

Fallbasiertes Schließen in der Nanotechnologie – *wie können Prozesse, die sich ähneln, das Chipdesign beschleunigen?**

*Das URANOS Projekt wird vom BMBF unter der Projektnummer 01M3075 gefördert.

Mirjam Minor
Daniel Schmalen
Alexander Tartakovski
Ralph Bergmann

www.wi2.uni-trier.de

Andreas Koldehoff



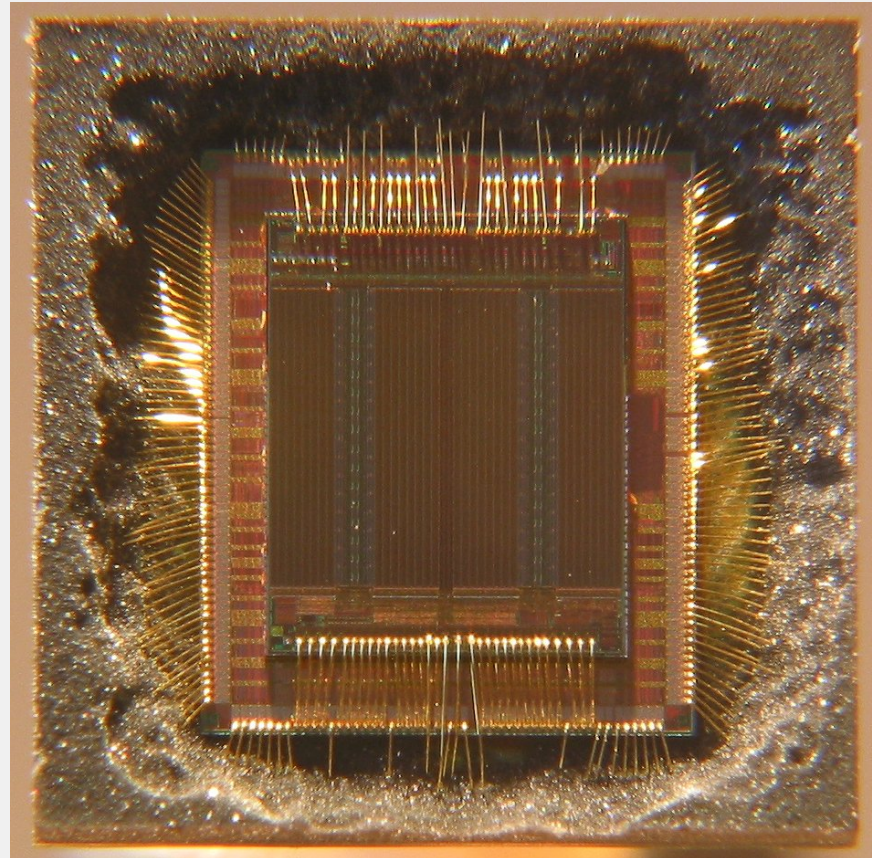
Agenda

- Motivation: Chipentwurf im Zeitalter der Nanotechnologie
- Agile Workflows
 - Begriffsbestimmung
 - Definitionen und Instanzen
 - Workflow-Pattern zur Beschreibung des Kontrollflusses
 - Ontologie-basiertes, konfigurierbares Kontextmodell
- Fallbasierte Unterstützung der Modellierung und Anpassung
 - Grundmechanismus (Fallbasis, Anfrage, Retrieval, Adaption)
 - Repräsentation der Fälle
 - Ähnlichkeitsberechnung (Editierdistanzen für Graphen)
 - Experiment mit verschiedenen Repräsentations-Varianten (Approximationen)



Chipentwurf mit Nanotechnologie

- Heutige Chiptechnologien:
 - 90 Nanometer,
 - Erste Prototypen mit 60nm
 - Infineon: 45 nm bis 2008
- Auswirkungen der neuen Technologien auf den Entwurfsprozess:
 - Neue Fehlertypen
 - Unsicherheit, ob die bisherigen Algorithmen verwendbar sind



Chipentwurf mit Nanotechnologie (Fortsetzung)

- Anforderungen an den Entwurf:

- „First time right delivery“
- Kurze „time-to-market“
- Unwägbarkeiten von Seiten des Kunden, viele weitere Risiken meistern



- Unser Ansatz:

- *Agile Workflow-Technologie* um laufende Prozesse an Veränderungen anpassen zu können
- *Intelligente Techniken* (Fallbasiertes Schließen, Ontologien) zur Unterstützung



Agile Workflows

- Für die inkrementelle und flexible Modellierung von Prozessen:
 - Workflow-Instanzen werden von Workflow-Definitionen (Templates) abgeleitet
 - Anpassung laufender Prozesse durch
 1. Ad-hoc-Änderungen an einzelnen Workflow-Instanzen
z.B. [Reichert et al. 2003, Weber et al. 2005]
 2. Modifikationen von Workflow-Definitionen, die schon Instanzen gebildet haben
z.B. [Reichert et al. 2003]
 3. Late-planning und hierarchische Dekomposition
z.B. [van Elst et al. 2003, Freßmann et al. 2005]
- Wir haben einen neuen, kollaborativen Ansatz mit Schwerpunkt auf 1. und 3. entwickelt.

