

Linux Audio Conference 2007

Open Source as a Special Kind of Component-Based System Development

Steffen Evers
TU Berlin

Berlin, 24.03.07

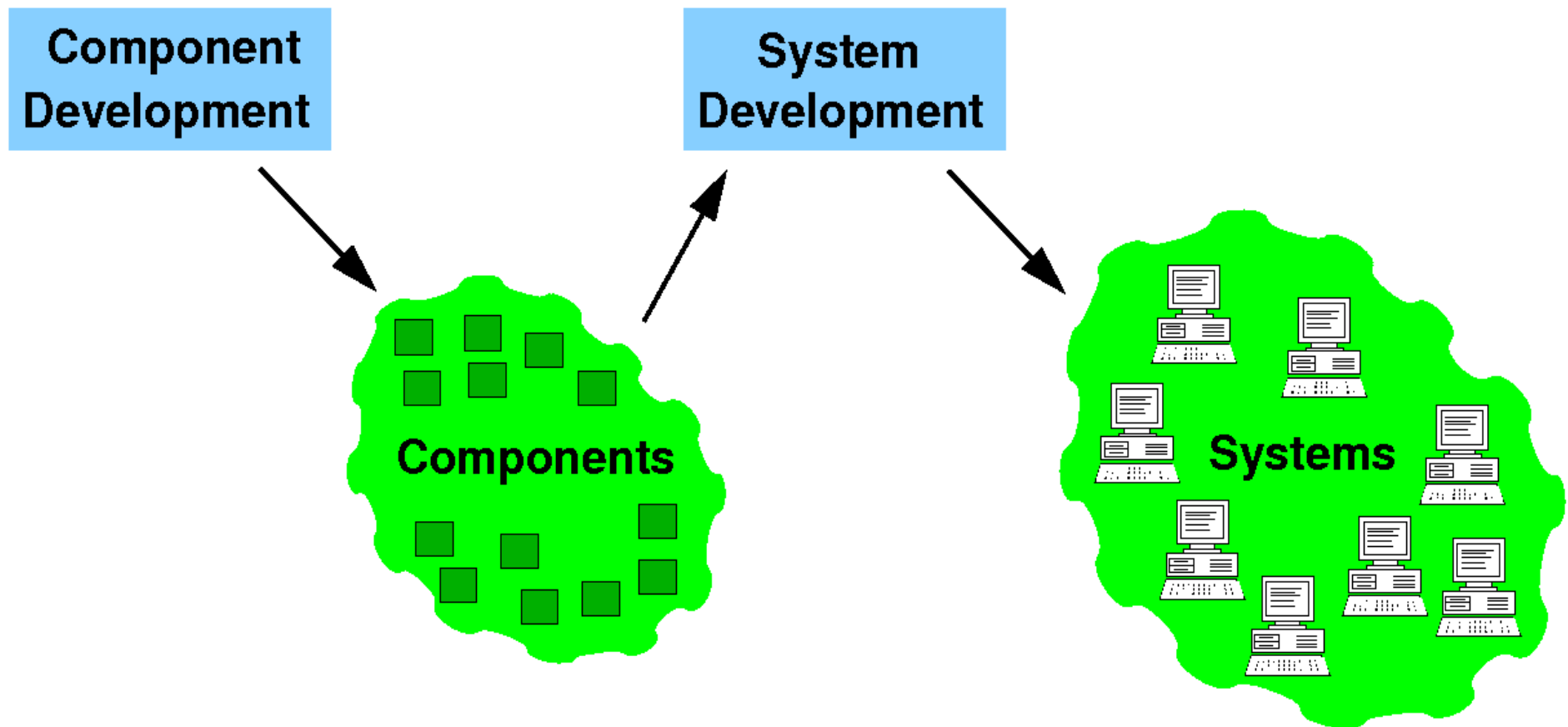
Component-Based System Development (CBSD)

origin is McIlroy's paper in 1968
“Mass-Produced Software Components”

two distinct activities:

- component development
- system development

Two Main Activities in CBSD



Some CBSD Technologies

- CORBA
- Enterprise Java Beans
- .NET
- Web Services

Software Reuse

The major motivation behind
component-based development
is *software reuse*

Component Access & Types of Software Reuse

type	use	study	adjust
black box	yes	no	no
glass box	yes	yes	no
gray box	yes	partial	partial
white box	yes	yes	yes

Context Handling Dilemma of Components

knowing how the component will be used ...

- focused time frame is the future
- users choose a component *after* completion
- big number of different contexts
- context is in a constant change
- context contains unknown inconsistencies

Components Context

developers have to consider several context areas of a component:

- socio-technical system
- hardware
- co-existing components
- development processes
- tools and infrastructure

CBSD Hypotheses

- CBSD has been considered a promising approach since 1968. However, the lack of an efficient component development model prevented its propagation.
- The essential problem of component development is the context handling.

Open Source Hypotheses

- The Open Source community has developed a suitable component development model over time.
- Open Source software development uses an unusual method for context handling with good results.

Open Source Definition

conditions on the attached *license*

derived from Debian FSG which are derived from Free Software Definition:

freedom to ...

- use
- study
- adjust
- redistribute

What Is The *Essence* of the Open Source Phenomenon?

