Transformationen in musterbasierten Systemspezifikationen

Christian Peper

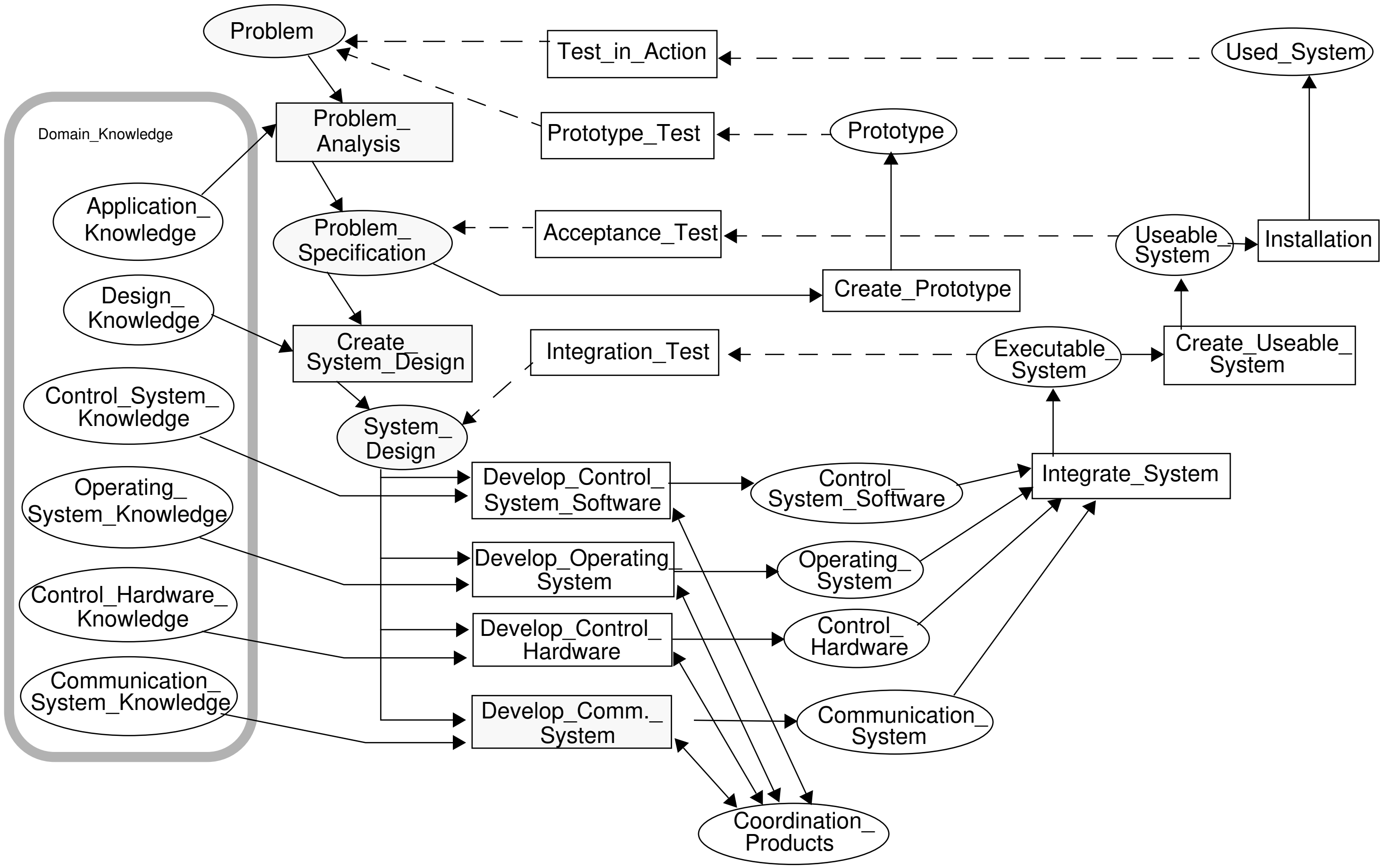
AG Rechnernetze

Sonderforschungsbereich 501

Universität Kaiserslautern

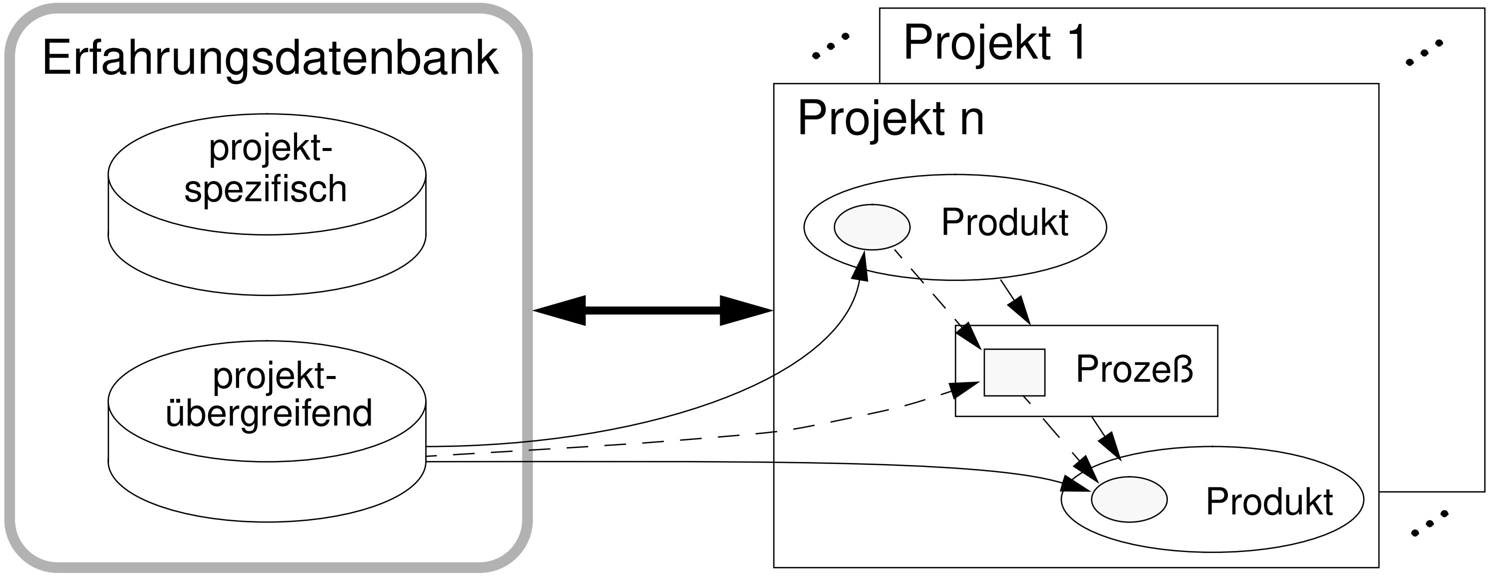
17. Juni 1999

SFB 501: Referenzprozeßmodell