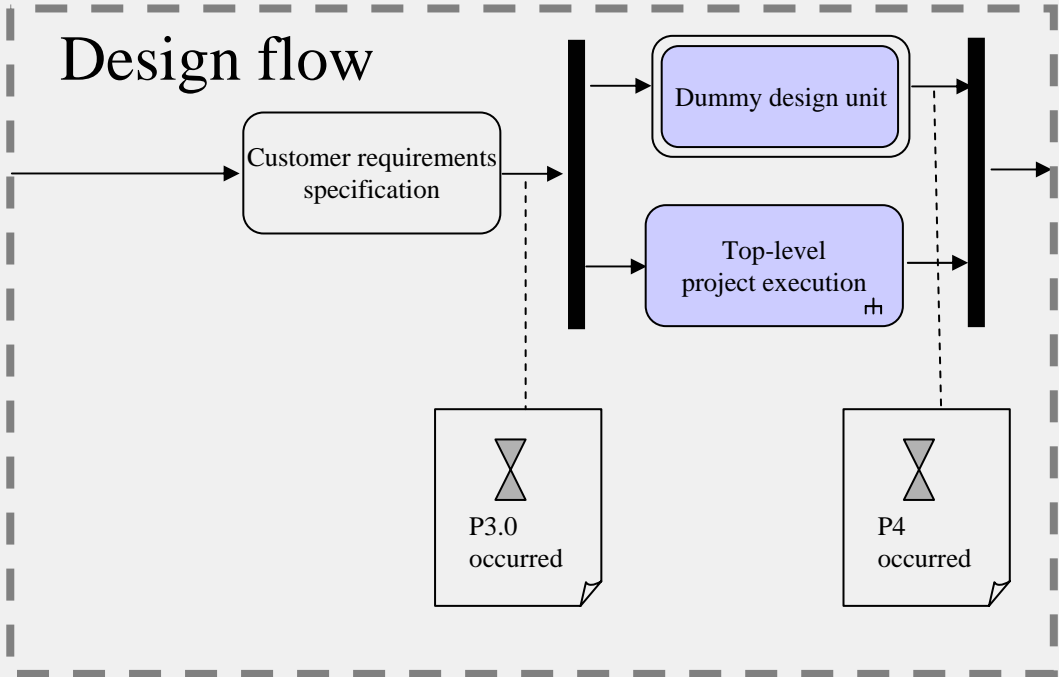
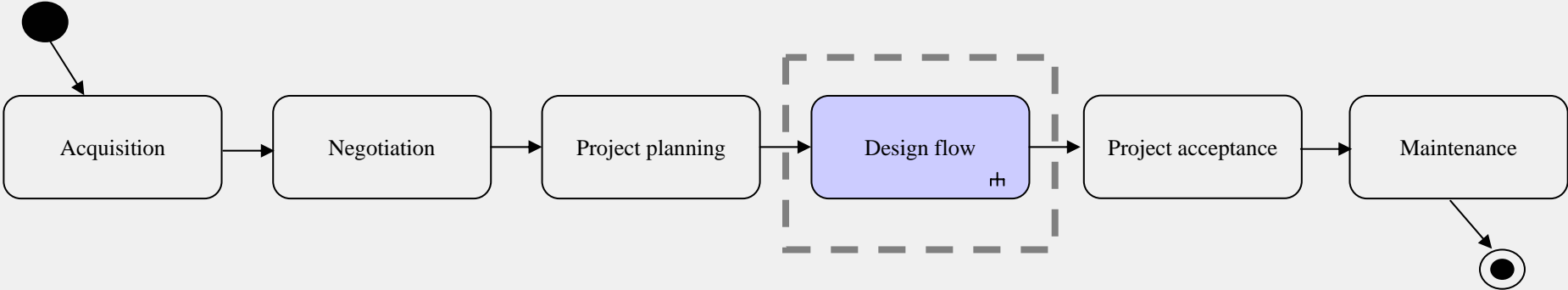


Definitionen und Instanzen

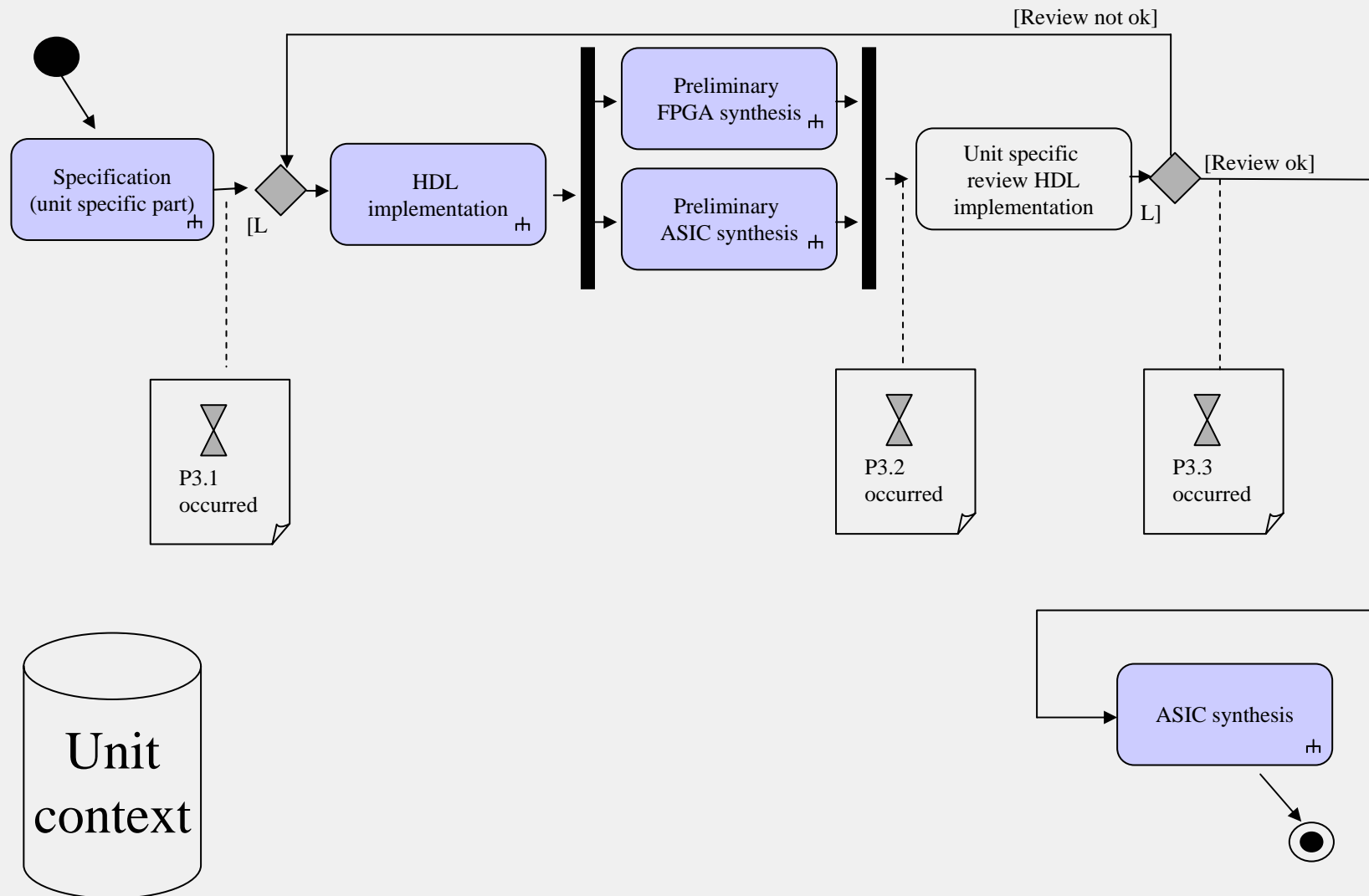
- Workflow-Definitionen (WFD's) nach SciWay 2.0, einer standardisierten Beschreibung der Design-Schritte bei unserem Industriepartner sci-worx:
 - Für Top-Level-Design-Projekte
 - Für Design-Units (Module)
- Definitionen und Instanzen bestehen jeweils aus
 1. Kontrollfluss-Struktur von Tasks (Aufgaben)
 2. Kontext-Modell



