

Dissertation am

Institut für Informatik und Wirtschaftsinformatik
Institute for Computer Science and Business Information Systems
WIRTSCHAFTSWISSENSCHAFTEN



Erstgutachter: Prof. Dr. Klaus Pohl

Zweitgutachter: Prof. Dr. Bruno Müller-Clostermann

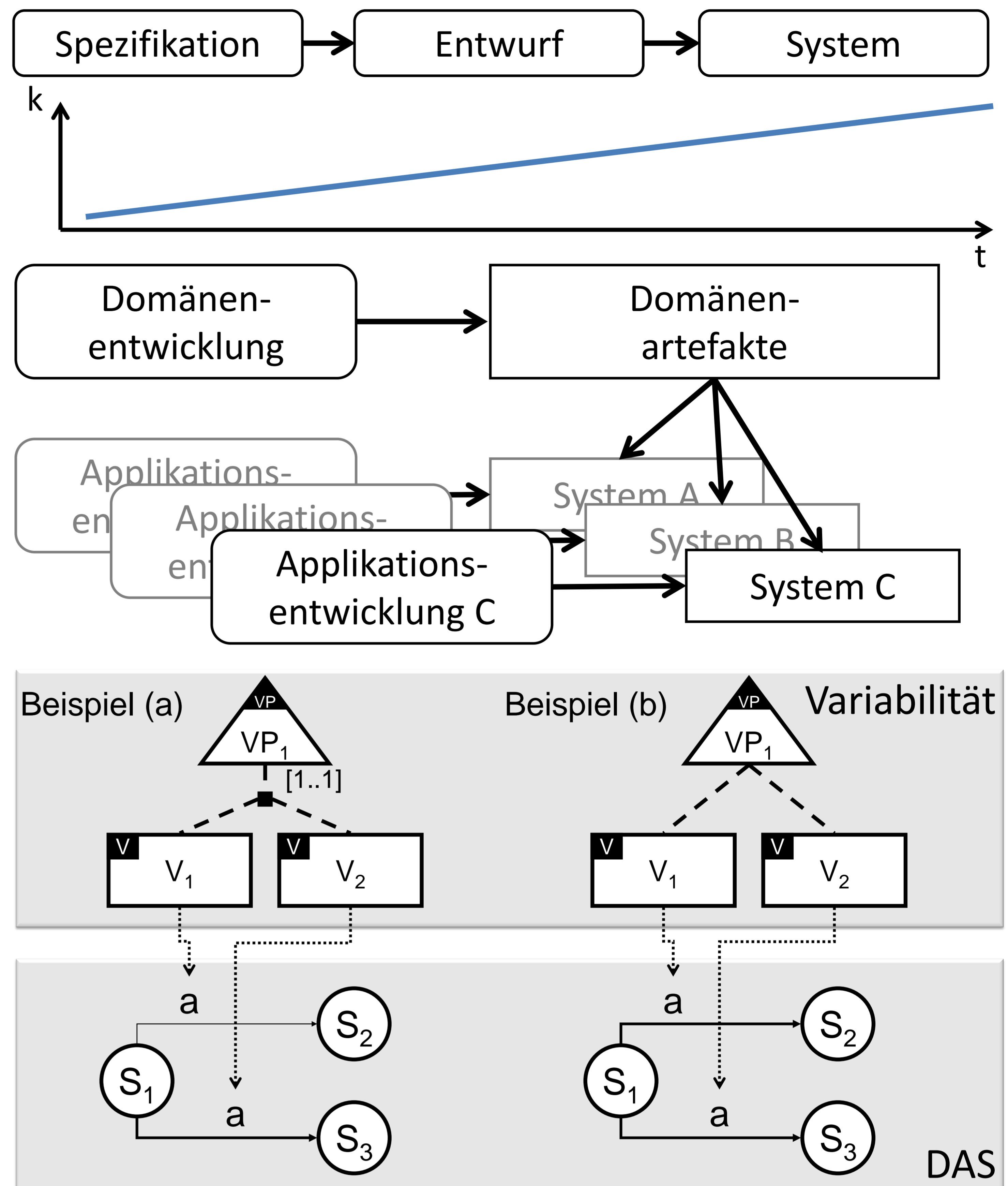
Grundlagen

Die Entwicklung eines Softwaresystems besteht typischerweise aus den Phasen Spezifikation, Entwurf und Realisierung. Je später Fehler (bspw. eine Inkonsistenz) im Entwicklungsprozess (t) identifiziert werden, desto höher sind die Kosten k für die Fehlerbehebung.

Produktlinienentwicklung (PLE) ist ein Paradigma zur parallelen Entwicklung gleichartiger Produkte. PLE unterscheidet Domänenentwicklung (Entwicklung für die Wiederverwendung) und Applikationsentwicklung (Entwicklung mit Wiederverwendung). Frühzeitige Qualitätssicherung insbesondere in der Domänenentwicklung verbessern daher Qualität der gesamten Produktlinie.

Problem

Die Domänenanforderungsspezifikation (DAS) ist Teil der Domänenentwicklung und dient der Entwicklung von Domänenartefakten. Die DAS dokumentiert die Anforderungen an eine Menge von Produkten (Variabilität). Die Variabilität wird durch ein Variabilitätsmodell (VM) beschrieben. Existierende Ansätze zur Konsistenzprüfung von Einzelsystemen sind auf DAS nicht anwendbar, da diese keine Variabilität berücksichtigen.



Ziel dieser Arbeit Konzeption von grundlegenden Verfahren für eine umfassende automatisierte Konsistenzprüfung von Domänenanforderungsspezifikationen in der Produktlinienentwicklung

