

Meta Model Based Architecture for Software Process Instantiation

Peter Killisperger

University of South Australia, University of Applied Sciences - München

Georg Peters

University of Applied Sciences – München

Markus Stumptner, Georg Grossmann

University of South Australia

Thomas Stückl

Siemens AG

ICSP 2009

Vancouver

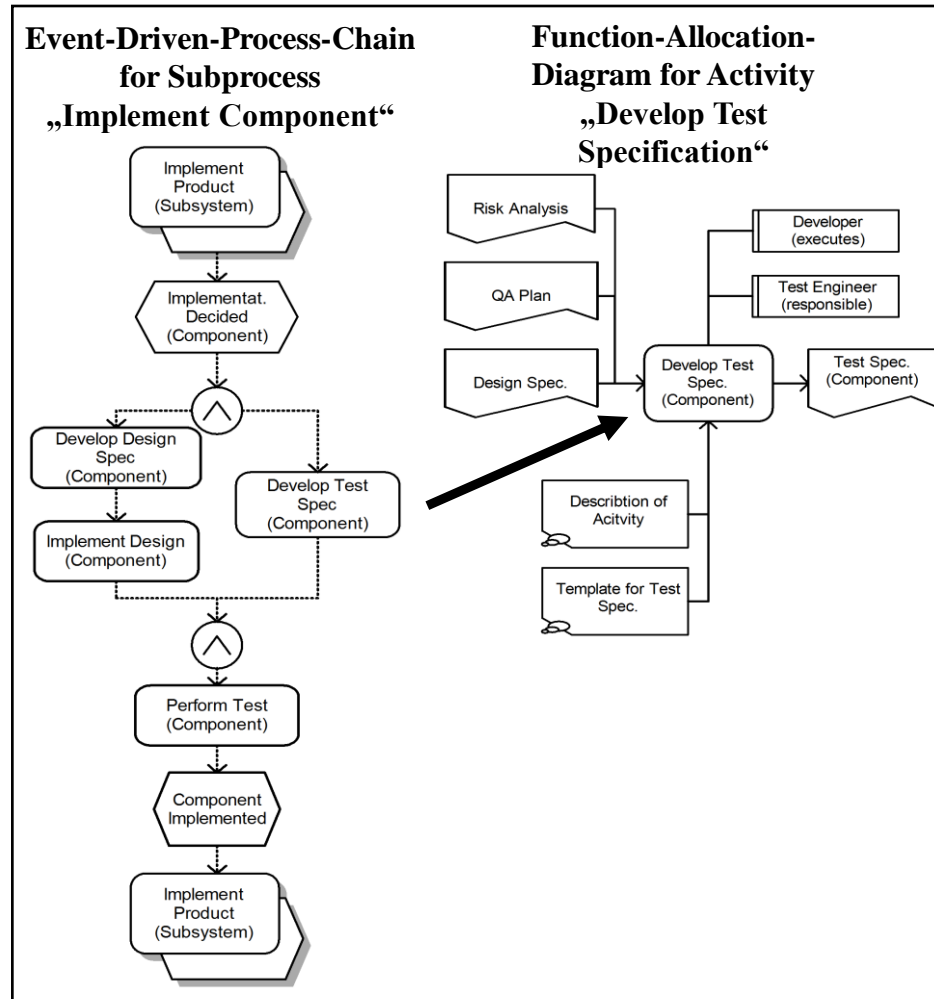
Agenda

1. Introduction to Siemens SW Processes and their Application
2. Improvement of Current Practice
3. Meta Model Based Architecture for Software Process Instantiation
4. Evaluation
5. Conclusion

Software Processes at Siemens

- Software developing business units within Siemens define their processes according to their **individual needs**.
- Because of their size and complexity, they are not defined for projects individually but in a generic way as reference process.
- Siemens' processes are **similar to workflows** but their focus is to be read and understood by humans and not to be automated by IT.
- Processes have to comply with **modeling restrictions** of the **Siemens Process Framework (SPF)**.
- They are modeled by using semi-formal **Event Driven Process Chains (EPC)** and **Function Allocation Diagrams (FAD)**.