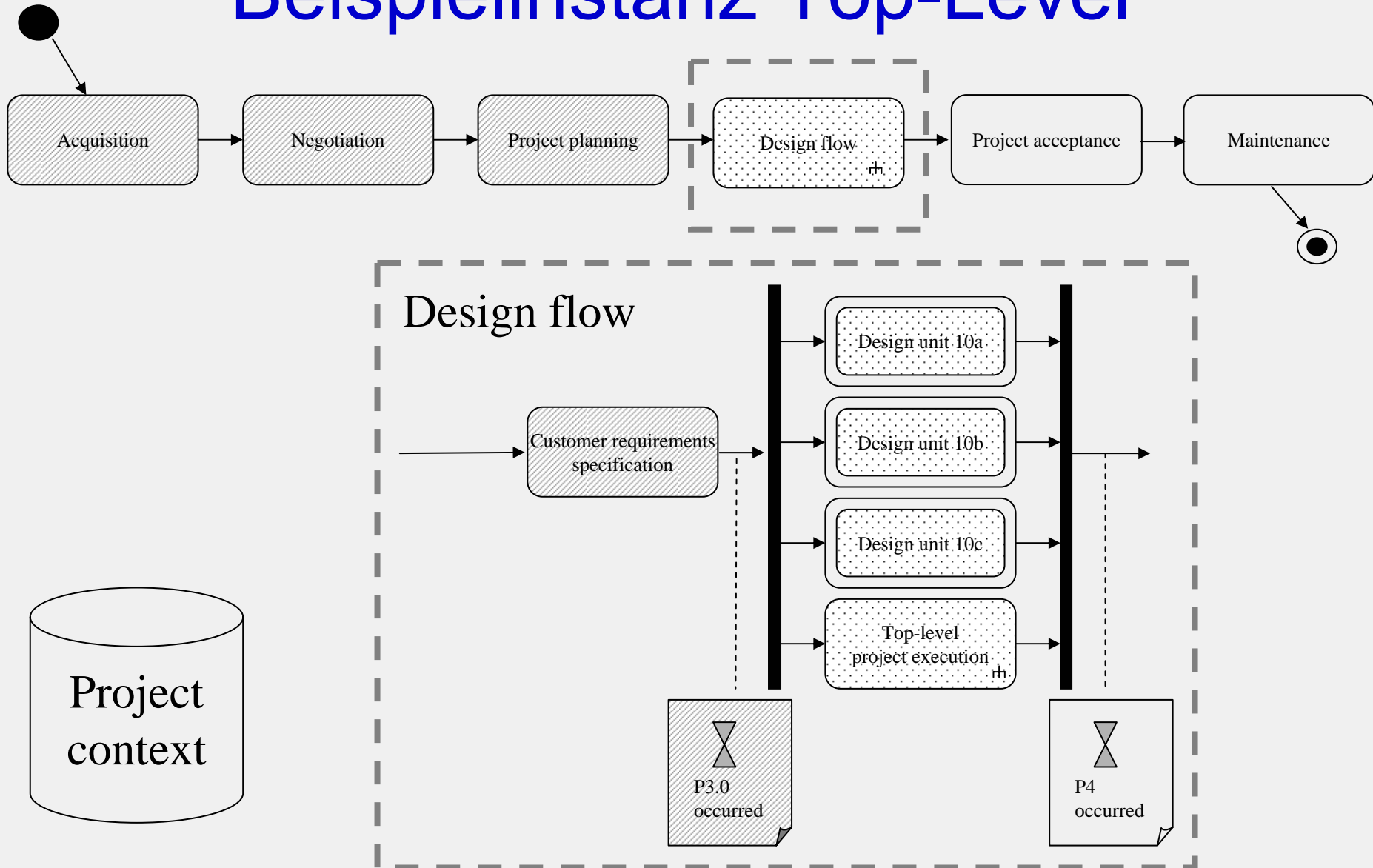
WFD für Top-Level-Design-Projekte



WFD für Design-Units



Beispielinstanz Top-Level



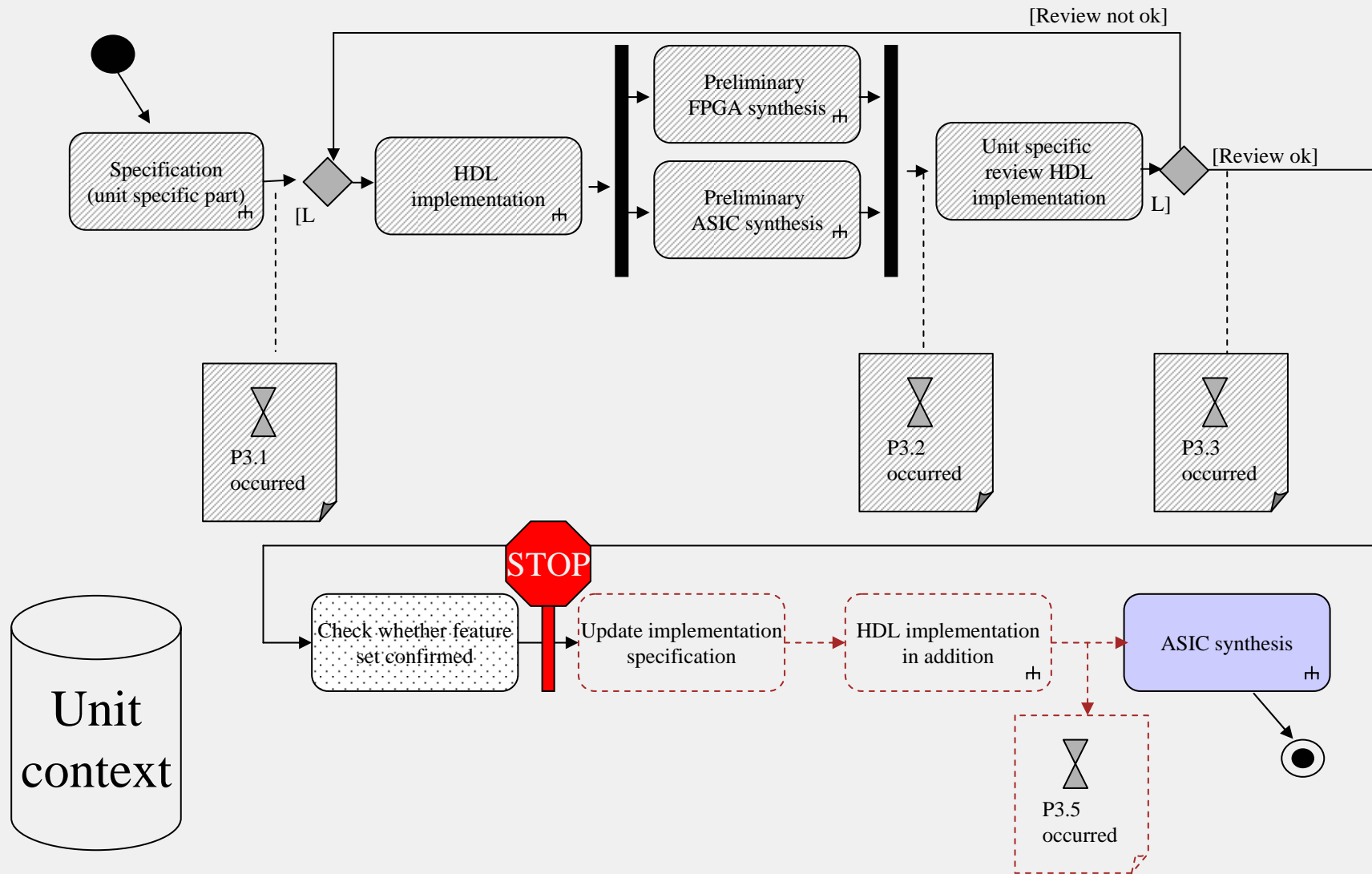
Sprache zur Beschreibung des Kontrollflusses

