

EuroPLoP 2008

13th European Conference on Pattern Languages of Programs

CEP Focus Group

Dr. Adrian Paschke

RuleML Inc. and Biotec Center Dresden

Adrian.Paschke@gmx.de

Dr. Rainer von Ammon

CITT GmbH Regensburg



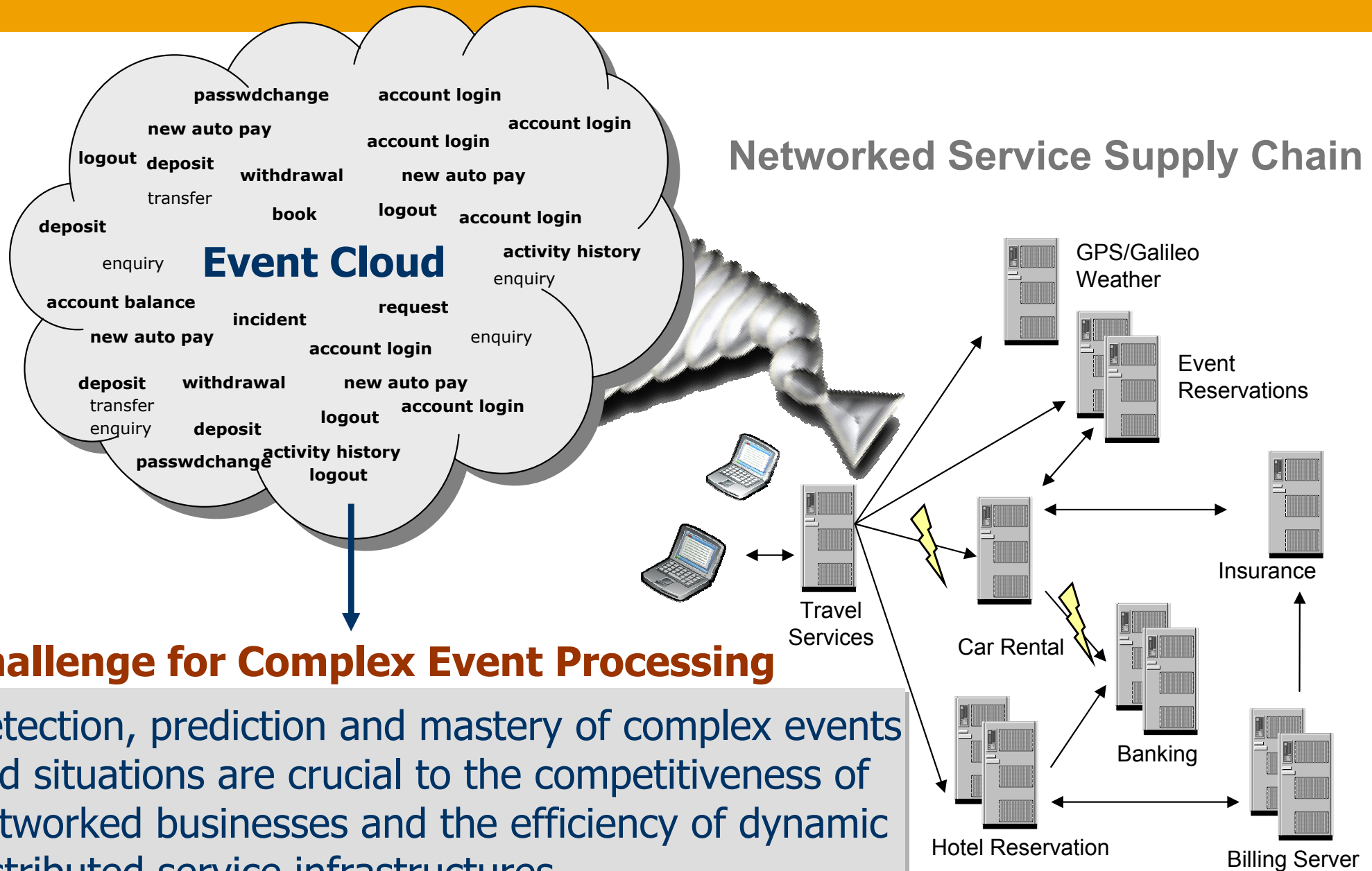
Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- Modeling and Pattern Languages for CEP
- Brainstorming on automation and support of CEP design
- Summary of achieved results, ideas and next steps