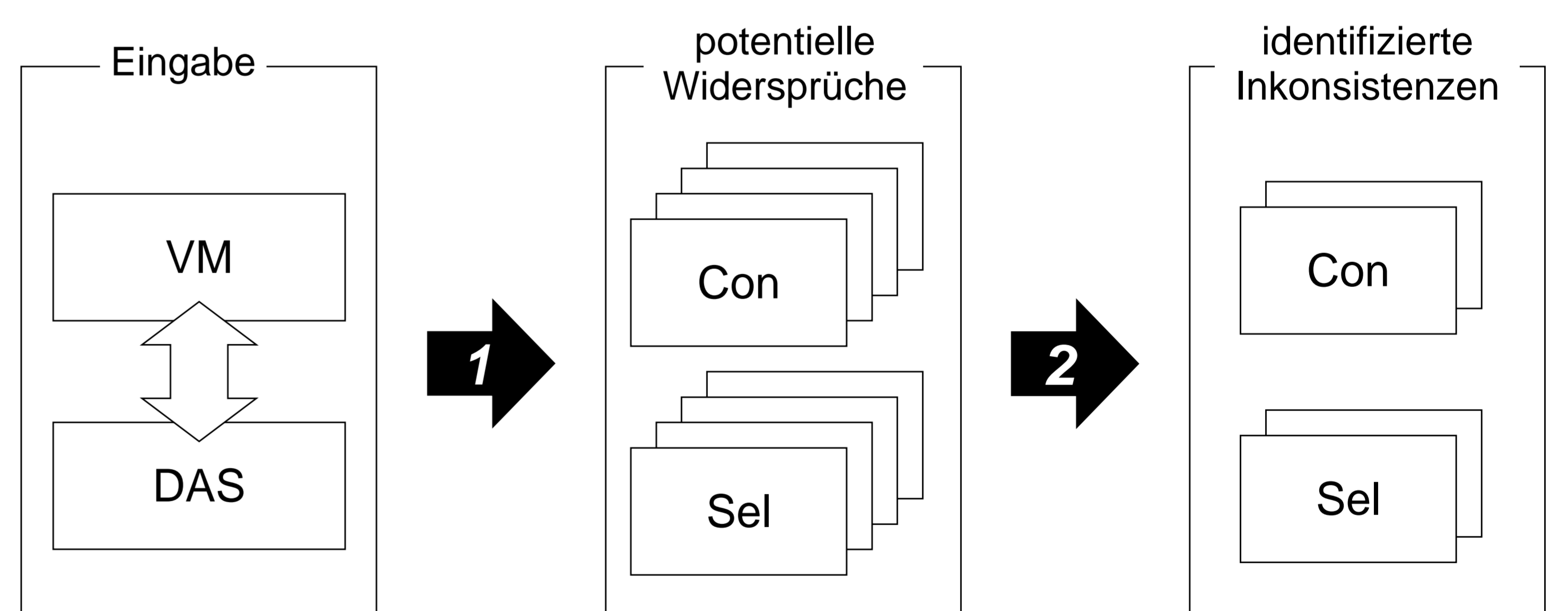
Prinzipieller Lösungsansatz

Als Eingabe wird die zu prüfende DAS und das Variabilitätsmodell erwartet. Schritte des prinzipiellen Lösungsansatzes sind:

- (1) Analyse der DAS auf potentielle Widersprüche
 - (2) Prüfung der potentiellen Widersprüche gg. VM
- Ergebnis sind identifizierte Inkonsistenzen in DAS. In dieser Arbeit wurde dieser Lösungsansatz auf verschiedene Konsistenzprüfungen angewandt.

Beispiel „Prüfung auf Determinismus“

Ein Softwaresystem zeigt ein deterministisches Verhalten, wenn das System für einen definierten Zustand z und ein Ereignis stets das gleiche Verhalten zeigt. In einer Produktlinie kann dieses Verhalten für unterschiedliche Produkte verschiedenen ausgestaltet sein, muss jedoch für das gleiche Produkt stets eindeutig sein. Auf der rechten Seite wird die formale Definition dieses Widerspruch und die Identifikation dieses Widerspruch am Beispiel dargestellt.



$$\text{Widerspruch}_{\text{Determinismus}}(K_i, VRel_{EAA}) := \{(Con_i, Sel_i)\} \text{ mit}$$