- Notation der Kontrollfluss-Elemente nach den Workflow-Pattern in [van Aalst et al. 2003]
 - Fünf Basis-Pattern
 - Sequenz
 - AND-split
 - AND-join
 - XOR-split
 - XOR-join
 - Loops als strukturierte Schleifen mit
 - einem Eintrittspunkt (LOOP-join) und
 - einem Austrittspunkt (LOOP-split).

(Loop-Blöcke dürfen verschachtelt, aber nicht verschränkt sein.)
- Drei zusätzliche Kontrollfluss-Elemente für die Adaptivität
 - Platzhalter-Tasks für Sub-Workflows
 - Platzhalter-Tasks für Sub-Diagramme
 - Stopp-Schilder



Beispiel-Instanz Design-Unit 10a



Kontext-Modell

- Der Kontext eines Workflows besteht aus einer Menge von Kontextfaktoren.
- Im Chipdesign sind dies Faktoren
 - mit wichtigem Einfluss auf den Design-Prozess des Projekts,
 - aus dem Applikationskontext (Kunde, Endkunde) und aus dem Entwurfskontext (Designer, Tools),
 - erfasst in einem formalen Kontextmodell.
- Je nach Anwendungstyp spielen unterschiedliche Faktoren eine Rolle, d.h. die Kontextmodelle müssen *konfigurierbar* sein!



Beispiel-Kontextfaktoren

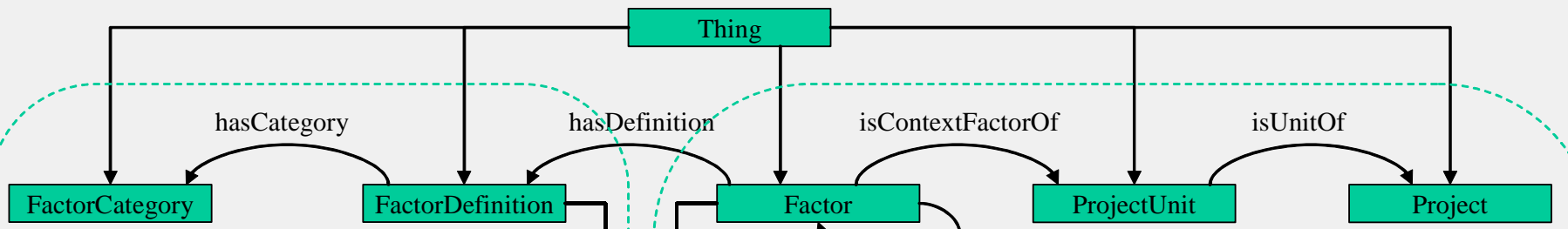
- Z.B. im Multimedia-Bereich:
 - Ports der einzelnen Units
 - „**Uncertainty of customer requirements specification:**
*GMC with global motion search in case of prerunning ME module
(for further discussion)*“
in der Unit „*Motion Compensation*“ des Projekts „*Multi-Standard-Video-Encoder*“
- Z.B. im Automotive-Bereich:
 - Temperaturbereich, z.B. in Automatikgetrieben
 - angestrebte Lebensdauer
 - Sicherheitsrelevante Faktoren wie Verifizierbarkeit



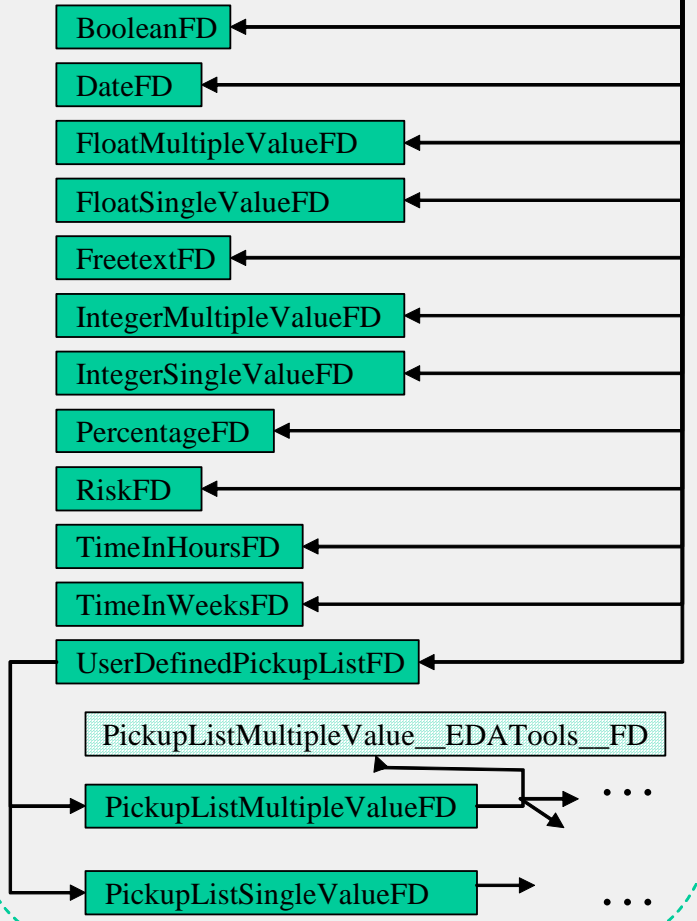
Austauschformat

- Für den Import und Export von Kontextmodellen
- Basiert auf OWL/RDF/XML (SemanticWeb-Standard)
- Unterscheidung in Modell-Definition und Anwendung:
 - *Pool von Faktoren* zur Definition von Kontextfaktoren (Name, Kontextkategorie, Wertebereich, Default-Wert)
 - *Anwendung der Faktoren* auf Design-Projekte (Zuordnung zu Projekt-Units, Werte, Abhängigkeiten)
- Top-level Ontologie legt die Modellstruktur und die Datentypen fest, die vom Kontexttool unterstützt werden (siehe nächste Folie)