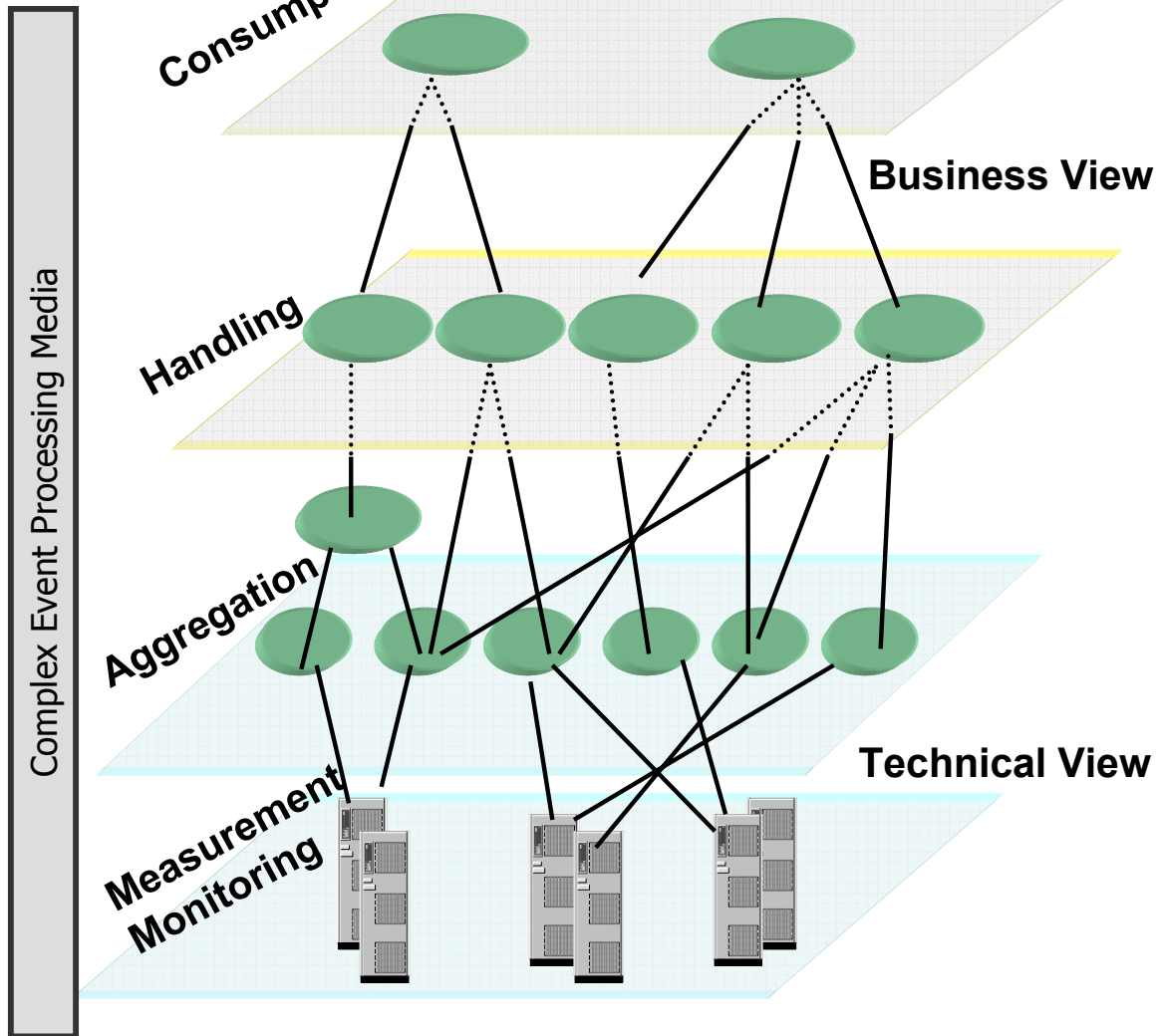
Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- Modeling and Pattern Languages for CEP
- Brainstorming on automation and support of CEP design
- Summary of achieved results, ideas and next steps

Complex Event Processing - Why do we need?



