LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

REFLECT

A Contract-Based Approach to Adaptivity in User-Centric Pervasive Applications

Martin Wirsing, Moritz Hammer, Andreas Schroeder, Sebastian Bauer
Ludwig-Maximilians-Universität München

Maxim: "The best assistant is the one you do not notice"

- Co-drive Vehicle
  - A co-driver observes the driver carefully, watches out for obstacles, provides for a lively atmosphere on longer trips …
  - You are alone – a vehicle plays the co-driver role …

- Mood player
  - Music and lighting/illumination have significant influence on our emotional state …
  - You come home, you are alone and not in your best mood ….

- Awareness-rich Adaptive Displays
  - Advertising at TV is easy – you know who is watching and you can specifically target the audience: young, old, male/female…
  - Your display is at the street – would be good to know who is watching, how they react …

- **Goals of user-centric pervasive adaptive systems**
  - Assist us while being "invisible"
  - Adapt to the user's changing context, state and demands
  - Influence the user's environment to improve her/his well-being

- **Context:** Location, activity, etc.
  - **State**
    - Physical (e.g. movement, temperature)
    - Cognitive (e.g. high mental workload)
    - Emotional (e.g. anger, annoyance, happiness)
  - **Demands**
    - Relax when stressed during work
    - Avoid distraction while handling difficult driving situation
    - Etc.