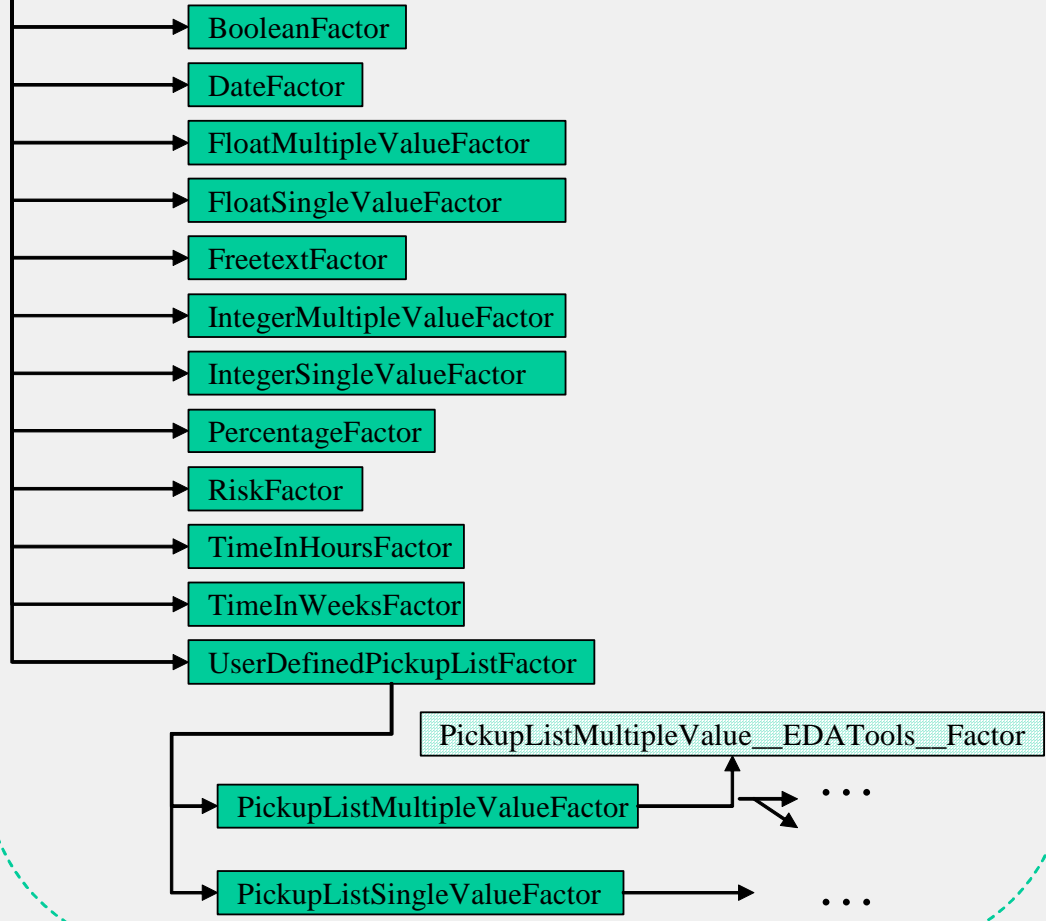




Pool of context factors



Applied context factors



Agenda

- Motivation: Chipentwurf im Zeitalter der Nanotechnologie
- Agile Workflows
 - Begriffsbestimmung
 - Definitionen und Instanzen
 - Workflow-Pattern zur Beschreibung des Kontrollflusses
 - Ontologie-basiertes, konfigurierbares Kontextmodell



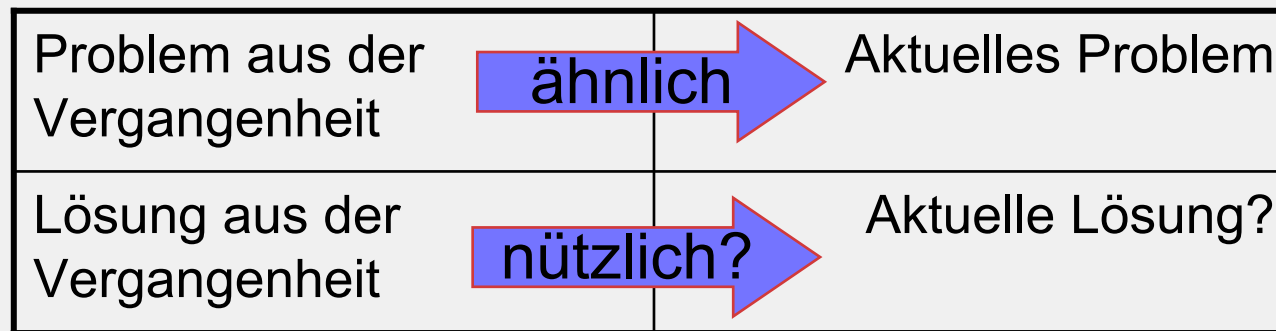
Fallbasierte Unterstützung der Modellierung und Anpassung

- Grundmechanismus (Fallbasis, Anfrage, Retrieval, Adaption)
- Repräsentation der Fälle
- Ähnlichkeitsberechnung (Editierdistanzen für Graphen)
- Experiment mit verschiedenen Repräsentations-Varianten (Approximationen)



Fallbasierte Unterstützung

- Grundprinzip des Fallbasierten Schließens:



- Anwendung für die Workflow-Modellierung:
Wiederverwenden von Erfahrungswissen, wie Workflow-Instanzen angepasst wurden