modelliert durch B1

SFB 501: Erfahrungsdatenbank



Bisher: Musterbasierte Techniken auf *Produktebene*

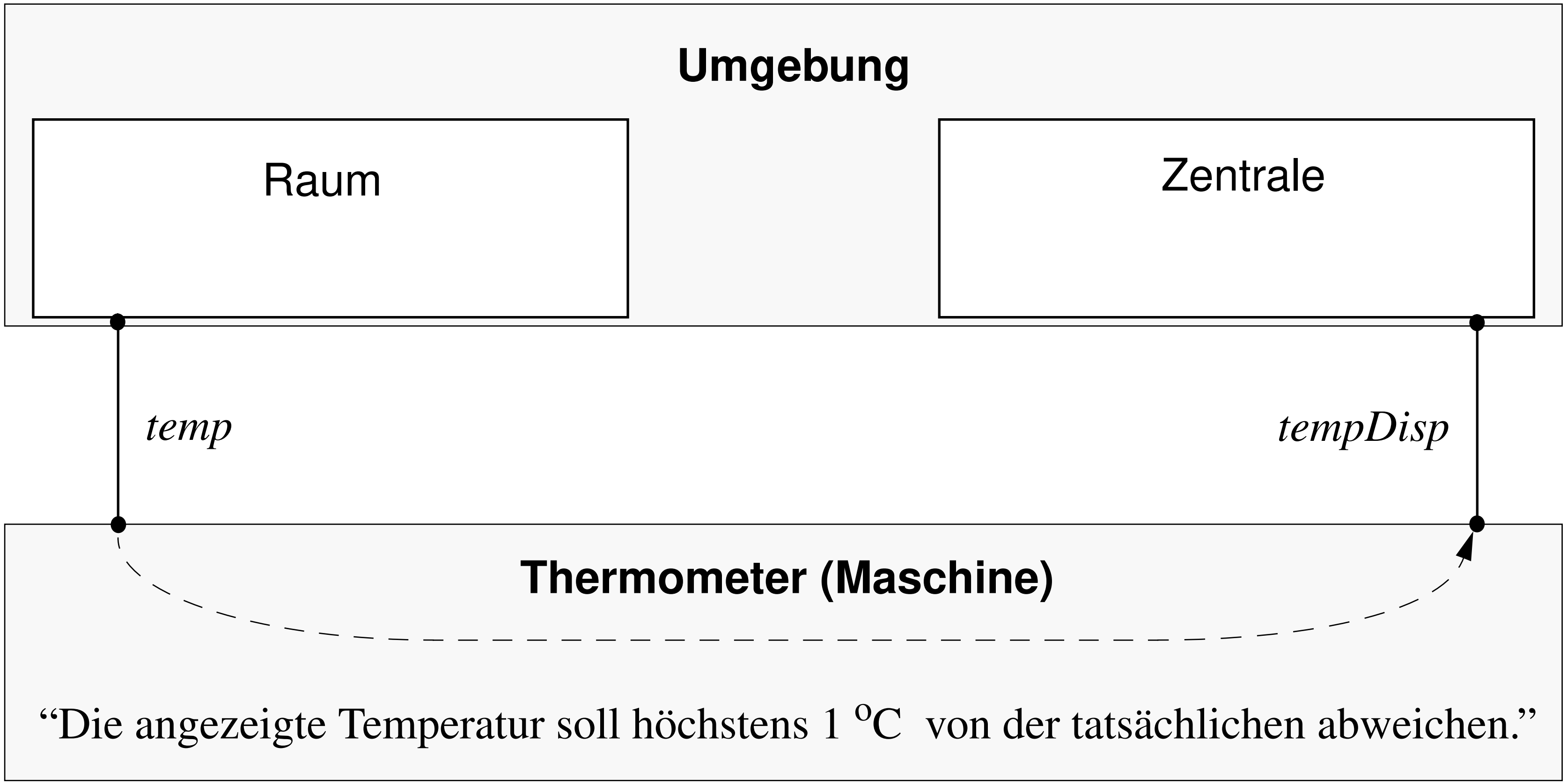
Sinnvoll: Musterbasierte Techniken auf *Prozeßebene*
$\rightarrow$ Transformation von Produktfragmenten, d.h. Musterinstanzen

