

Composition and Optimization

Morgan Ericsson¹, Welf Löwe¹, Christoph Kessler² and
Jesper Andersson¹

¹ Växjö University, Sweden

² Linköping University, Sweden

Outline

- Problem
- Definitions
- Composition
- Composition of Parallel Components
- Conclusions and Future Directions

Problem

- Developing software is not easy...
- ... even small things are difficult
- Focus on functional vs. non-functional requirements
- Increasingly complex environment and requirements

Solution?

- Reusable components
- Often helps solve the functional part
- Traditionally, little focus on the non-functional parts
- Can sometimes even make the non-functional parts worse

Motivation*

- Linné observatory in southern sweden
- Sensor network with several thousand receivers, each with “high” bandwidth (~2gbps each).
- Several disciplines, several simultaneous users
- Unprocessed data cannot be stored

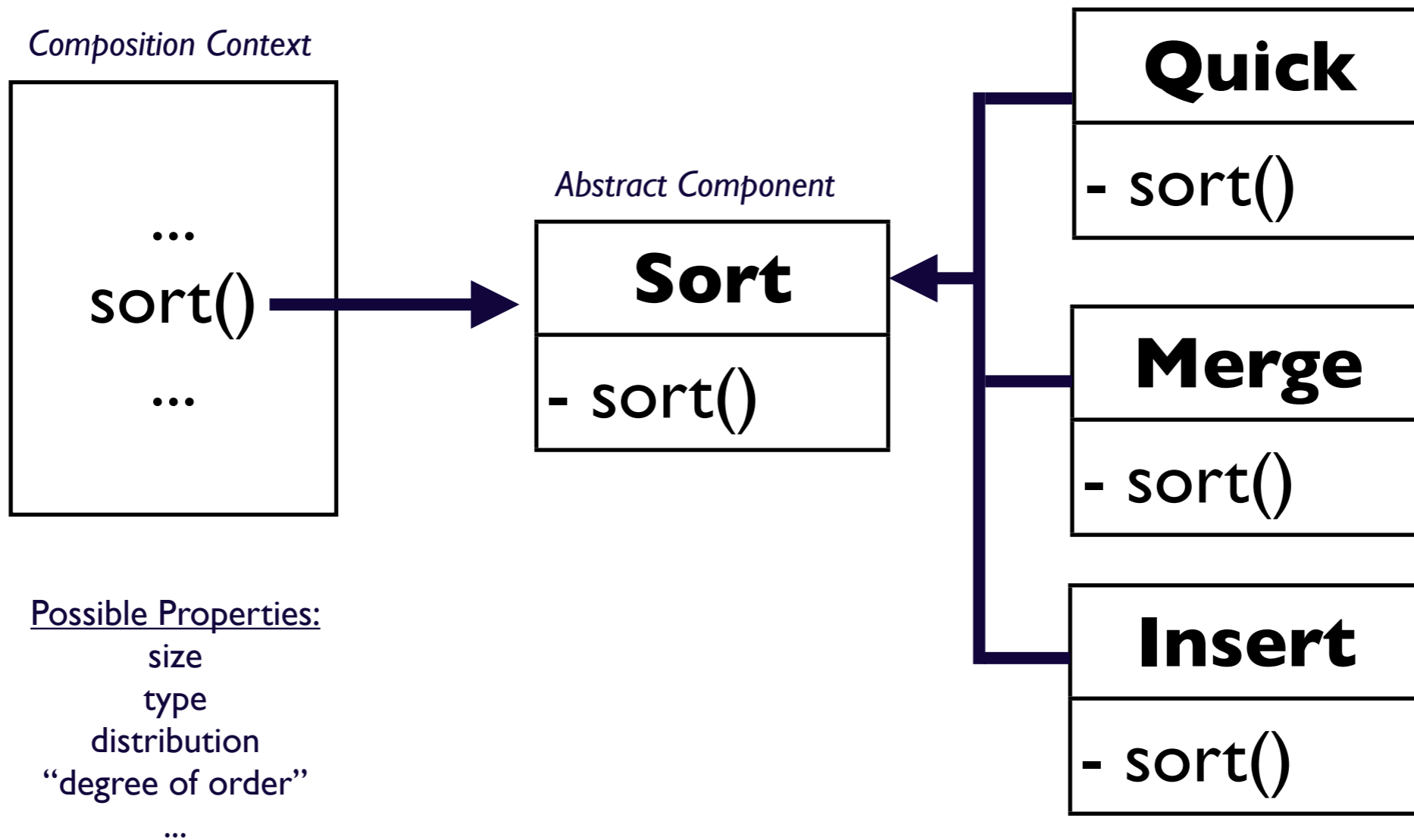
Definitions

- *A component* is a modular abstraction - a set of procedures with an interface.
Components can be *abstract*
- *Composition* is the binding of a procedure call to a procedure definition
- *A component variant* is a component substitutable for an abstract component

Definitions (cont'd)

- A *composition context* is the procedure containing the procedure call
- A composition context contains a number of properties - *context properties*