Repräsentation der Workflows

- Kontrollfluss...
 - ... einer Workflow-Definition wird repräsentiert als Menge von Elementen (Tasks, Kontrollfluss-Elemente) und einer Vorgänger-Nachfolger-Beziehung
 - ... einer Workflow-Instanz wird repräsentiert als Differenz zu ihrer Workflow-Definition:
 1. Strukturelle Änderungen bzgl. Tasks und Kontrollfluss-Elementen
 2. AbarbeitungszustandBeides lässt sich durch *add* und *delete*-Mengen beschreiben: Abgearbeitete Elemente kommen einfach in die *delete*-Menge.
- Kontextmodell
 - ... wird als Menge von Attribut-Werte-Paaren repräsentiert (klassischer, fallbasierter Ansatz zur Repräsentation von Fällen)

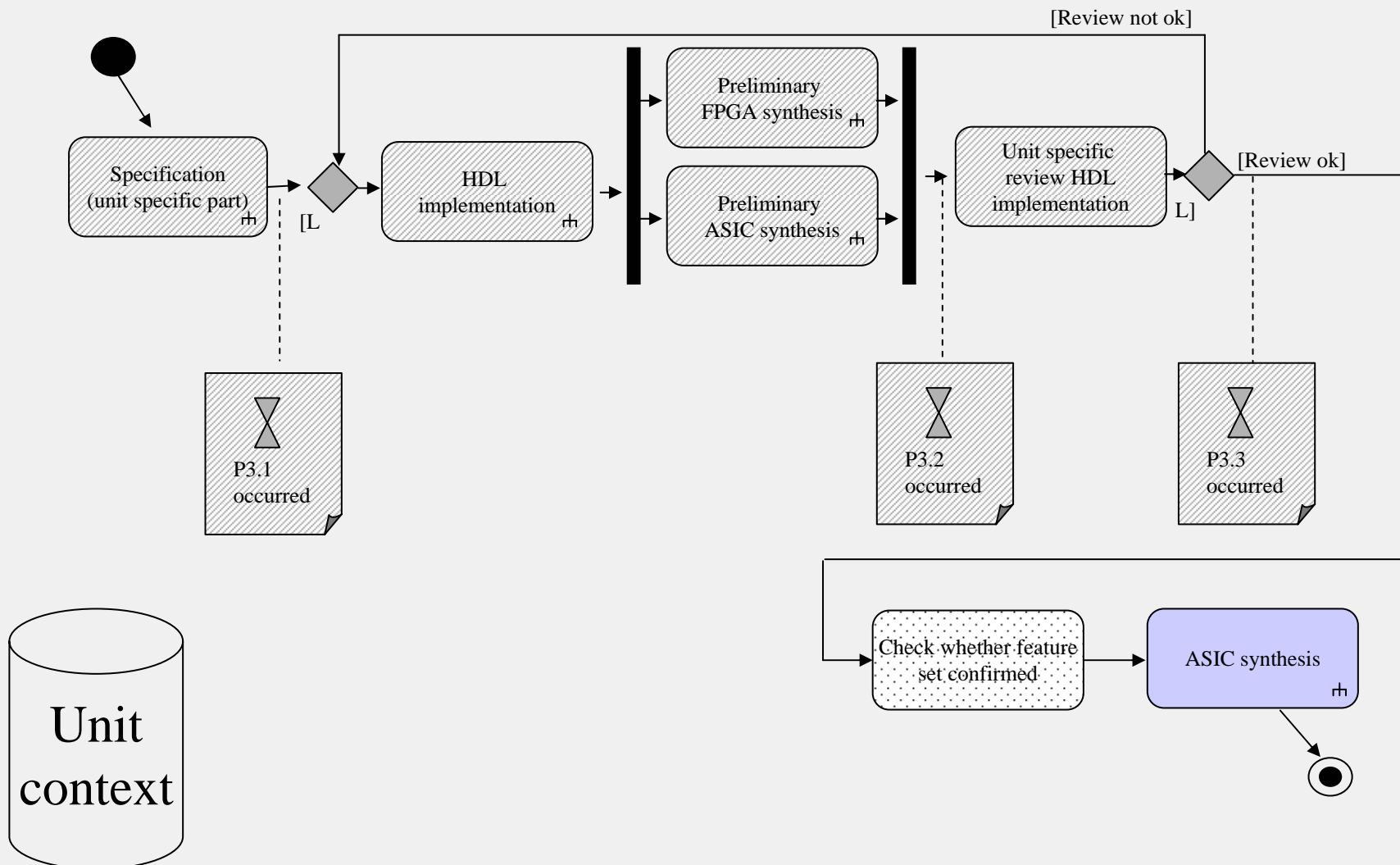


Repräsentation von Fällen/Anfragen

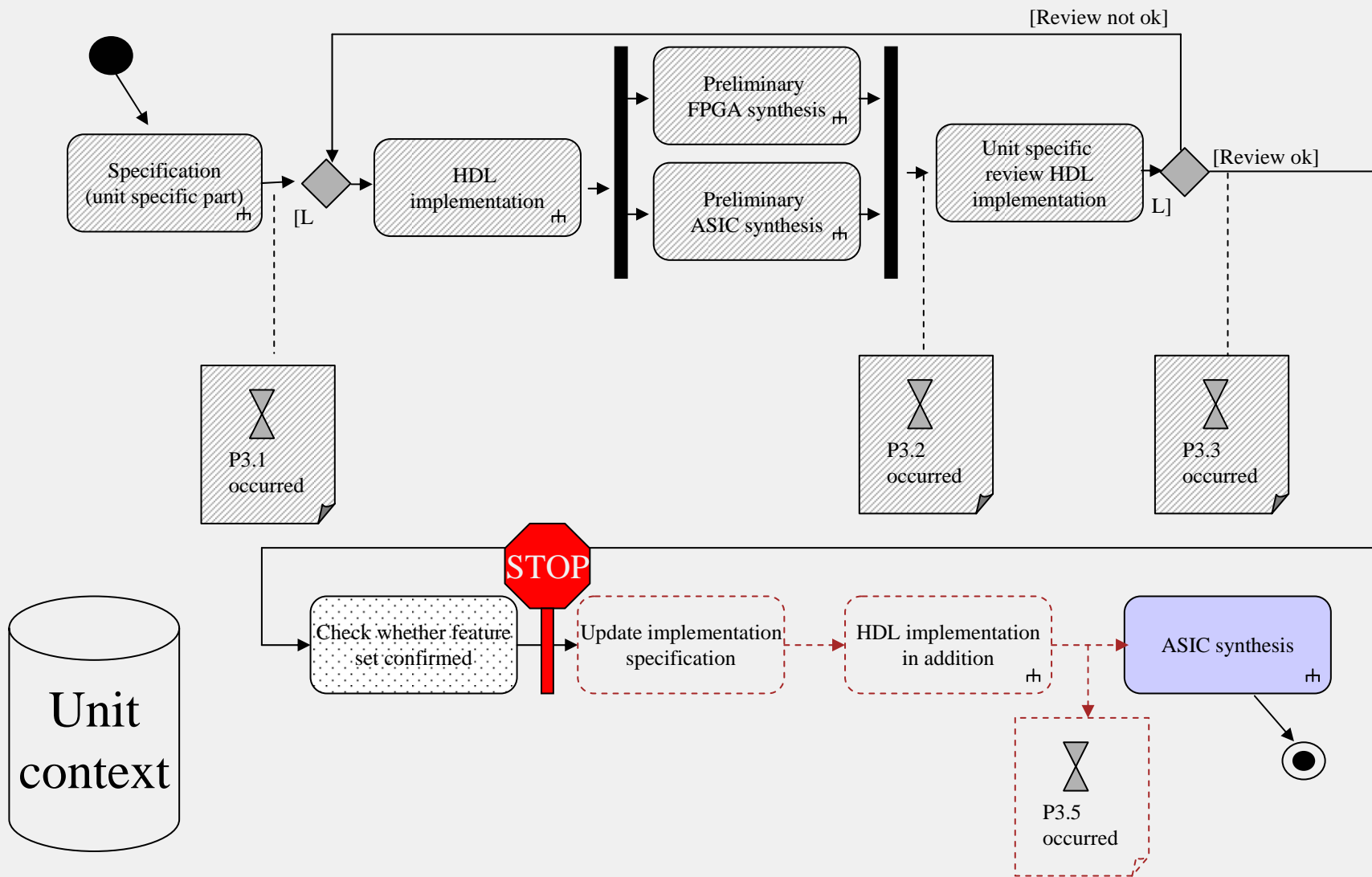
- Ein Fall repräsentiert Erfahrungswissen, wie eine laufende Workflow-Instanz modifiziert wurde.
- Fall = Paar von Revisionen einer laufenden Workflow-Instanz $[X, X']$ mit
 - X : Problemteil des Falls, die Revision vor der Modifikation
 - X' : Lösungsteil des Falls, die modifizierte Revision
- Eine Anfrage besteht nur aus einer aktuellen Revision Y .
- Retrieval: Bestimmung der *best matching cases* aus der Fallbasis durch paar-weisen Vergleich von Anfrage und gespeicherten Fällen.



Revision X einer Beispiel-Instanz



Revision X' einer Beispiel-Instanz



Ähnlichkeit von Workflow-Instanzen

1. Kontextmodell: klassische Ähnlichkeitsfunktion nach dem lokal-global-Prinzip, z.B. gewichtete Summe
2. Kontrollfluss: Ähnlichkeitsfunktion nutzt Editierdistanz von Graphen (wie transformiere ich G_1 in G_2)

Gesamtähnlichkeit zweier Fälle: aggregierter Wert aus 1., 2.

Im weiteren Vortrag betrachten wir nur 2.



Graph-Editierdistanz

[Bunke, Messmer 1993]

- Atomare Editier-Operation e
 - Füge neuen Knoten ein
 - Füge neue Kante ein
 - Lösche einen Knoten
 - Lösche eine Kante
 - Modifiziere Knotenbezeichner
 - Modifiziere Kantenbezeichner
- Kosten $c(e) \in [0,1]$