Ziele

- **Muster**
  - Wiederverwendung von Produktfragmenten
  - Hohe Abstraktion

- **Transformationen**
  - Wiederverwendung von Prozeßfragmenten (Entwicklungsschritten, Designentscheidungen, ...)
  - Hohe Abstraktion
  - Bessere Verfolgbarkeit
  - Generierung

Beispiel: Problembeschreibung Thermometer



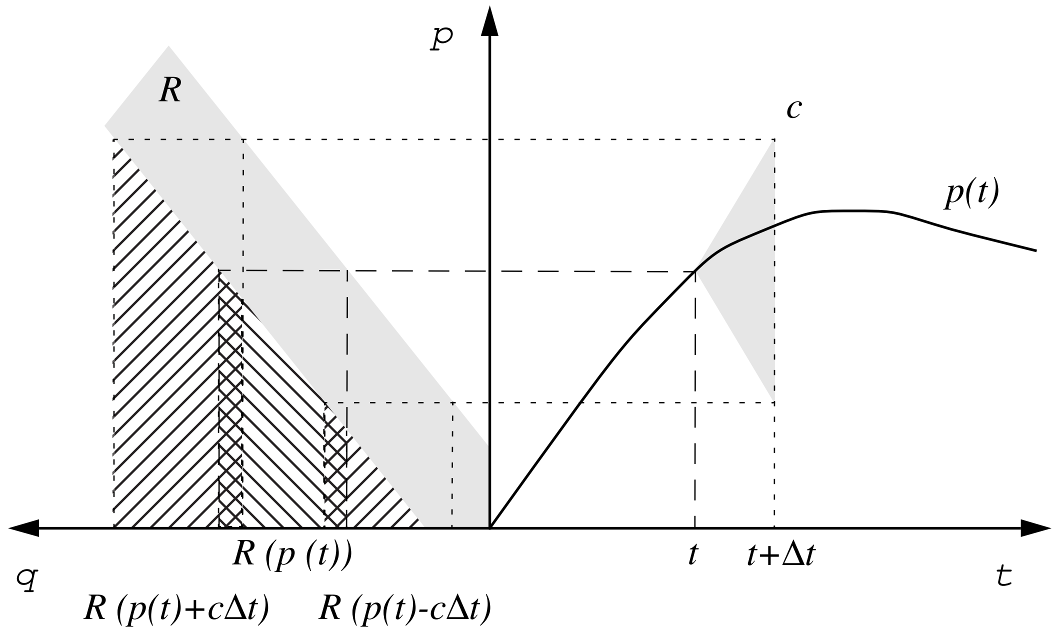
FOREST-Musterdefinition (1)