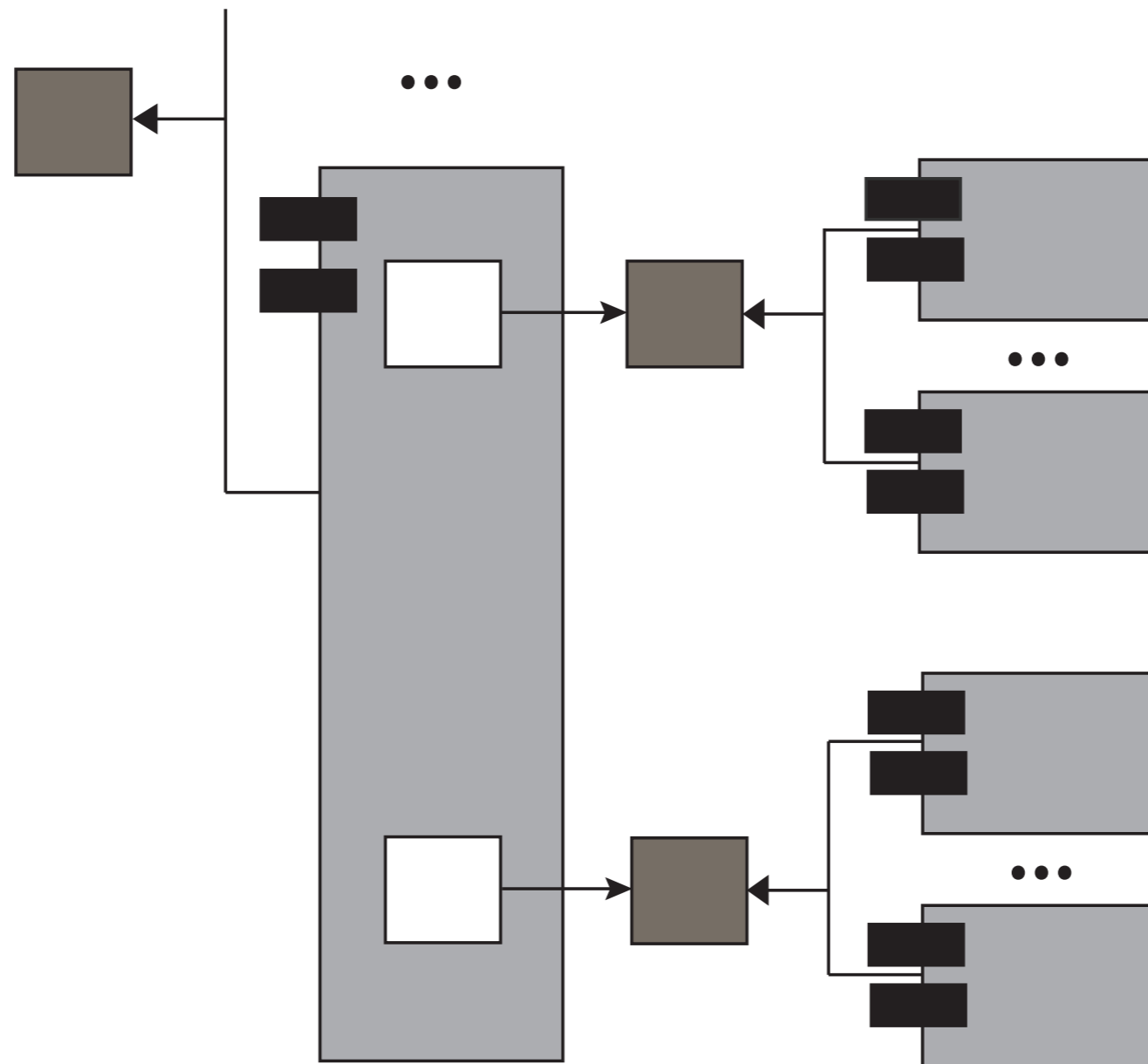
A Simple Example



Goal and Metrics

- *Composition metrics* are mappings of context properties and composition variants to numerical values
- *A composition goal* is a set of functions over the composition metrics

Recursive and Self-similar



Context-Aware Composition

- Based on the composition goal functions and the metrics
- Select the best-fit variant for each distinct property state
- Happens locally within each component
- The best-fit variants are exposed

Implementation Concerns

- Pre- and post-conditions are used to remove incorrect compositions
- Weights are generated off-line
 - Models or samples
- Composition happens when enough information is available
 - Statically or dynamically

Composition of Parallel Components

- Extend the composition to consider variant and number of processors to spend on the variant
- Can be modeled as the variant independent malleable task scheduling problem
- Store best variant together with best processor allocation and schedule

Conclusions

- Context-aware compositions addresses the issue of non-functional requirements and composition
- Software engineering approach
- Composition happens when enough information is available
- Language independent and not restricted to a single problem domain (has been used

Todo

- Use it for a larger range of goals. Currently, mostly used to minimize average case execution time.
- Use the composition technique for a “proper” development project
- Develop better tools to support users

Composition and Optimization

Morgan Ericsson¹, Welf Löwe¹, Christoph Kessler² and
Jesper Andersson¹

¹ Växjö University, Sweden

² Linköping University, Sweden