- x, y seien gerichtete Graphen mit bezeichneten Knoten und Kanten, Editierdistanz zwischen x und y :

$$\delta(x, y) = \min \left\{ \sum_{i=1}^k c(e_i) \mid (e_1, \dots, e_k) \text{ transformiert } x \text{ in } y \right\}$$



Abstrakter *View* auf Workflow-Instanzen

- Die Berechnung allgemeiner Graph-Editierdistanzen ist NP-vollständig!
- Wir betrachten eingeschränkte Graphen (Workflow-Instanzen) und davon zunächst nur
 - Tasks (N),
 - Task-Namen ($TaskNames$),
 - Reihenfolge von Tasks, gegeben durch Sequenz-Pfeile (E)

$View = \langle N, E, name \rangle$

$name : N \rightarrow TaskNames$



Editierdistanz im *View*

- $\delta(V_1, V_2)$
für zwei Views $V_1 = \langle N_1, E_1, name_1 \rangle$ und $V_2 = \langle N_2, E_2, name_2 \rangle$
- Für die Berechnung von δ kann man zwei Charakteristika nutzen:
 - Die Namen der Tasks in einer Workflow-Instanz sind eindeutig.
 - Zwei Tasks t_1 und t_2 aus verschiedenen Workflow-Instanzen können als identisch betrachtet werden *genau dann wenn* sie den gleichen Namen haben.

D.h. man kann mit Differenzmengen arbeiten:

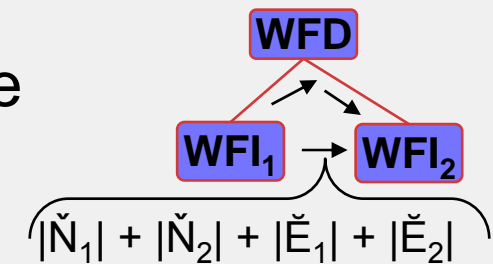
$$\begin{aligned}\check{N}_1 &:= N_1 \setminus N_2 & \check{E}_1 &:= E_1 \setminus E_2 \\ \check{N}_2 &:= N_2 \setminus N_1 & \check{E}_2 &:= E_2 \setminus E_1\end{aligned}$$

so dass $\delta(V_1, V_2) = |\check{N}_1| + |\check{E}_1| + |\check{N}_2| + |\check{E}_2|$ (quadratisch!)



Berechnung der Ähnlichkeitsfunktion

- Optimierung der Berechnung von δ mit Hilfe der Workflow-Definition,



dazu werden \check{N}_1 , \check{N}_2 , \check{E}_1 , \check{E}_2 anders definiert:

$$\check{N}_1 := \{N_T \cup \text{add.nodes}_{V_1} \setminus \text{delete.nodes}_{V_1}\} \setminus \{N_T \cup \text{add.nodes}_{V_2} \setminus \text{delete.nodes}_{V_2}\}$$

$\check{N}_2 \dots$

- Differenzwert δ in Ähnlichkeitswert *sim* umrechnen:

$$\text{sim}(V_1, V_2) := 1 - \frac{\delta(V_1, V_2)}{|N_1| + |N_2| + |E_1| + |E_2|}$$



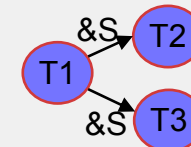
Experimente mit drei verschiedenen Approximationen