Example: Siemens Process



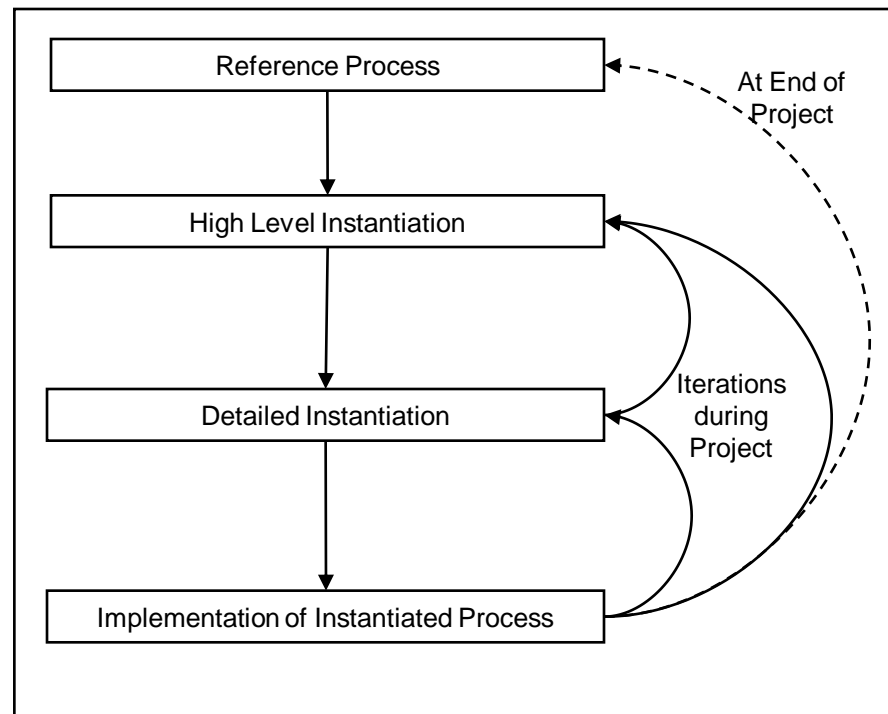
Current Instantiation and Application

- Processes are **reference processes**.
- They are used as **general guidelines** published on Siemens' intranet.
- They are instantiated minimally by manual creation of **project specific plans**.
- A more far reaching manual instantiation is **error-prone and time consuming**.
- **Tool support** is desirable in order to **reduce effort** and to **guarantee adherence of SPF modeling restrictions**.

Constituents of Instantiation

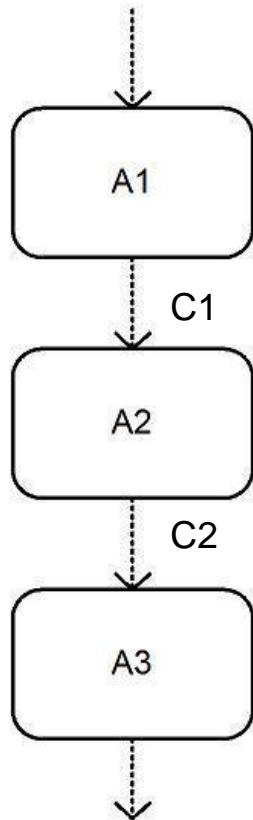
1. **Tailoring.** Tailoring is „the act of adjusting the definitions and/or of particularizing the terms of a general process description to derive a new process applicable to an alternative (and probably less general) environment” (Ginsberg, M. & Quinn, L., 1995).
2. **Resource Allocation.** The assignment of resources to activities to carry them out is called resource allocation (Aalst, W. & Hee, K., 2004). In most cases human participants are allocated.
3. **Instantiation of Artefacts.** The general artefacts have to be individualized for a project at hand and associated with files implementing the artefacts.

New Software Engineering Framework

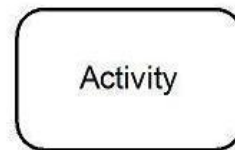


Example

Snapshot of Process before Instantiation



Snapshot of Method Manual



Context Activity

inv: self.incomingCF->size()==1

inv: self.outgoingCF->size()==1



Context ControlFlow

inv: self.source.oclIsKindOf(Activity)

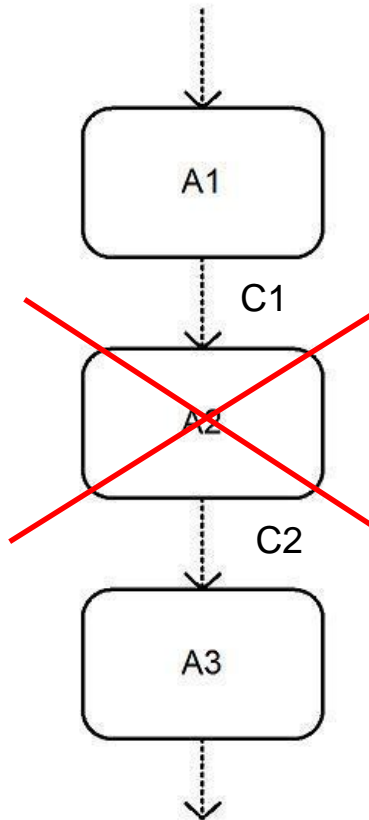
or self.source.oclIsKindOf(StartEvent)

inv: self.target.oclIsKindOf(Activity)

or self.target.oclIsKindOf(EndEvent)