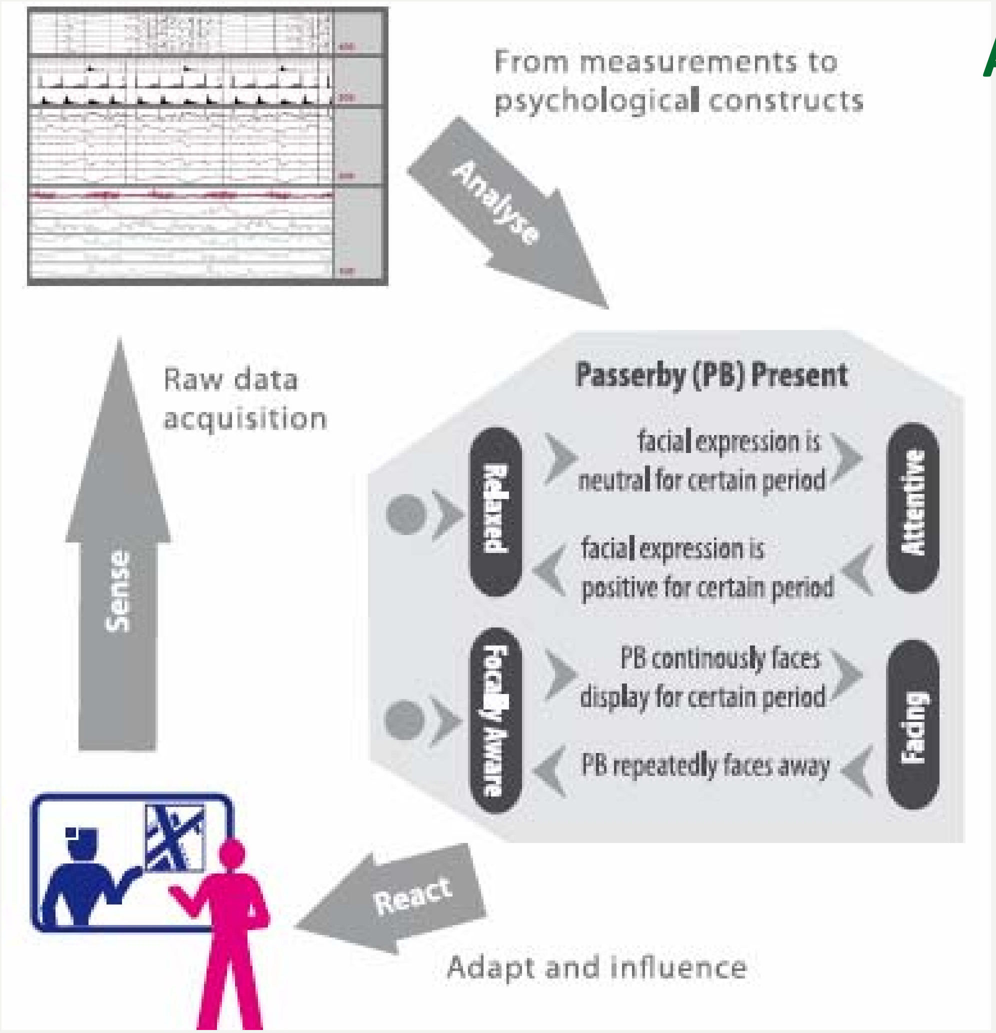
REFLECT

- **FET 7th FP EU project**
- Fraunhofer FIRST, Ferrari, Philips, LMU Munich, LJMU Liverpool, Uni Groningen, Uni Pavia, Institut Mihajlo Pupin
- develops
  - context-aware software infrastructure
  - based on adaptive software components
- studies
  - physical, cognitive, and emotional interaction

Adaptive Advertising

- Using Interactive displays (LED, rear projections)
  - Position, movement, gestures
  - Attention, emotional reaction
- Possible contents
  - Auto-active content
  - Single-viewer content
  - Multiple-viewer advertising games
  - Ambient content
- Adaptation to
  - User: number of viewers, position, attention
  - Environment: current events, context formed by the sum of people

REFLECT Approach: The Sense-Analyse-React Loop



Adaptive Advertising

- **Sensing:** The viewer's position, facial expression, gaze direction, and presence in front of the display are sensed through cameras.
- **Analysing:** The viewer's attention, emotion and interests are inferred from the sensed data.
- **Reacting:** The display alters the displayed ad content in order to react to and direct the user's interest, and emotions by adding additional effects or including content from its direct environment.

