



A Component Model for Dynamic Adaptive Systems

Andreas Rausch

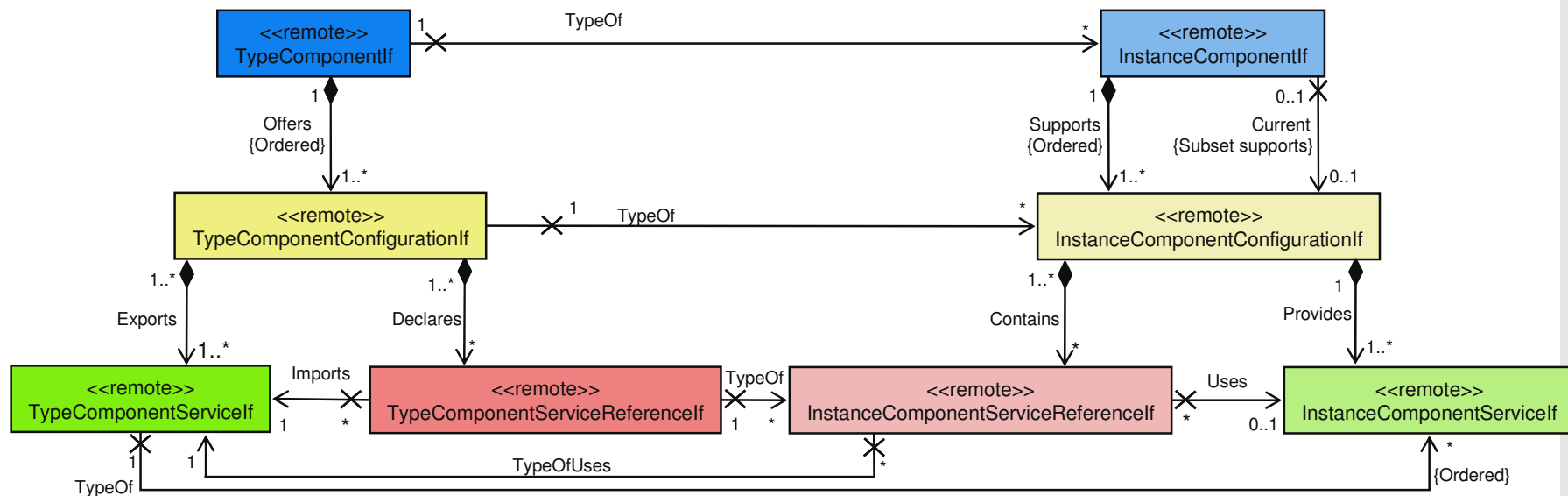
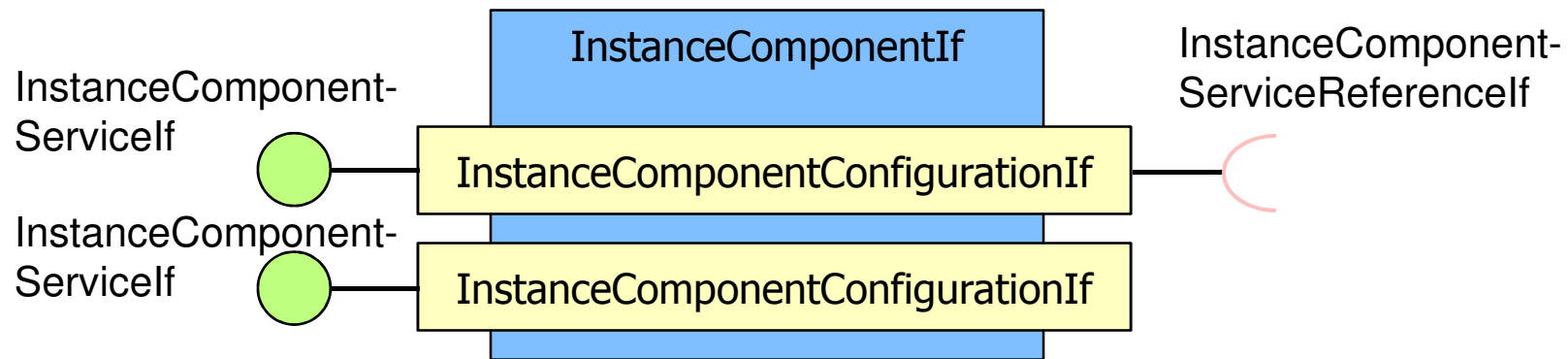
Clausthal University of Technology
Institute of Computer Sciences – Software Systems Engineering
Chair of Prof. Dr. Andreas Rausch

- Current state of software systems...
 - We are more or less able to provide complex IT systems
 - ...
- Properties of next generation software systems...
 - High networked software systems and infrastructures
 - Increasing availability of software services
 - Growing usage of personal IT-gadgets
 - More specialized functionalities and user interfaces
 - Higher degree of decentralized autonomy and controllability
 - Increased demands in software systems dependability
- A crystal ball view...