Example

Snapshot of Process after Instantiation



Violated Constraints of Method Manual

C1

inv: self.target.oclIsKindOf(Activity)
or self.target.oclIsKindOf(EndEvent)

C2

inv: self.source.oclIsKindOf(Activity)
or self.source.oclIsKindOf(StartEvent)

Research Issue

Implement Basic Instantiation Operations in order to support project managers / owners in instantiating Processes by executing their decisions in the process by maintaining correctness of the process.

- A process is correct when it complies with the restrictions on the process defined in a method manual.
- A method manual is a meta model defining permitted entities and constructs.
- It is derived from restrictions of the used process definition language and organizational restrictions.

Meta Model Based Architecture

We developed a framework that executes instantiation decisions made by humans on a (correct) process and guarantees correctness of the resulting (instantiated) process.

- The framework must consider the environment in which a change is performed.
- For every change of a specific process step instance “s” the context of that change is the set of instances and relationships connected to “s”.
- The abstract description of these entities is what we call scope description of an operation.

Meta Model Based Architecture (cont.)

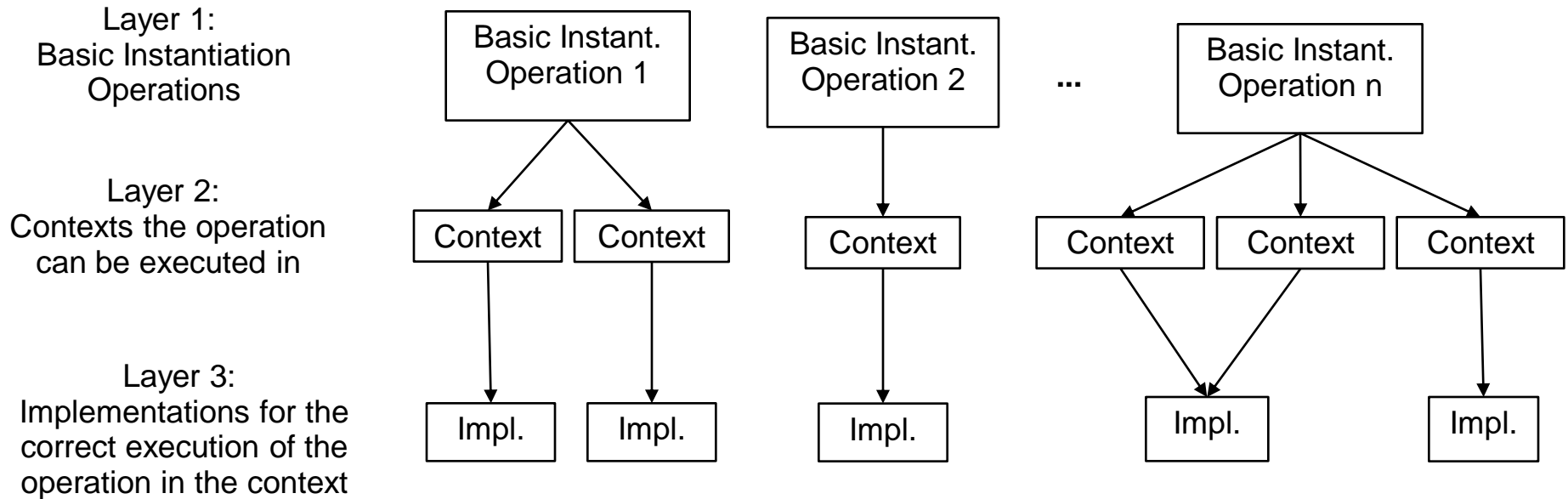
Scope descriptions have to be defined by human experts:

1. Identify restrictions on **classes of entities** of the method manual **directly affected** by the Basic Instantiation Operation.
2. Identify **classes of entities** allowed in the process **having a relationship with the classes of entities directly affected** by the Basic Instantiation Operation.
3. Identify restrictions on **related classes of entities** which might be breached by execution of the Basic Instantiation Operation.