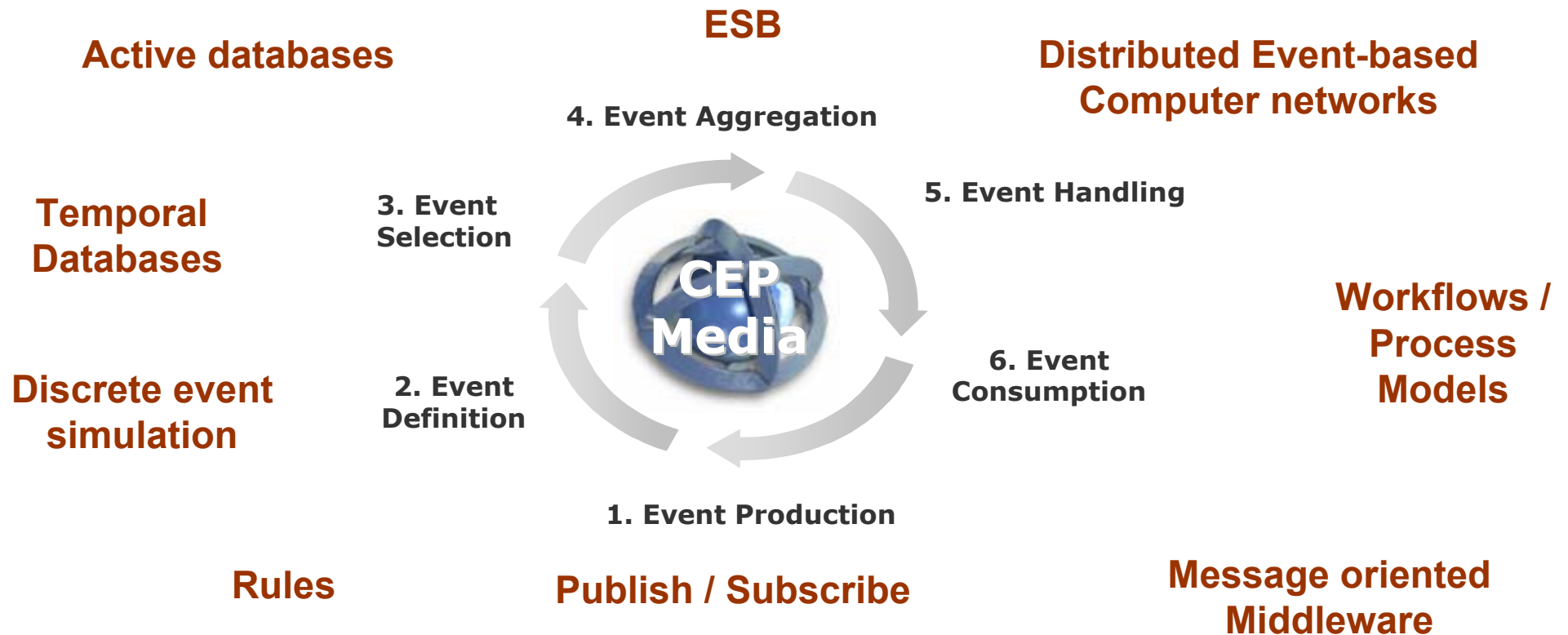
Complex Event Processing – What is it?



- Complex Event Processing (CEP) is a discipline that deals with event-driven behavior
- Selection, aggregation, and event abstraction for generating higher level complex events of interest



Problem Domain



- Currently, “CEP” covers a huge, unstructured field of application domains, technologies and products
- Potential adopters have major problems in understanding and adequately designing and implementing successful CEP solutions
- **CEP engineering remains a laborious trial and error process with slow development and change cycles.**

Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- Modeling and Pattern Languages for CEP
- Brainstorming on automation and support of CEP design
- Summary of achieved results, ideas and next steps

Solution Approach – CEP (Design) Patterns

“A design pattern addresses a recurring design problem that arises in specific design situations and presents a solution to it”

(Buschmann, *et. al.* 1996)

„A pattern is the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts.“

(D. Riehle and H. Zullighoven: *“Understanding and Using Patterns in Software Development”*)

- A design pattern describes a proposed solution of a permanent recurring software design problem.
- Design patterns have become a wide-spread means to transfer knowledge about successful designs.
- They are more or less formalized descriptions of solutions to certain problem classes, and are well suited to establish a body of knowledge for a particular application domain

Fundamental Terminologies - Modeling / Design Perspective on CEP

■ *CEP Model*

A CEP model is a *representation of* a CEP system, whereas systems can be physically observable elements or more abstract concepts like CEP modeling languages.