- license
- software
- business model
- software development model
- ideology
- method of cooperation
- group of people
- infrastructure

Finding the Right Perspective

- software development models
- theory of cognition
- complex adaptive systems
- sociology
- network theory
- project management
- theory of evolution

Open Source Development as CBSD

CBSD	OSD
component	package
system	computer
component development	open source projects
system development	distribution development & computer administration

Reuse in Open Source

Freedom to use, study, adjust, redistribute allows white box reuse.

But what types are actually used?

Component Access & Types of Software Reuse

type	use	study	adjust
black box	yes	no	no
glass box	yes	yes	no
gray box	yes	partial	partial
white box	yes	yes	yes

Reuse in Open Source

More access means more work.

As people are lazy they try reuse in the following order:

1. black
2. glass
3. gray
4. white

Reuse and Open Source Roles

component user (system development):

- black box: regular user
- glass box: expert user

component developer (component development):

- gray box: casual developer
- white box: core developer

Reuse and Feedback

reuse	feedback description
black box	basic report with behavior description
glass box	report with additional analysis of internals
gray box	minor internal changes
white box	interface adjustments and fundamental modification

Context Handling for Open Source Components

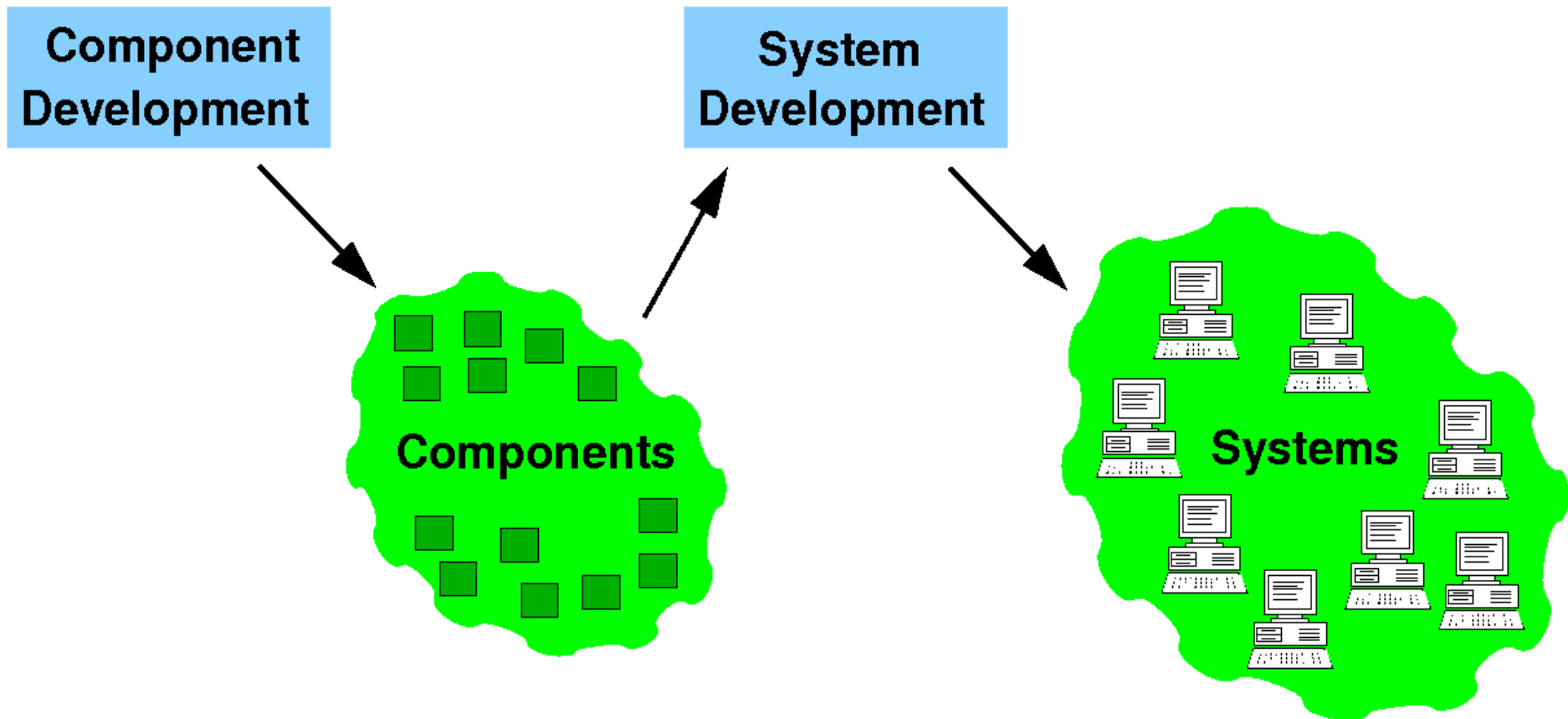
- reuse approach
- few explicit requirements
- developers have high “artistic freedom”
- developers implicitly incorporate their views of the context
- reports hint to alternative views
- code contributions incorporate different views
- project leader has to find balance between generality and accuracy

OSS Context Specification

Show Me the Code!

- process model
- contains all relevant similarities
- only valid specification
- fusion with requirement specification (like MDA)
- perceptible result
- no transformation processes
- no gap between abstraction and accuracy

Two Coupled Processes with Dependent Domains



Summary

- CBSD is an emerging approach
- component development requires handling complex requirements
- the open source community has developed promising processes to handle such circumstances
- the special kind of context handling is an essential part of it
- freedom seems to be a precondition for their method

The End

Thank you for your attention.