- Erweiterung des View, so dass auch die anderen Kontrollfluss-Elemente (AND-split, AND-join usw.) berücksichtigt werden

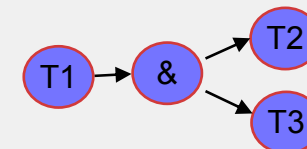
- Kontrollfluss-Elemente außer Sequenzpfeilen werden zu „direkten“ Kanten abstrahiert



- Wie 1., aber die Kanten bekommen Namen



- Für jeden Typ von Kontrollfluss-Element außer dem Sequenzpfeil wird je ein Stellvertreter-Knoten eingeführt (Multigraph: Bags von Kanten)

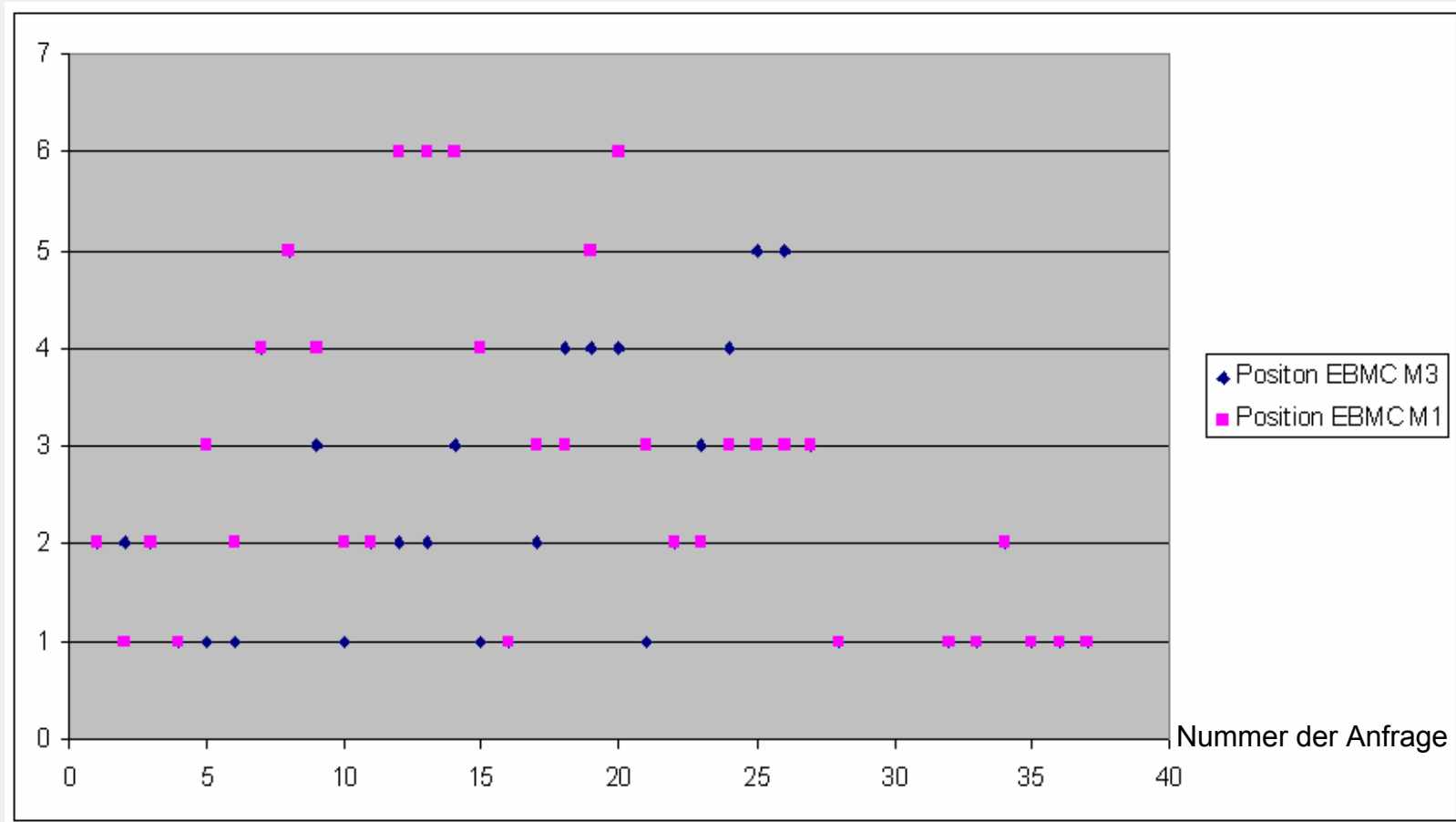


Evaluierungs-Szenario

- Experimentelle Evaluierung für Methode 1 und 3
 - Test-Fallbasis mit 37 Workflow-Instanzen aus echten Change-Request-Dokumenten in beiden Repräsentationsformen
 - Bestimmung des empirisch besten Falls (EBMC) für 35 Instanzen
 - Jeder Fall wurde als Anfrage an den Rest der Fallbasis gestellt
 - Die Retrieval-Ergebnisse für M1 und M3 wurden bezüglich des jeweiligen EBMC untersucht:
 - War der EBMC unter den ersten 10 Fällen?
 - Bei welcher Methode konnte er eine bessere Position im Retrieval-Ergebnis erreichen?



Evaluierungsergebnisse



Auswertung der Ergebnisse

- M3 (Stellvertreter-Knoten für jeden Typ von Kontrollfluss-Element) war besser als M1 (Kontrollfluss-Elemente zu Kanten ohne Namen abstrahiert):
 - In beiden Methoden war der empirisch beste Fall 34 mal unter den ersten zehn Fällen gelandet (21 mal sogar unter den ersten drei!).
 - Bei 17 Anfragen war die Position für M1 und M3 gleich, 6 mal war M1 und 12 mal war M3 besser.
 - M1 benötigte naturgemäß weniger Speicherplatz als M3 (ca. 2/3 der Knoten und Kanten von M3).



Zusammenfassung

- Agile Workflows
 - Kontrollfluss
 - Kontext-Modell
- Fallbasiertes Schließen
 - Modellierungsunterstützung durch Erfahrungswissen
 - Kontrollfluss-Ähnlichkeit auf der Basis von Graph-Editier-Distanzen
 - Verschiedene Repräsentationsformen (abstrahierter View + drei „echte“ Approximationen der empirischen Nützlichkeit)
 - Formative Evaluierung: Berücksichtigung von Kontrollfluss-Elementen als Knoten (M3) liefert bessere Ergebnisse als wenn sie in Kanten „hinein-compiliert“ werden.



Danke für die Aufmerksamkeit!

Fragen?
Anmerkungen?