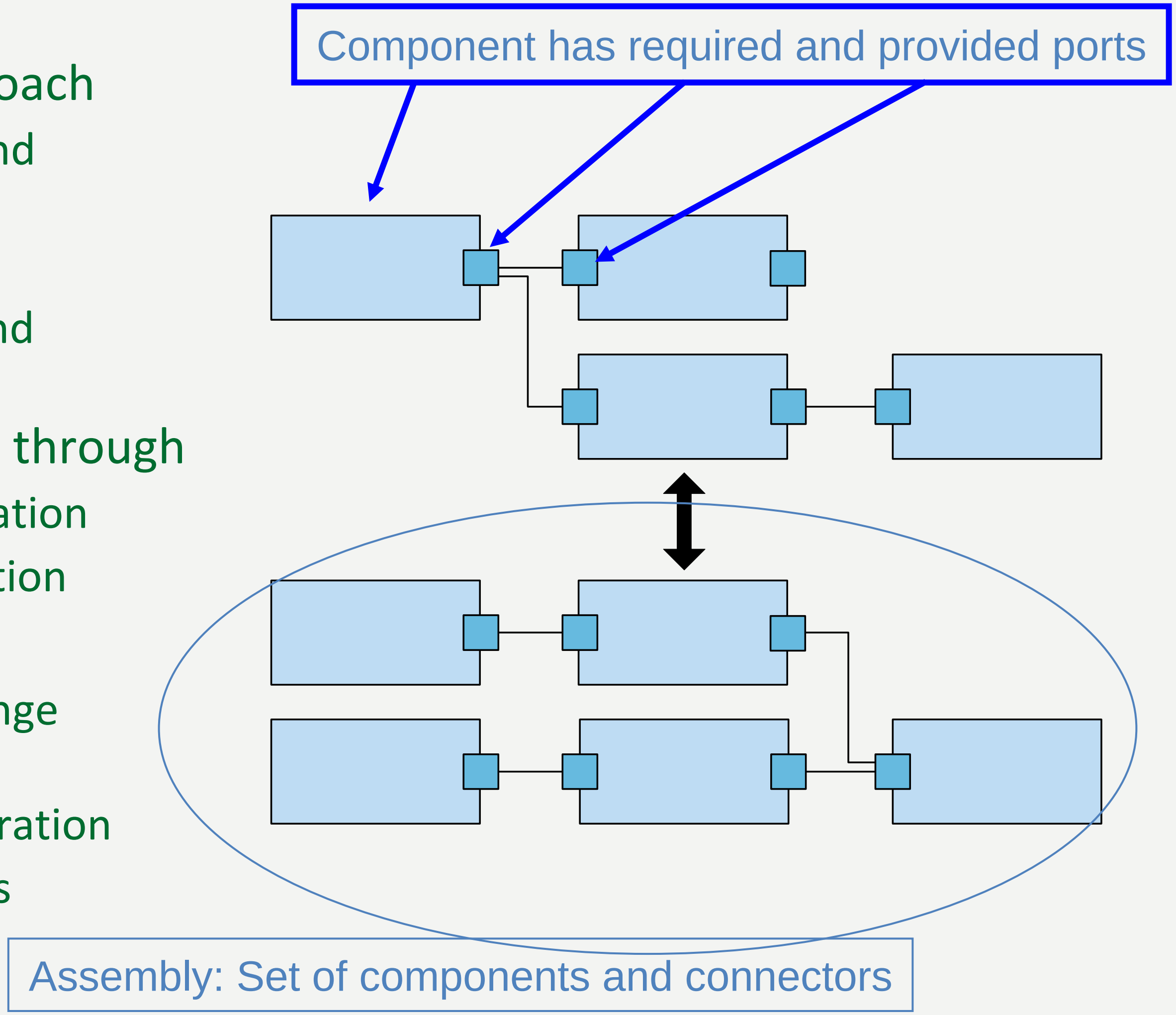
Adaptive Advertisement Film

Outline

1. Introduction to user-centric pervasive adaptive systems
2. Case Study: Adaptive Advertising
3. Assume-Guarantee Contracts for verification of systems under continuous reconfiguration
4. LTL for real-time and configurations: ReMITL
5. Specification and Verification of the case study using A/G contracts and ReMITL
6. Conclusion

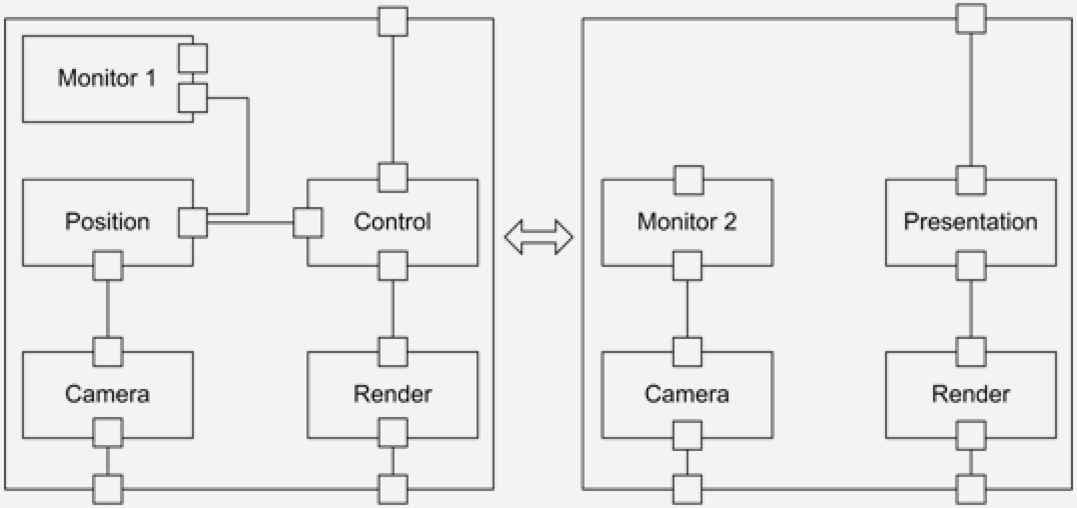
REFLECT System Design

- Component-based approach
  - Basis for modularity and verification
  - Layered middleware
  - Framework services and components
- Realization of adaptivity through
  - Parametric reconfiguration
  - Structural reconfiguration
- Easy experimentation
  - System setup and change facilitated
  - Transparent reconfiguration
  - Separation of concerns