■ *CEP Reference Model*

A CEP reference model is an abstract representation of the entities and relationships involved in a problem space. It forms the conceptual basis for the development of more concrete CEP models of the space, and ultimately CEP implementations, in a concrete application/computing context by customizing the CEP reference model to a particular usage context.

■ *CEP Patterns and CEP Pattern Languages*

CEP Patterns capture and formally codify good designs and best experience-based best practices in a CEP pattern language based on a common vocabulary in such a way that it is possible for others to reuse them. They successfully convey insight into common problems and their solutions.

Fundamental Terminologies - Processing Perspective on CEP

■ *Event Pattern*

An event pattern definition (event definition or event type) describes the structure of an (atomic or complex) event, i.e. it describes its internal structure and detection condition(s).

■ *Event Instance*

A concrete instantiation of an event pattern is a specific event instance (also event object).

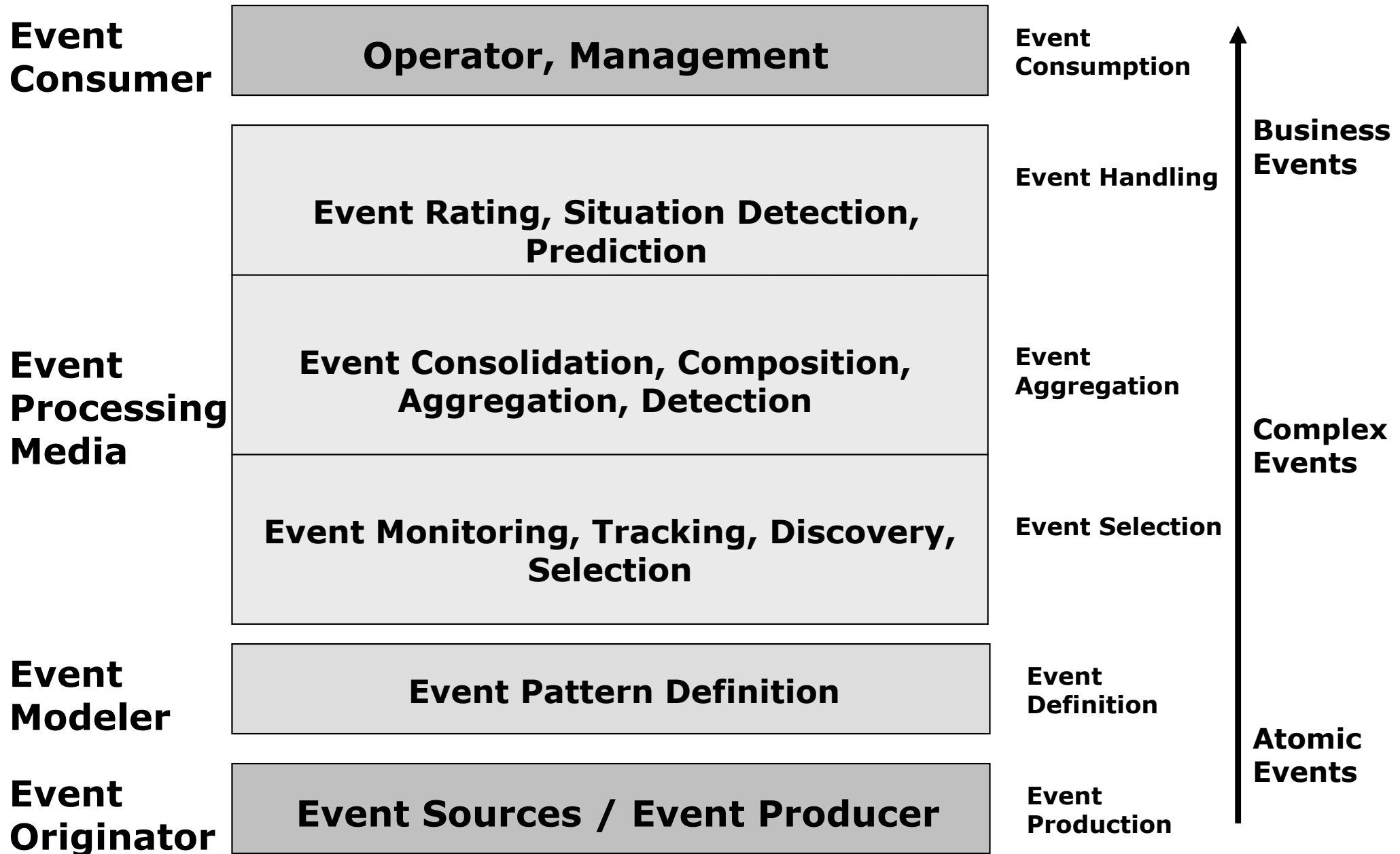
■ *Complex Event Processing and Event Processing Languages*