Next system generation: IT cities (ultra-large-scaled systems with high dependability requirements!

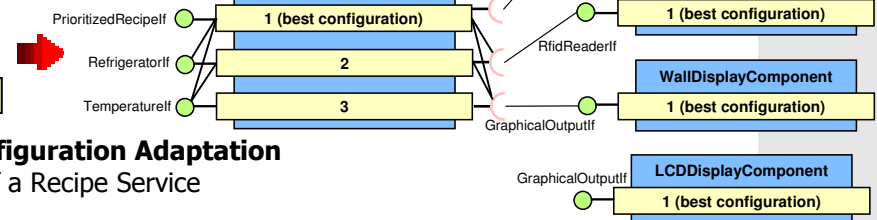
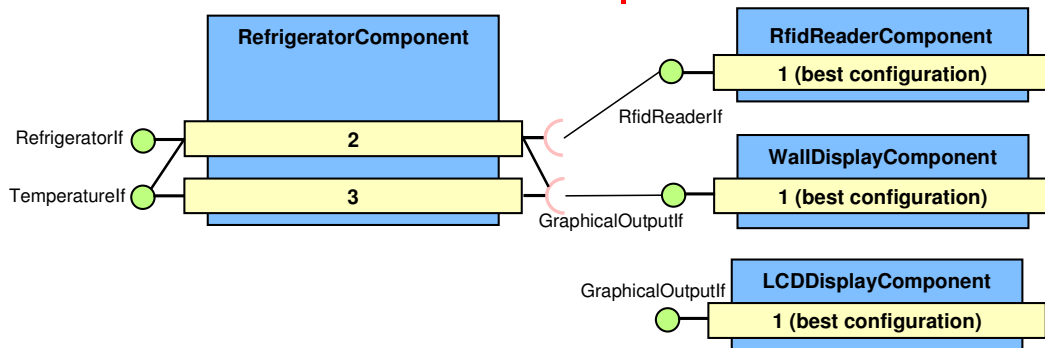
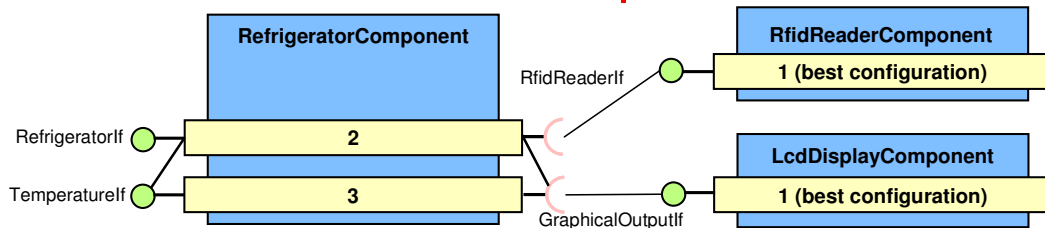
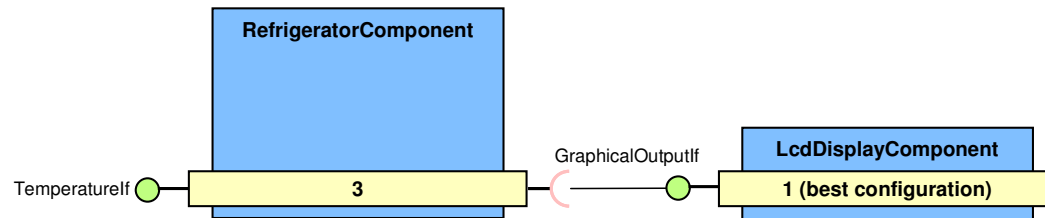
Those IT cities have to be dynamic adaptive (dynaptive) and thus an infrastructure to dynamically connect and wire the IT systems/components is required



- Normal comfort feature
Warn in case of refrigerator malfunction
- Prevent that elderly people eat spoiled food
Warn if expired food is inside the refrigerator



- Provide comfort features
Offer recipes (context-sensitive)



- Results we have reached so far
 - ☑ Formal system model for dynaptive systems
 - ☑ Middleware implementing the system model
 - ☑ Implemented demonstration prototypes proving the applicability of these concepts
- Questions emerging directly from the work presented:
 - Physical Level: Integration of additional devices and physical interfaces and communication channels
 - System Level: Integration of ad-hoc communication between smart nodes and additional protocols
 - Infrastructure Level: Integration of event based communication
 - Application Level: Sound description techniques and code generation for the components
- Questions for which the broader community should be seeking answers:
 - Formal system model: What kind of adaptation concepts exist and what are they about?
 - Description techniques: How can we describe and reason about all kind of relevant aspects of dynaptive systems on the proper abstraction level?
 - Iterative system evolution process: How can we provide and evolve ultra-large-scaled systems over time?
 - Tool support: What kind of tool support is required and how can it be provided?
 - Standard domain architectures: How can we agree on and provide standard domain architectures on an industrial accepted level?

Any Questions