System Design

- Generally
  - Configurations provide desired functionality only in some contexts
  - System is reconfigured to provide adequate functionality for all contexts
  - **How can we check that it really does?**
- In the example, specifically
  - Ad must show changing content
  - Ad must react to the viewer if there is someone



Verification Goals

- We want to
  - Verify correctness of the system design (configuration and reconfiguration rules)
  - Locate design problems
  - Make explicit assumptions of each configuration
  - Identify adequate reconfiguration triggers
- For this, we introduce
  - Semantic framework for system behaviours and assume/guarantee contracts
  - Language for specifying assume/guarantee contracts

Assume/Guarantee Contracts

- Signatures $Sig \ni sig = (Req, Prov)$, $Req, Prov \subseteq Nm$
- States (Prop.) $State(sig) \ni s \subseteq sig_{Req} \cup sig_{Prov}$
- Runs (RT) $Run(sig) \ni r : \mathbb{R}^{\geq 0} \to State(sig)$
- Finite variability (only non-Zeno runs)
  $$\forall t < t' \in \mathbb{R}^{\geq 0}. |\{r(t'') \mid t \leq t'' \leq t'\}| < \infty$$
- Assertions $Assert(sig) \ni A \subseteq Run(sig)$
- Req-receptivity: for all runs $r \in A$, $t \in \mathbb{R}^{\geq 0}$, $req \subseteq Req$
  there is a $r' \in A$ such that $r'(t) \cap Req = req$
  and $\forall t' < t. r(t') = r'(t')$
- Composition of assertions ($sig = \sup(sig_1, sig_2)$)
  $$A_1 : sig_1 \otimes A_2 : sig_2 = A_1 \uparrow^{sig} \cap A_2 \uparrow^{sig}$$

- Assume-Guarantee Contract $(A : sig_A, G : sig_G)$
  Contract satisfaction
  $M : sig \models (A : sig_A, G : sig_G)$ iff $M \cap A\uparrow^{sig} \subseteq G\uparrow^{sig}$
  M and G must be Req_{sig}-receptive
- Refinement
  $(A, G) \preceq (A', G')$ iff $A' \subseteq A$ and $G \subseteq G'$

Lemma 1 *(Refinement)*
if $M \models (A, G)$ and $(A, G) \preceq (A', G')$ then $M \models (A', G')$

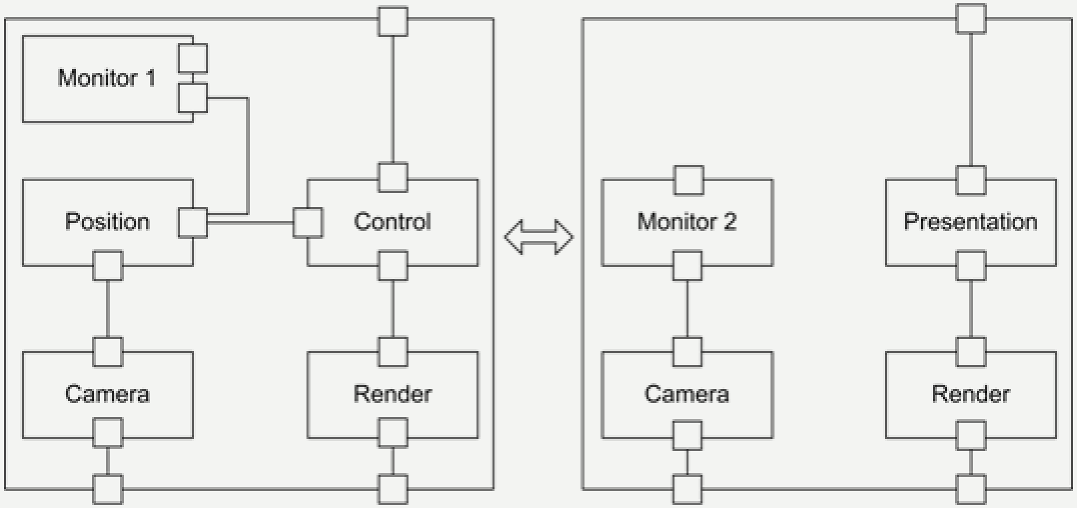
Lemma 2 *(Compositionality)*
if $M_1 \models (A_1, G_1)$ and $M_2 \models (A_2, G_2)$ then