Complex event processing describes the process of event selection, aggregation, hierarching, event abstracting and composing of complex events from raw events for generating higher level events of interest. Event Processing Languages support the specification of event patterns / event definitions, selection and consumption policies, as well as the rules for event processing.

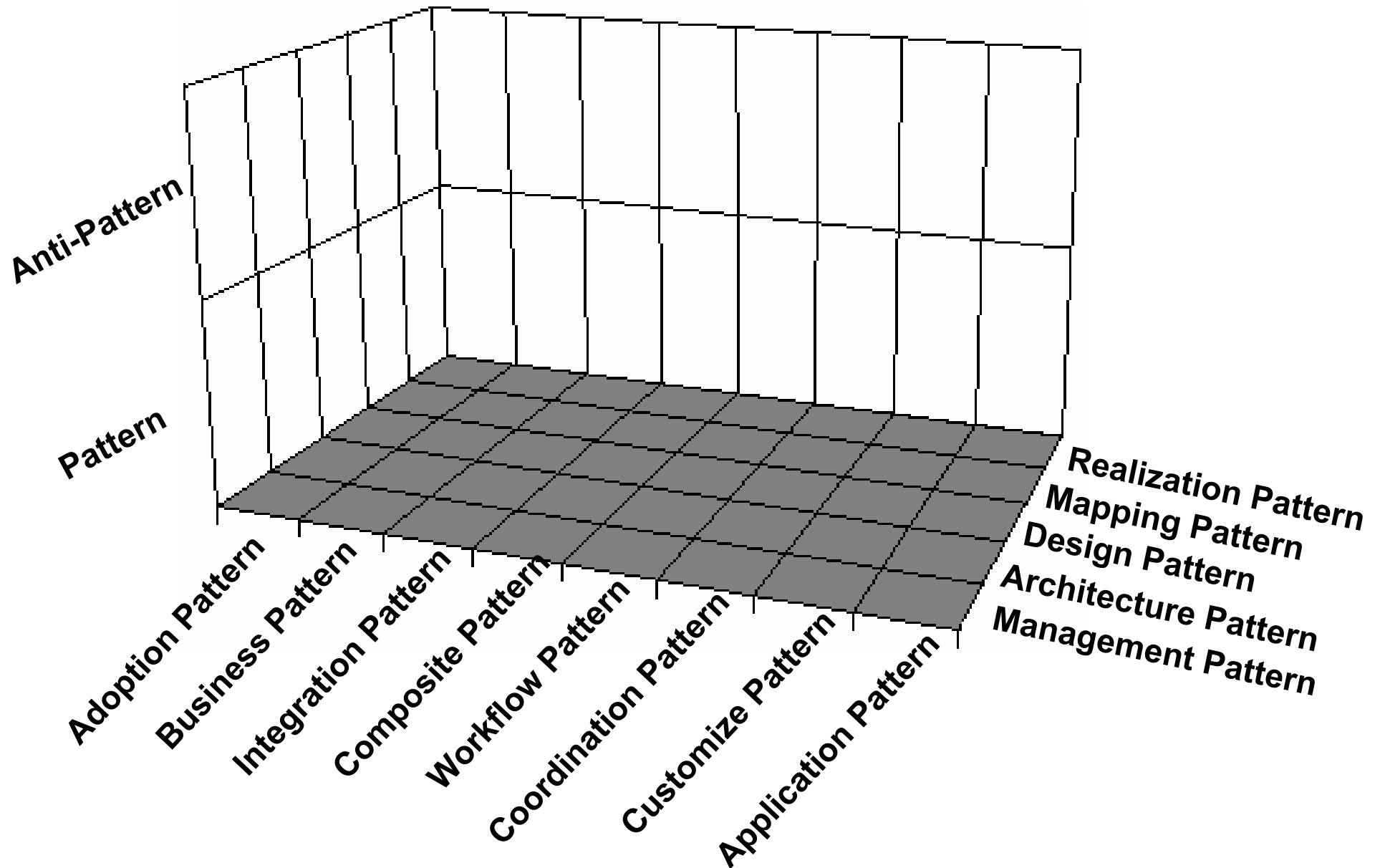
Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- Modeling and Pattern Languages for CEP
- Brainstorming on automation and support of CEP design
- Summary of achieved results, ideas and next steps