Pattern *SluggishSourceCopy (p, q, R, c)*

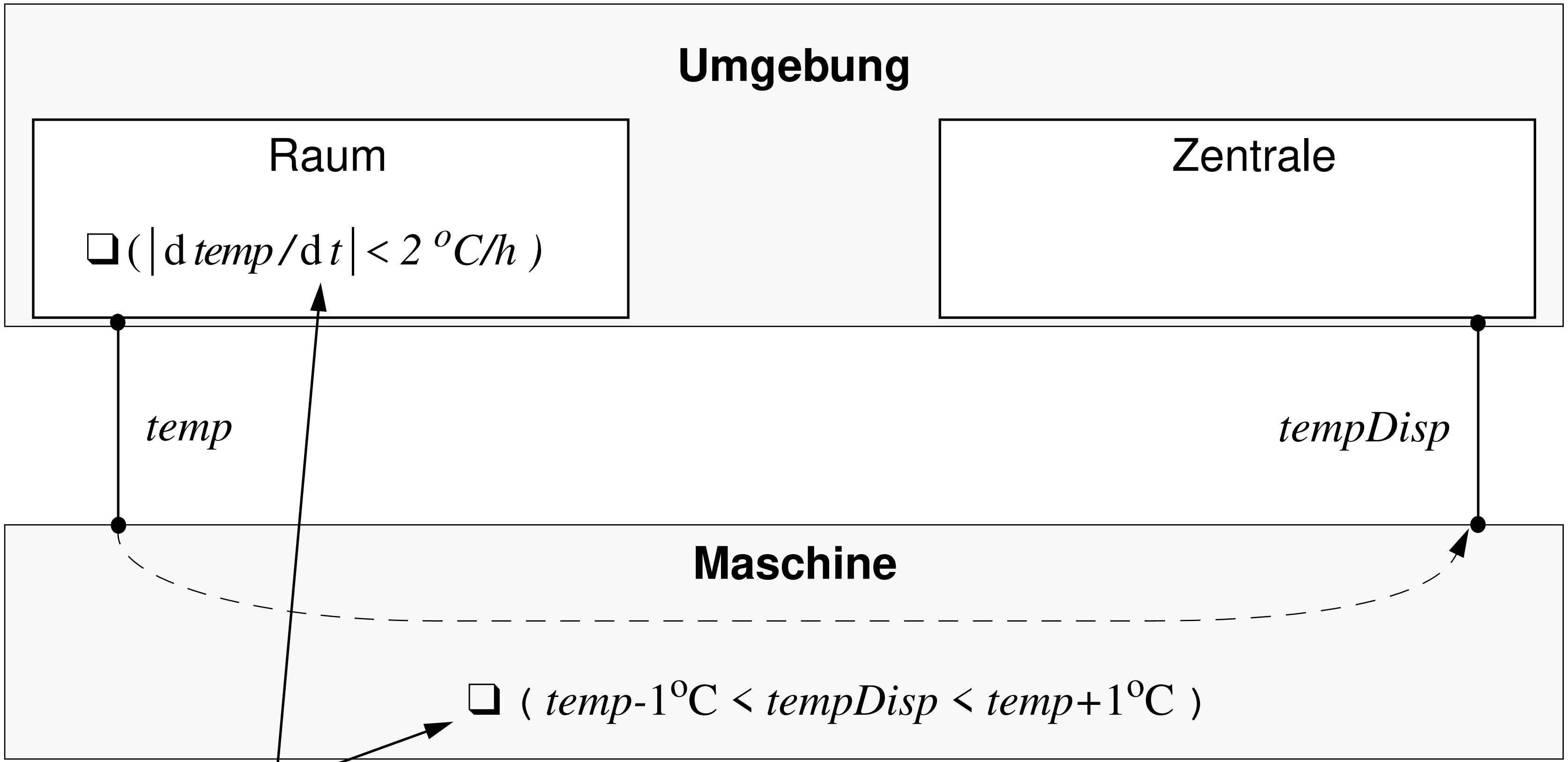
Intention : q is depending on a restricted p with respect to R.

Definition : ❑ $(p \Rightarrow_c^R q)$, where $p \Rightarrow_c^R q =_{\mathrm{Df}}$ $(p,q) \in R \wedge (|dp/dt| < c)$

Illustration :