$$M_1 \otimes M_2 \models (A, G_1 \uparrow^{sig} \cap G_2 \uparrow^{sig})$$

- where $A : sig$ such that $A \subseteq A_1 \uparrow^{sig} \cup A_2 \uparrow^{sig}$ and $A \cap G_1 \uparrow^{sig} \subseteq A_2 \uparrow^{sig}$ and $A \cap G_2 \uparrow^{sig} \subseteq A_1 \uparrow^{sig}$

Going Back to the Example

- So far, we have
  - Component-based system with contracts for each component
- Missing are
  - Support for flexible interconnection
  - A language to specify contracts



- Assembly signatures $sig_A = (Req, Prov)$
- Assembly states $State(sig_A) \ni s = (Port, Con)$
  - Ports $Port \subseteq Req \cup Prov$
  - Connectors $Con \subseteq Req \times Prov$
- Runs and assertions as before
- Assembly guarantees must be $Req \cup Prov$ -receptive
- Canonical extension of restriction, lifting, refinement and composition notions to assembly signatures
- We consider assembly guarantees only: composition with contract $(Assert(sig), G_A : sig)$

- Formulas

$$\phi ::= n.p \mid n_1.p_1 \sim n_2.p_2 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_I \phi_2$$

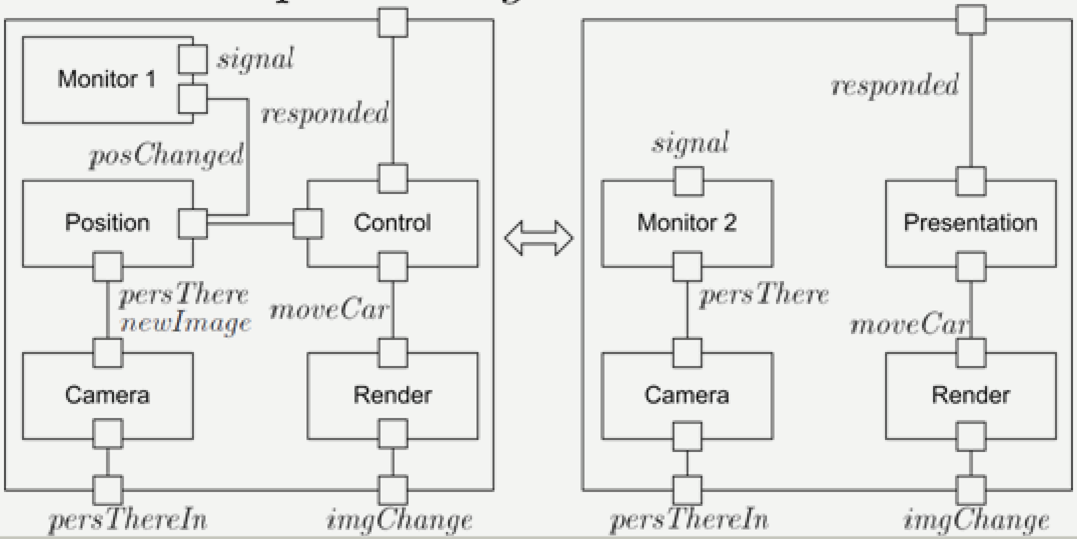
- Port: $(r,t) \models n.p$ iff $n.p \in Ports_{r(t)}$
- Connection: $(r,t) \models n_1.p_1 \sim n_2.p_2$ iff $(n_1.p_1, n_2.p_2) \in Con_{r(t)}$
- Negation: $(r,t) \models \neg\phi$ iff $(r,t) \not\models \phi$
- Conjunction: $(r,t) \models \phi_1 \wedge \phi_2$ iff $(r,t) \models \phi_1$ and $(r,t) \models \phi_2$
- Until: $(r,t) \models \phi_1 \mathcal{U}_I \phi_2$ iff $\exists t' \in t + I.\, (r,t') \models \phi_2$
  and $\forall t \leq t'' \leq t'.(r,t'') \models \phi_1$