General CEP Reference Architecture



A Multi-Dimensional Categorization Scheme



Categorization of Patterns

■ Categorization according to Good and Bad Solutions

- *CEP Patterns*
- *CEP Anti-Patterns*

■ Categorization according to the Abstraction Level

- *Guidelines and Best Practices*
- *Management patterns*
- *Architecture patterns*
- *Design patterns*
- *Mapping patterns*
- *Idioms / Realization patterns*
- *Smells / Refactoring patterns*

■ Categorization according to the Intended Goal

- *Adoption patterns*
- *Business patterns*
- *Integration patterns*
- *Composite patterns:*
- ...

■ Categorization according to the Management Level

- *Strategic patterns*
- *Tactical patterns*
- *Operational patterns*

Example Refactoring Pattern

Name	Exception to the Complex Event
Description	A complex event is not detected in a particular situation. This situation can be described by \neg EXEC
Pattern definition before refactoring	... (A, B, C) ...
Pattern definition after refactoring	... \neg EXEC (A, B, C) ...
Addresses (Smell):	Inconsistency

Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- **Modeling and Pattern Languages for CEP**
- Brainstorming on automation and support of CEP design
- Summary of achieved results, ideas and next steps

Pattern Languages

Pattern Template

Name

A name used for identification

Problem

A repeating problem that occurs in a domain

Solution

Best practice solution to that problem

Consequences

Advantages and disadvantages of the recommended solution

Examples

A few examples where the recommended solution has already been applied

Anti-Pattern Template

Name

A succinct name to convey the essence of the anti-pattern

Problem / Bad solution

The commonly occurring mistake or bad solution that relates to the anti-pattern

Symptoms

The indications or signs of the problem

Consequences

The results of applying this anti-pattern

Root cause

This provides the context for the anti-pattern, that is, where a pattern was applied incorrectly and resulting in a problem or failed solution

Suggested solution(s)

Refactored solution that solves the problem and ensures more benefits

Design Pattern Template

Name	Adapted from Gamma, E., et al., <i>Design Patterns - Elements of Reusable Software</i> . 1995: Addison-Wesley.
Intent	Describes the underlying coordination problem and the basic principle of the design solution.
Also Known As	Other names of the pattern, if existing.
Motivation	Depicts a scenario of a concrete design problem and gives a solution.
Applicability	Itemizes the situations or conditions, in which the pattern can be applied. <ul style="list-style-type: none">- Participants characteristics- Resources- Other environmental factors
Design Structure	Describes the design structure.
Participants	Lists the classes and objects, which are described by the design pattern.
Consequences	Discusses the advantages and disadvantages of the pattern, and makes suggestions about variations. <ul style="list-style-type: none">- Solution goals- Computational and Resource properties
Variants	Brief description of variants, extensions or specializations of the design pattern.
Known Uses	Examples of applications, in which the design pattern is used.
Related Patterns	Lists patterns, which describe a similar task, and shows the relations to other patterns.