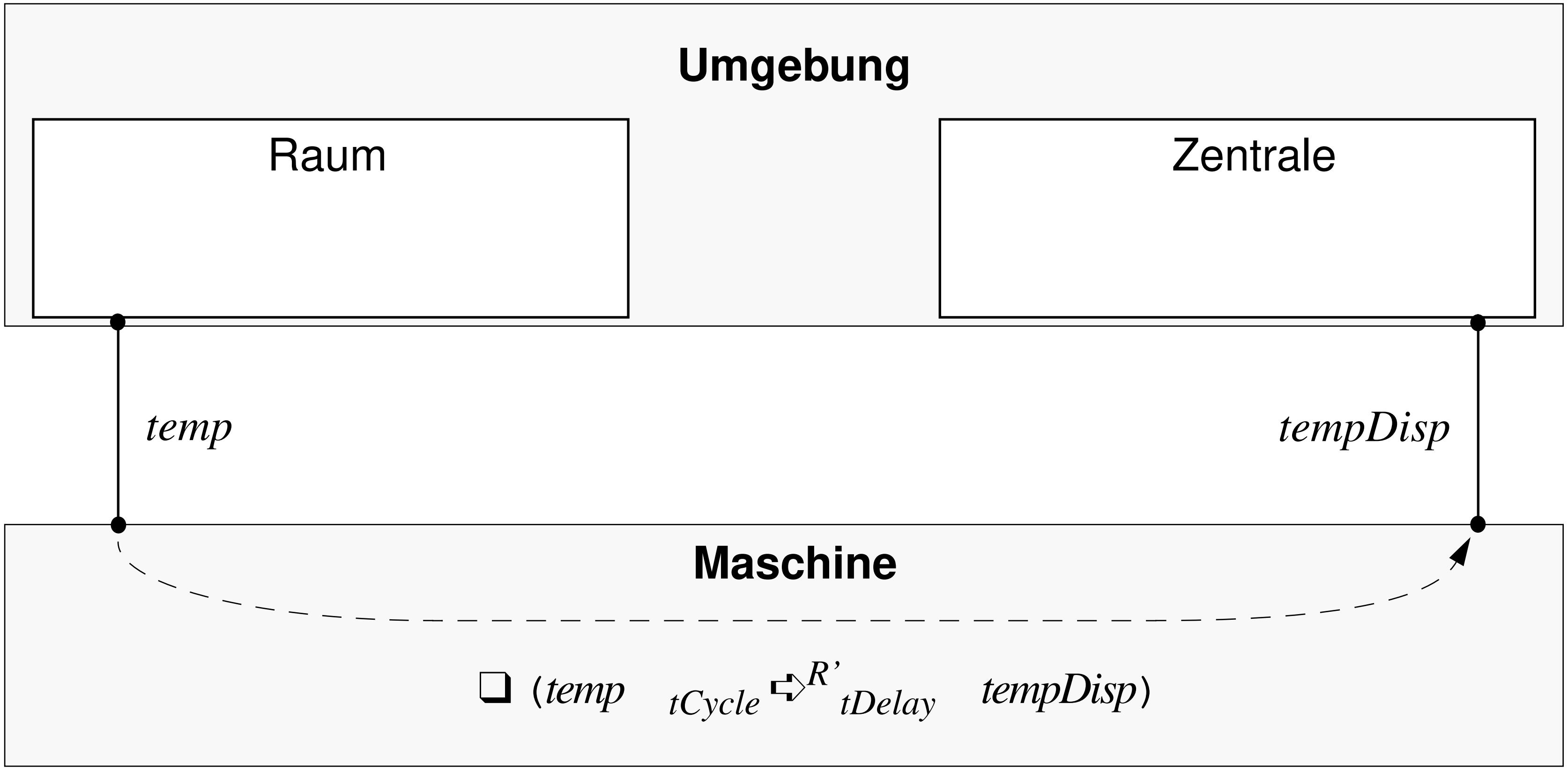
Beispiel: Anforderungen



Musterinstanz : ❑ $(temp \Rightarrow^{R}_{2^oC} tempValue)$

$(temp, tempValue) \in R$ iff $temp - 1^oC < tempValue < temp + 1^oC$

Beispiel: Erster Designschritt



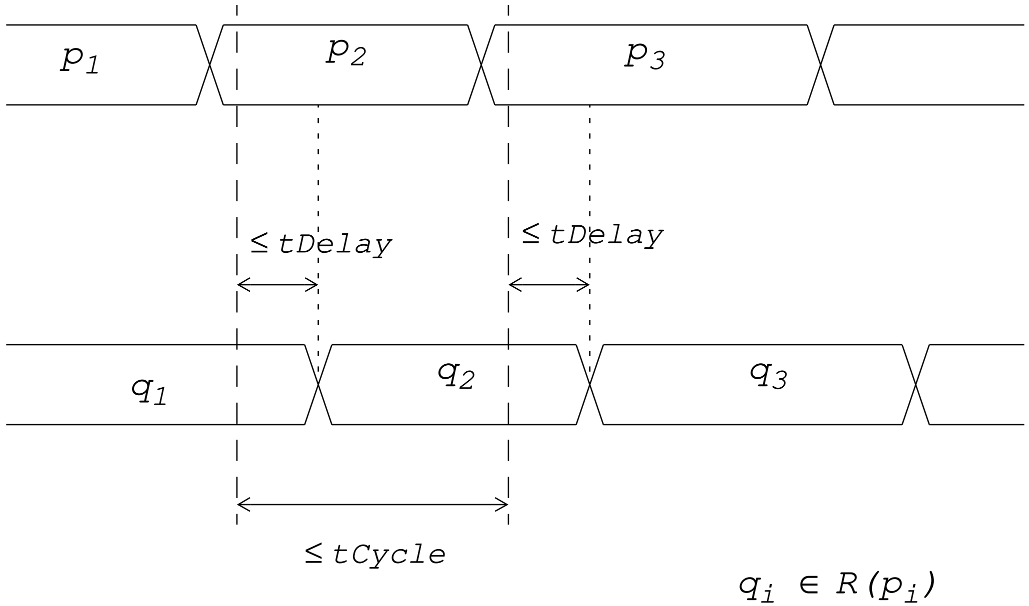
Musterdefinition (2)

Pattern *CyclicDelayedCopy (p, q, R, tCycle, tDelay)*

Intention : *p* is regularly measured and copied to *q* as *R(p)* with respect to a transport delay.

