

Characterization of Components

Jun Han

Peninsula School of Computing & Info. Tech.
Monash University, Melbourne, Australia

introduction

- *what is a component?*
different people have different answers ... or
“*I can feel it, but cannot describe it ...*”
- *why should we characterize components?*
 - clear understanding and easy comprehension – for component (re)use in system composition
 - reasoning about components and composed systems
 - effective management and development of components and systems
 - tool and language support
 - ...
- *how?:*
“objects” “architectural components” “OO characterization”

objects

- *OOA*:

object = attributes + operations

- *OOAD*:

object = attributes + operations +
sequencing ... + interactions/scenarios

– richer than *OOA*: state transitions, interactions ...

architectural components

- software architectures and their description languages (SADLs):
 - focus: *system* architecture
 - diverse views of components: not clearly characterized
 - characterization(?): interface, types, semantics, constraints, evolution

- an OO view of existing features of components:

ArchComponent = attributes + operations +
constraints/sequencing/interactions

- external *observables* ...
- constraints: internal invariants vs. external restrictions
- constraints vs. sequencing vs. interactions ...

OO characterization of components

- *attributes*: (observable) structural elements
- *structural constraints*: component invariants ...
- *operations*: behavioural elements, means of interaction with context
- *operational constraints*: permissible operation patterns/sequences – (external) interaction restrictions ...
- *events*: for reactive control (eg, in JavaBeans)
- *multi-interfaces*: multiple roles in a use context; each interface contains some or all of the above
- *scenarios*: a component may be used in different contexts with different interface settings ... (dynamic component adaptation)
- *ilities*: characterization of non-functional properties (eg, security, performance and reliability)

summary

Component = attributes + structural constraints +
operations + operational constraints +
events + multi-interfaces*scenarios + illities

- existing SADLs: not looking at components in such a way
 - existing component models (COM, JavaBeans, CORBA): too simplistic
- components should grow out-of and above current objects!*
- much needs to be done ...