Based on metric interval temporal logic (MITL):

Rajeev Alur, Tomás Feder and Thomas A. Henzinger. *The benefits of relaxing punctuality.* Journal of the ACM. Vol. 43, Nr. 1, 116 – 146. 1996.

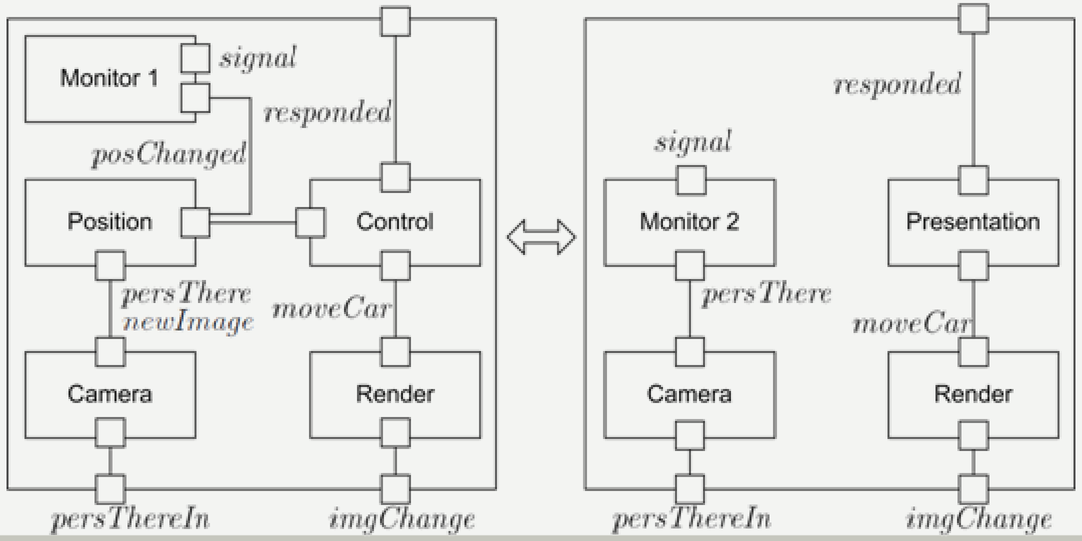
Adaptive Advertisement Specification

- Globally A: $\top$, G: $(\Box\Diamond_{[0,10]} imgChange) \wedge (\Box_{[0,1]} persThere \Rightarrow \Diamond_{[0,1]} responded)$
- Camera A: $\top$, G: $persThere \Leftrightarrow persThereIn \wedge \Diamond_{[0.02,0.03]} newImage$
- Position A: $\Diamond_{[0.01,0.10]} newImage$
  G: $posChanged \iff persThere$
- Control A: $\top$, $responded \iff posChanged \wedge moveCar \iff posChanged$



Example (2/3)