Definition : ❑ $(p \;{}_{tCycle}\!\Rightarrow^{R}{}_{tDelay}\; q)$ is defined as FIFO with time bounds

Illustration :



Transformations-Definition (1)

Transformation *SluggishSourceToCyclicDelay*

Intention : Replaces a SluggishSource by a CyclicDelay pattern.

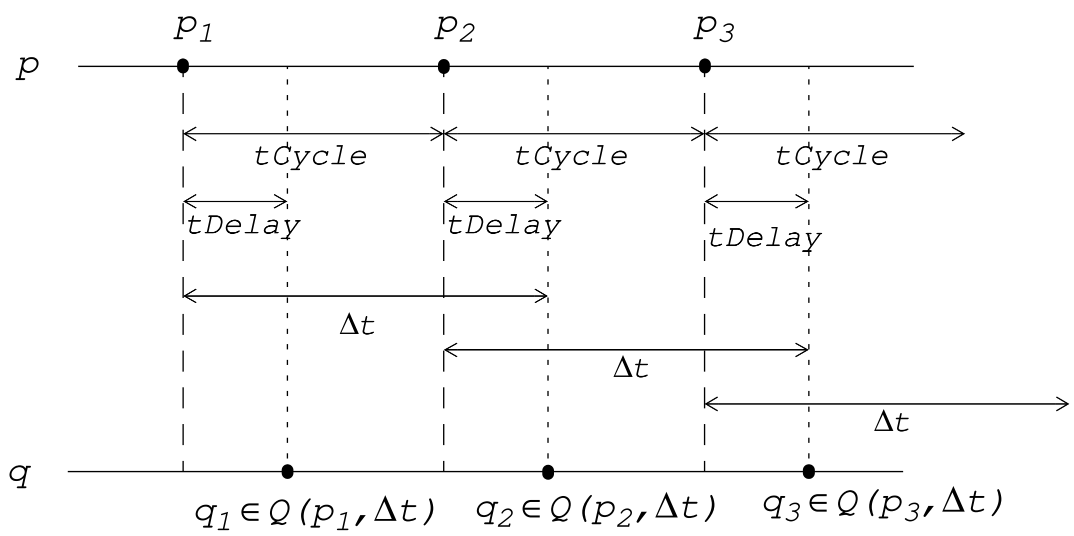
Definition : $\mathit{SluggishSourceCopy}\ (p, q, R, c) \rightarrow \mathit{CyclicDelayedCopy}\ (p, q, R', tCycle, tDelay)$

Constraints : Let $Q\ (p,\delta t) := \bigcap_{p' \in [p-c\cdot\delta t,\ p+c\cdot\delta t]} R\ (p')$