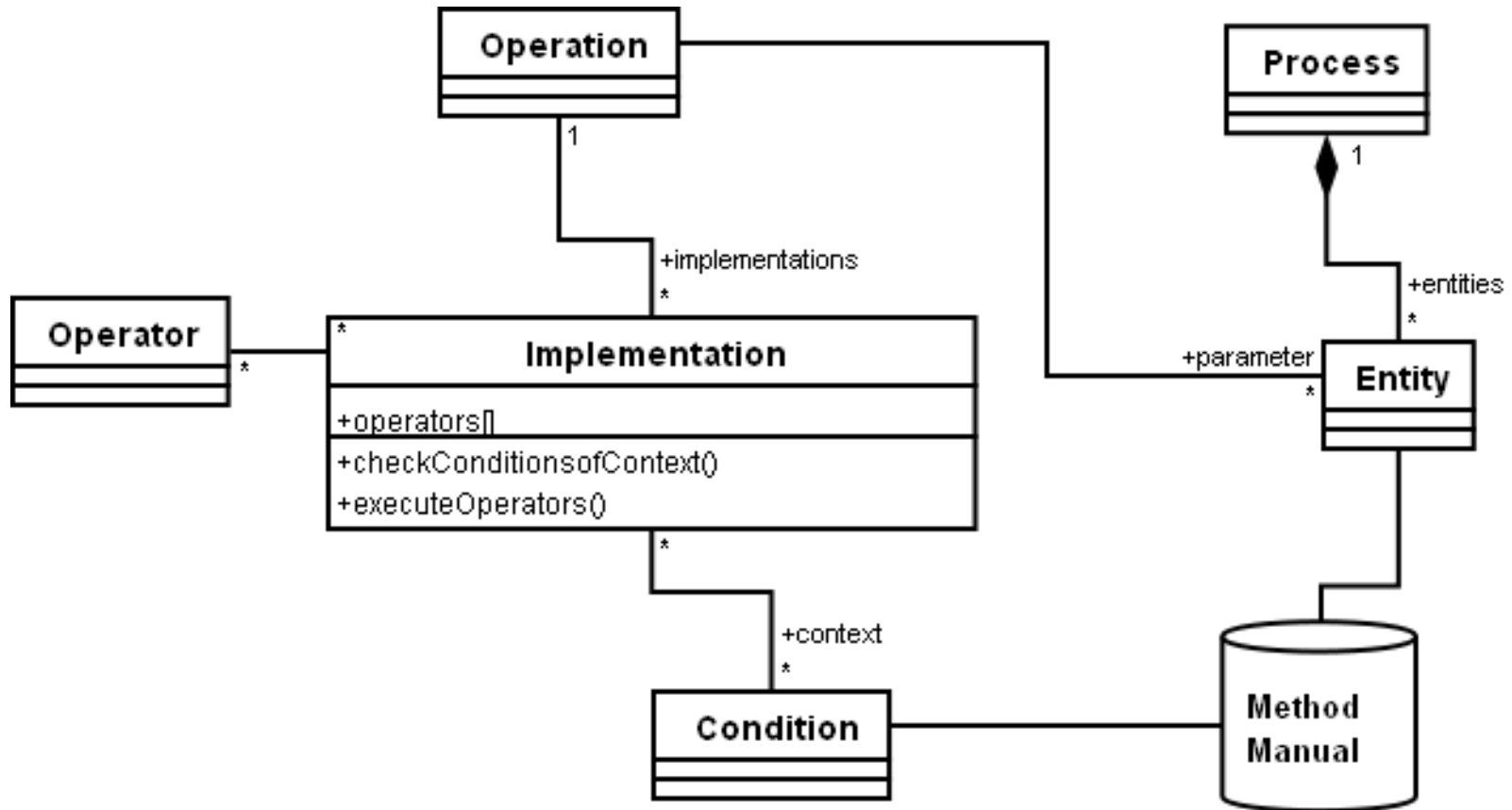
Meta Model Based Architecture (cont.)

- From the scope description all **contexts can be generated** in which an operation can be executed in.
- Every individual context has to be **associated with a particular implementation** which executes the Instantiation Operation in this explicit context.
- **Not every context requires** an individual implementation.

Meta Model Based Architecture (cont.)



Meta Model Based Architecture (cont.)



Evaluation / Prototype

- The **architecture has been implemented** for a software process of a Siemens' business unit comprising about **3000 entities**.
- Two exemplary Basic Instantiation Operations „**Inserting an Activity**“ and „**Deleting an Activity**“ have been chosen for implementation.
- **Description of Scope** has been defined and **Contexts** generated.
- **Implementations** for Contexts have been developed iteratively.

Conclusion

- We developed a **flexible architecture** for instantiation of software processes.
- We defined a **procedure** to implement the architecture for a particular organization.
- We **applied it on a reference process** of a particular business unit of Siemens.

Future Work:

- Increase **independence** from domain experts when implementing the architecture.
- Integrate ability to **handle change** in method manual.

Method Manual Based Process Generation and Validation

Questions ?