$$Con_i := \{t_m, t_x\} \text{ mit } t_m = z_m n_m z'_m, t_x = z_x n_x z'_x:$$

$$(z_m = z_x) \wedge (n_m = n_x), \text{ mit } (1 \leq m < x \leq |T_i|)$$

$$Sel_i := \bigwedge_{v \in V'} v \text{ mit } V' = \{v \mid \exists (v, T') \in VRel_{EAA}: T' \cap Con_i \neq \emptyset\}$$

Durchführung von Schritt (1) am Beispiel:

$$\text{Widerspruch}_{\text{Determinismus}}(\text{Beispiel}) := \{(Con_1, Sel_1)\} \text{ mit}$$

$$Con_1 := \{(S_1, a, S_2), (S_1, a, S_3)\}$$

$$Sel_1 := v_1 \wedge v_2$$

Durchführung von Schritt (2) am Beispiel:

<i>Prüfung für Beispiel (a)</i>	<i>Prüfung für Beispiel (b)</i>
$VM_{(a)} = v_1 \text{ XOR } v_2$	$VM_{(b)} = v_1 \text{ OR } v_2$
$VM_{(a)} \wedge Sel_1 = \text{false}$	$VM_{(b)} \wedge Sel_1 = \text{true}$

(a) ist konsistent

(b) ist inkonsistent



Dr. Kim Lauenroth ist wissenschaftlicher Assistent am paluno – The Ruhr Institute for Software Technology und vertritt dort das Gebiet Produktlinienentwicklung. Sein Forschungsschwerpunkt liegt im Requirements Engineering und Design von Produktlinien, insbesondere in der formalen Spezifikation und Verifikation.