General Information

Detailed Description

Discussion

Additional Information

Refactoring Pattern

Name

Name of refactoring

Addresses

Problem addressed

Description

General description

Code base before refactoring

Example code base

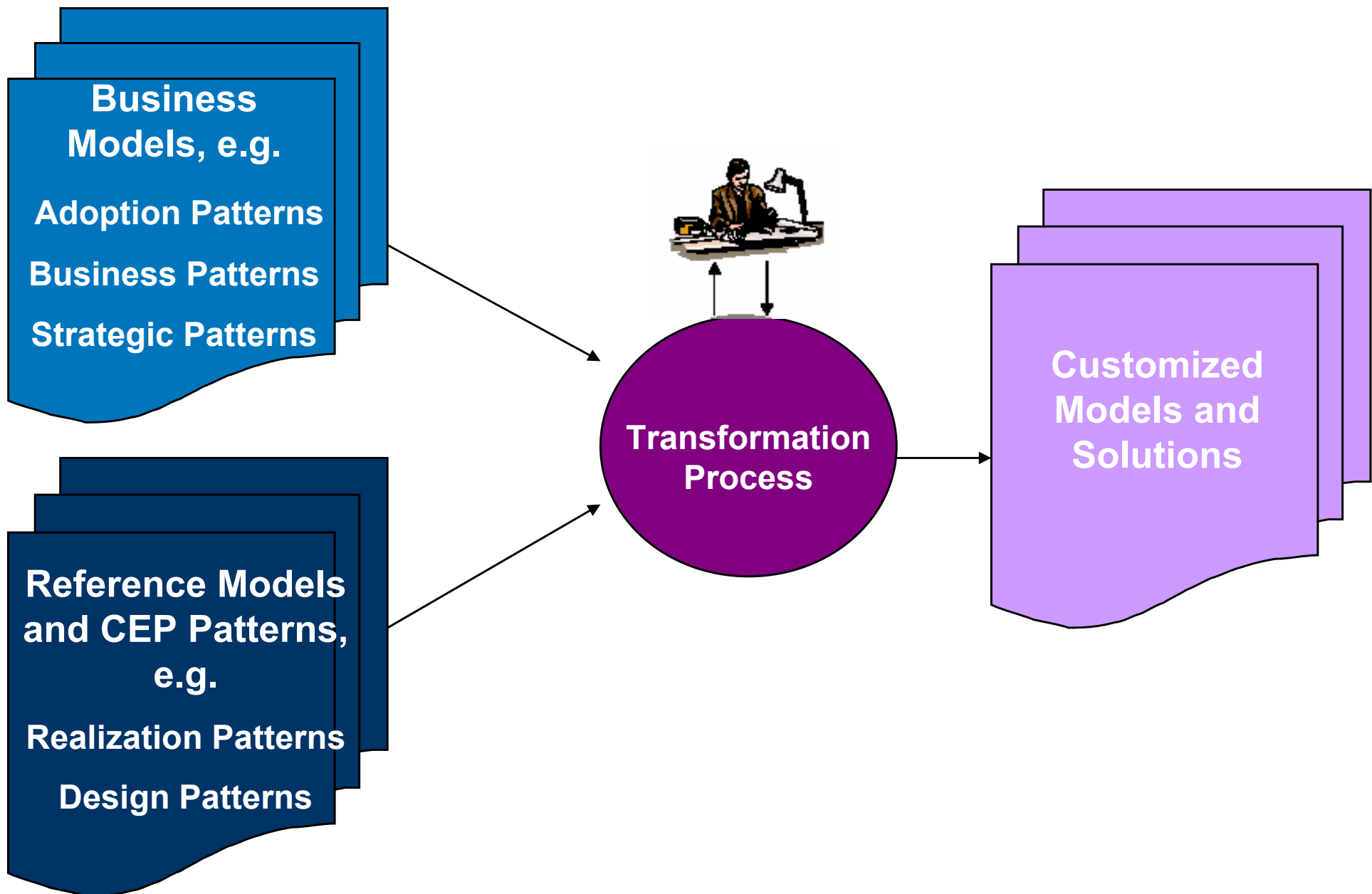
Code base after refactoring

Refactored code base

Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- Modeling and Pattern Languages for CEP
- **Brainstorming on automation and support of CEP design**
- Summary of achieved results, ideas and next steps

Semi-Automated Transformation



Agenda

- Introduction
- Theoretical Ground Work
 - Discuss fundamental terminologies, definitions and relations
 - Discourse the differences between CEP models & design patterns and complex event patterns
- Reference Models and CEP Patterns
 - Survey existing CEP reference models and patterns
 - Discuss categorization scheme for CEP patterns
- Modeling and Pattern Languages for CEP
- Brainstorming on automation and support of CEP design
- **Summary of achieved results, ideas and next steps**



Thank you !