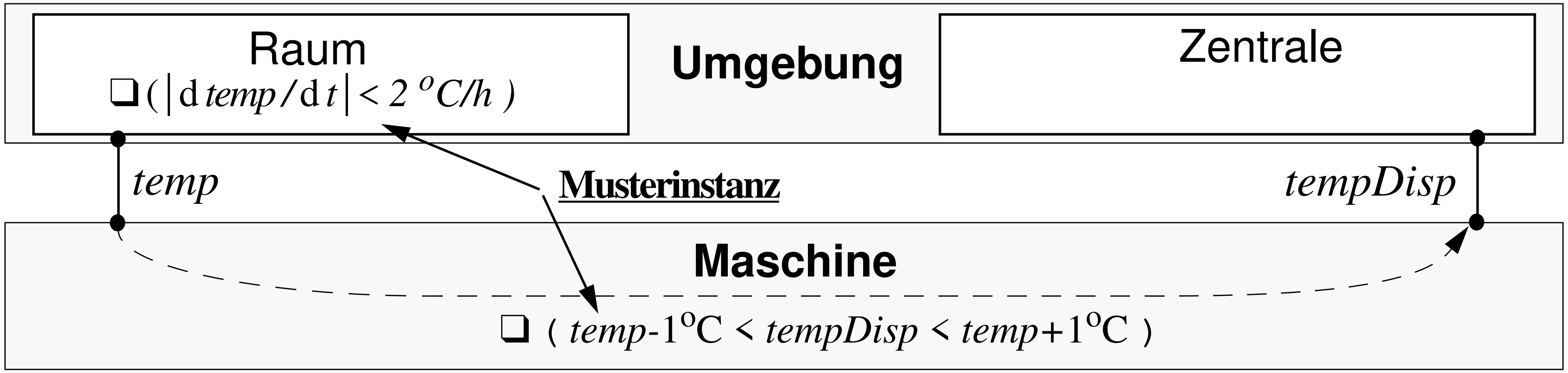
Let $\Delta t := tCycle + tDelay$

a) $\forall p \in P: R'(p) \subseteq Q\ (p,\Delta t) \neq \emptyset$

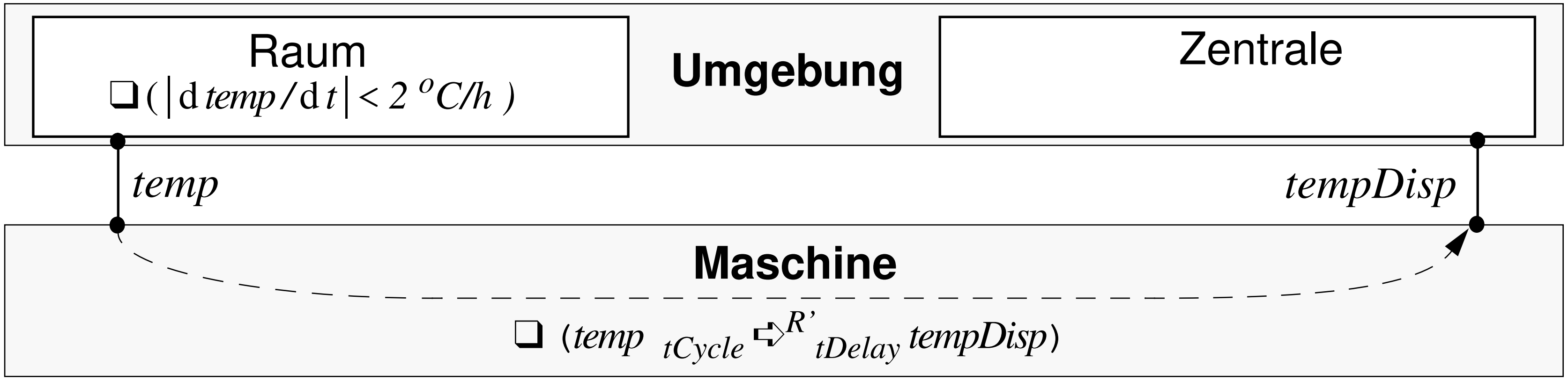
Illustration :



Beispiel: Transformation 1



Constraints: $\forall temp \in [-40{,}70] : R'(temp) \subseteq Q(temp, \Delta t) \neq \emptyset$
mit $\Delta t = tCycle + tDelay$



Transformations-Definition (2)

Transformation *DistCyclicDelay*

Intention : Distributes the cyclic delay among two components

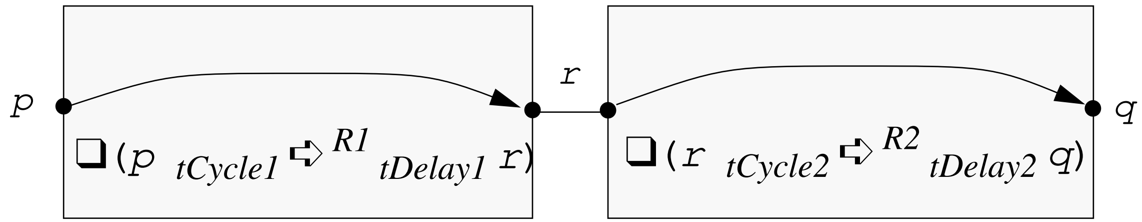
Definition : *CyclicDelayedCopy (p, q, R, tCycle, tDelay)* $\rightarrow$
CyclicDelayedCopy (p, r, R_1*, tCycle*$_1$*, tDelay*$_1$*)* $\wedge$
CyclicDelayedCopy (r, q, R_2*, tCycle*$_2$*, tDelay*$_2$*)*

Constraints : a) $R_1 \circ R_2 \subseteq R$

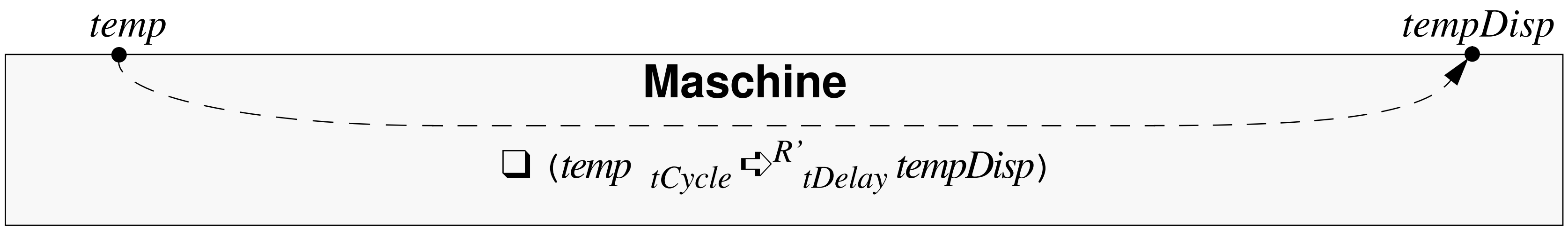
b) $tCycle_2 \leq tCycle_1 \leq tCycle$

c) $tDelay_1 + tDelay_2 \leq tDelay$

Illustration :