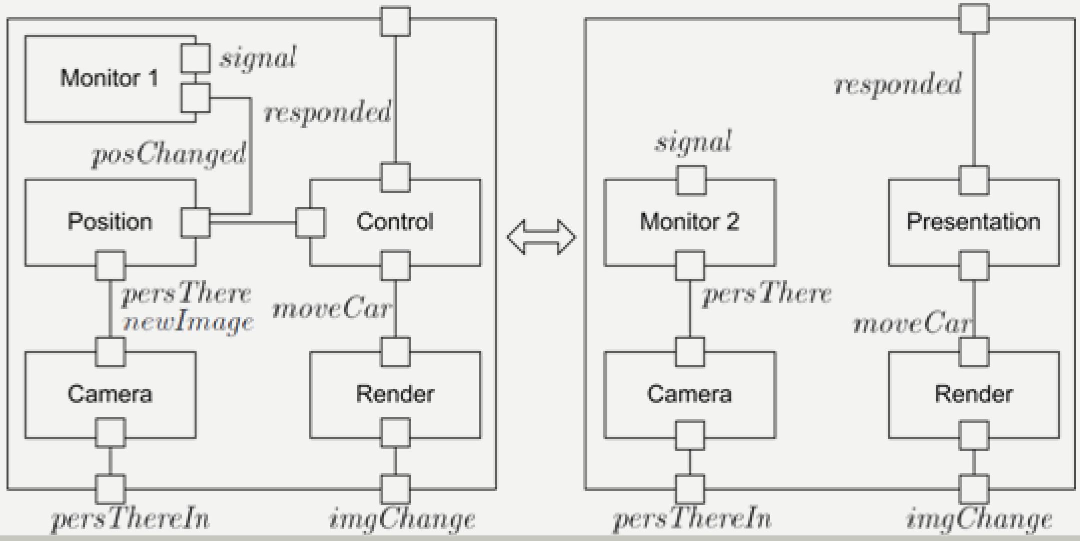
- Render $imgChange \iff moveCar$
- Presentation $moveCar \land \neg responded$
- Monitor 1 $\Box_{[0,3]} \neg posChanged \iff \Diamond_{[3,3.5]} signal$
- Monitor 2 $\Box_{[0,0.5]} persThere \iff \Diamond_{[0.5,0.7]} signal$



Example (3/3)

- Rule 1 $Monitor1.signal \Rightarrow \Diamond_{[0,0.2]} C_2$
- Rule 2 $Monitor2.signal \Rightarrow \Diamond_{[0,0.2]} C_1$
- Configuration $C_1 \vee C_2$,
  where e.g.

$$C_2 \equiv Camera.persThere \sim Monitor2.persThere \\ \wedge Presentation.moveCar \sim Render.moveCar \\ \wedge Camera.persThereIn \sim persThere \\ \wedge Render.imgChange \sim imgChange \\ \wedge Presentation.responded \sim responded$$



Proposition
Assembly satisfies the global guarantee
$(\Box\Diamond_{[0,10]} imgChange) \wedge (\Box_{[0,1]} persThere \Rightarrow \Diamond_{[0,1]} responded)$

Proof Idea (right part)

- Case 1: System in configuration 1. property holds immediately.
- Case 2: System in configuration 2.
  Assume $\Box_{[0,0.5]} persThere$ holds. Due to C_2 Monitor 2 guarantees $\Box_{[0,0.5]} persThere \iff \Diamond_{[0.5,0.7]} signal$ and $Monitor2.signal \Rightarrow \Diamond_{[0,0.2]} C_1$ holds.
  Thus, the system reconfigures itself to configuration 1 within 0.9 seconds, where the property holds immediately.

Concluding Remarks and Future Directions

- Summary
  - Introduction to user-centric pervasive adaptive systems
  - Practical approach with theoretical foundations with A/G contracts
- Future Work
  - More restrictive contract satisfaction semantics preserving refinement and compositionality:
    Always Assume/Guarantee contracts
  - Extending semantic framework and language to support **dynamic creation of component**
  - Operational model for the specification of implementations based on timed automata
  - Extension of the semantic framework to the probabilistic domain using continuous-time Markov decision processes

Thank you