow Value \times \mathbb{P}(Account)$	
<i>OTime</i>	= $Time \times Time$	(Observed Time)
<i>Used</i>	= $ProcessId \times Role \times ArtifactId \times \mathbb{P}(Account) \times OTime^0$	
<i>WasGeneratedBy</i>	= $ArtifactId \times Role \times ProcessId \times \mathbb{P}(Account) \times OTime^0$	
<i>WasTriggeredBy</i>	= $ProcessId \times ProcessId \times \mathbb{P}(Account) \times OTime^0$	
<i>WasDerivedFrom</i>	= $ArtifactId \times ArtifactId \times \mathbb{P}(Account) \times OTime^0$	
<i>WasControlledBy</i>	= $ProcessId \times Role \times AgentId \times \mathbb{P}(Account) \times OTime^0 \times OTime^0$	
<i>Overlaps</i>	= $Account \times Account$	
<i>Refines</i>	= $Account \times Account$	
<i>OPMGraph</i>	= $Artifact \times Process$ $\times Agent \times \mathbb{P}(Used)$ $\times \mathbb{P}(WasGeneratedBy) \times \mathbb{P}(WasTriggeredBy)$ $\times \mathbb{P}(WasDerivedFrom) \times \mathbb{P}(WasControlledBy)$ $\times \mathbb{P}(Overlaps) \times \mathbb{P}(Refines)$	