Beispiel: Transformation 2 und 3



Constraints: $\forall temp \in [-40{,}70]$:

$R'(temp) \subseteq Q(temp, \Delta t) \neq \emptyset$

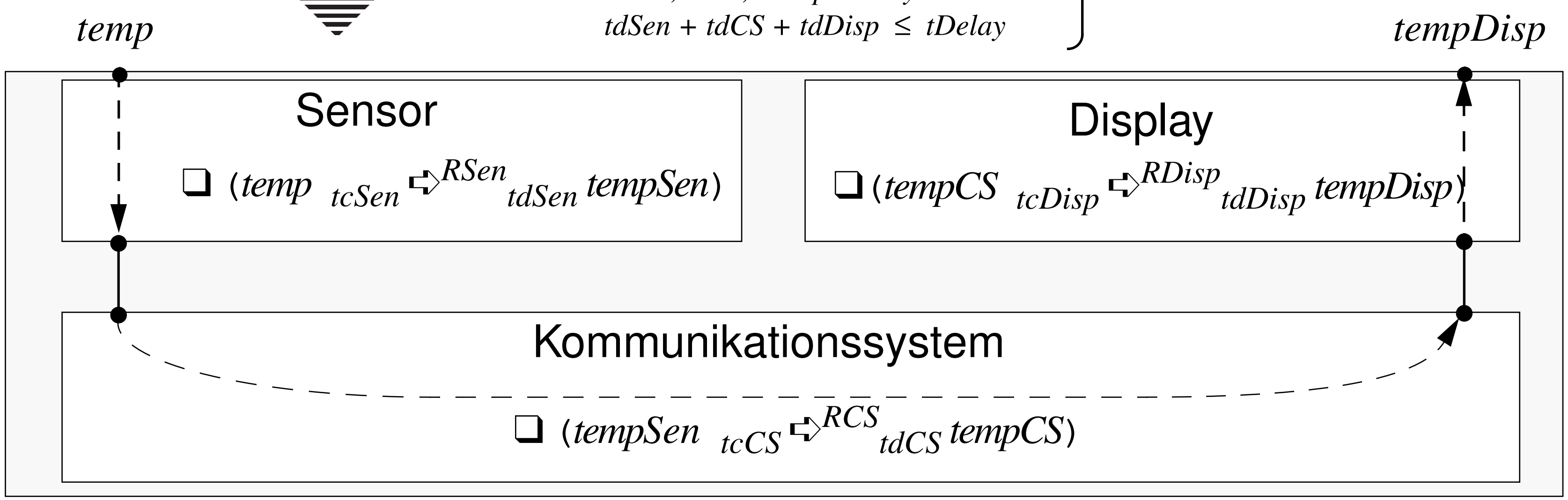
mit $\Delta t = tCycle + tDelay$

$RSen \circ RCS \circ RDisp \subseteq R'$

$tcSen, tcCS, tcDisp \leq tCycle$

$tdSen + tdCS + tdDisp \leq tDelay$

$$tcCS \leq 1619\text{ s} - tdCS$$



Beispiel: Transformation 4

temp *tempDisp*

Kommunikationssystem

$\square\ (tempSen\ {}_{tcCS}\!\Rightarrow^{id}_{tdCS}\ tempCS)$

Constraints: $tcCS \leq$ 1619 s - $tdCS$

Sen [FromSensor] [ToDisplay] Disp

Block TempTransport

$\square\ (\forall t\ :\ after\ Sen.\,?\,.tempInd\,(t)\ \ {}_{tcCS}\!\Rightarrow^{id}_{tdCS}\ \ after\ Disp.\,!\,.tempInd\,(t)\,)$

Zusammenfassung

- Gute Verfolgbarkeit der Prozeßschritte
- Strukturübergreifende Anwendbarkeit durch Muster
- Offen für Generierungsansätze
- Hohe Abstraktion von semantischen Details
  (invariant / implementierend)
- Unterstützung verschiedener Sichten

Ausblick

- **Probleme**
  - Auswertung der Constraints
  - Umfang der Abdeckung durch Muster
  - Semantik des FDT-Wechsels

- **Arbeiten**
  - Fallstudien
  - Werkzeugunterstützung (FOREST)