

A Language Approach to Discrete Control in Computing

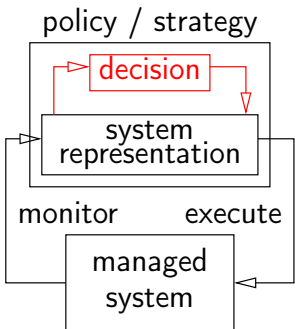
Eric Rutten & a.

INRIA / LIG

April, 2010 — FeBID @ EuroSys, Paris

Control Techniques for Adaptive Systems

- Adaptation control as a closed control loop

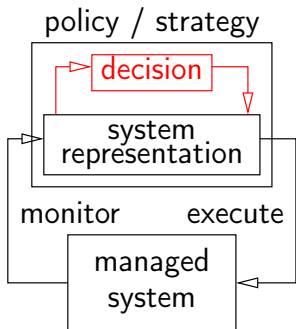


Discrete Control Techniques for Adaptive Systems

Use of **Discrete Event Systems** and supervisory control:

Petri nets, language theory (R&W), automata (synchronous)

- Adaptation control as a closed control loop

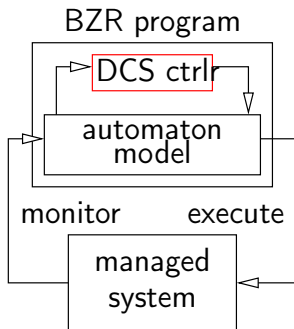
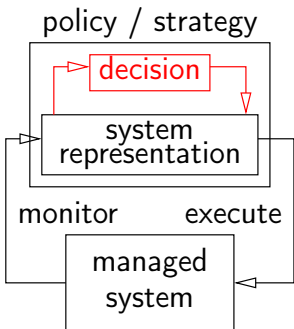


Discrete Control Techniques for Adaptive Systems

Use of **Discrete Event Systems** and supervisory control:

Petri nets, language theory (R&W), automata (synchronous)

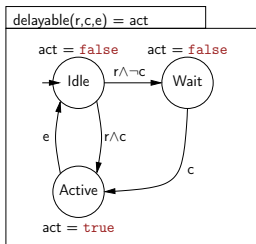
- Adaptation control as a closed control loop
- BZR programming language, and Discrete Controller Synthesis to compute the decision component (controller)



Examples of discrete modes

state \leftrightarrow configuration , level of consumption/quality, ...

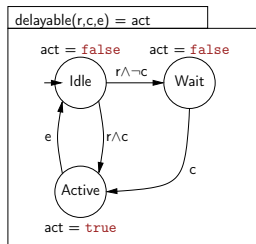
- computation task control



Examples of discrete modes

state \leftrightarrow configuration , level of consumption/quality, ...

- computation task control

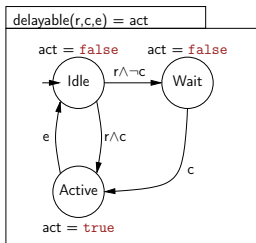


- modes: algorithm variants for a functionality (resource, QoS)

Examples of discrete modes

state \leftrightarrow configuration , level of consumption/quality, ...

- computation task control

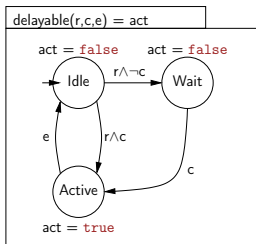


- modes: algorithm variants for a functionality (resource, QoS)
- placement and migration : task T_i on processor/core P_j
 resource budgeting: core taken for other application
 fault tolerance : migration/rollback upon processor failure

Examples of discrete modes

state \leftrightarrow configuration , level of consumption/quality, ...

- computation task control



- modes: algorithm variants for a functionality (resource, QoS)
- placement and migration : task T_i on processor/core P_j
 resource budgeting: core taken for other application
 fault tolerance : migration/rollback upon processor failure
- architecture control: frequency, DVS, stand-by in MPSoC

Discrete controller synthesis: principle

Goal

Enforcing a temporal property Φ on a system (on which Φ does not a priori hold)

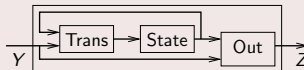
Discrete controller synthesis: principle

Goal

Enforcing a temporal property Φ on a system (on which Φ does not a priori hold)

Principle (on implicit equational representation)

State memory
Trans transition function
Out output function



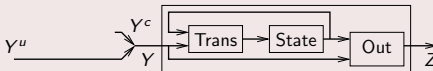
Discrete controller synthesis: principle

Goal

Enforcing a temporal property Φ on a system (on which Φ does not a priori hold)

Principle (on implicit equational representation)

State memory
Trans transition function
Out output function



- Partition of inputs into controllable (Y^c) and uncontrollable (Y^u) inputs

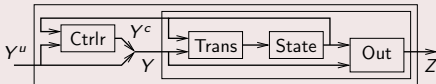
Discrete controller synthesis: principle

Goal

Enforcing a temporal property Φ on a system (on which Φ does not a priori hold)

Principle (on implicit equational representation)

State memory
Trans transition function
Out output function



- Partition of inputs into controllable (Y^c) and uncontrollable (Y^u) inputs
- Computation of a controller, *maximally permissive*, such as the controlled system satisfies Φ
- tool: `sigali` (H. Marchand, INRIA Rennes)

BZR: contracts and DCS

$f(x_1, \dots, x_n) = (y_1, \dots, y_p)$
$e_A \Rightarrow e_G$
with c_1, \dots, c_q
$y_1 = f_1(x_1, \dots, x_n, c_1, \dots, c_q)$ \dots $y_p = f_p(x_1, \dots, x_n, c_1, \dots, c_q)$

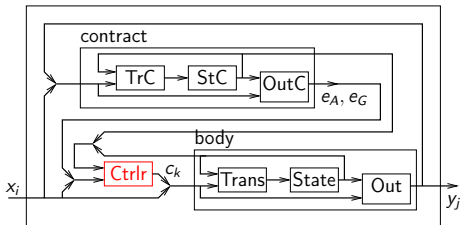
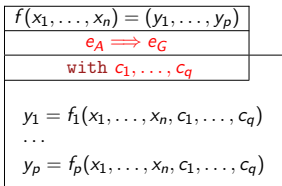
- built on top of heptagon synchronous nodes (M. Pouzet e.a.)

BZR: contracts and DCS

$f(x_1, \dots, x_n) = (y_1, \dots, y_p)$
$e_A \implies e_G$
with c_1, \dots, c_q
$y_1 = f_1(x_1, \dots, x_n, c_1, \dots, c_q)$ \dots $y_p = f_p(x_1, \dots, x_n, c_1, \dots, c_q)$

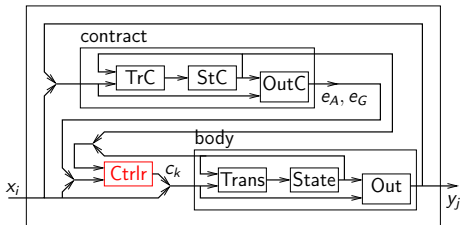
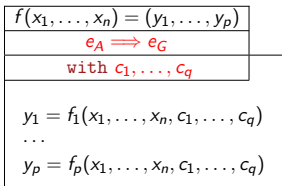
- built on top of heptagon synchronous nodes (M. Pouzet e.a.)
- contract construct :
 - **assuming** e_A (on the environment), **enforce objective** e_G
 - by constraining the additional **controllable variables**
 c_1, \dots, c_q local to the component (**with**)

BZR: contracts and DCS



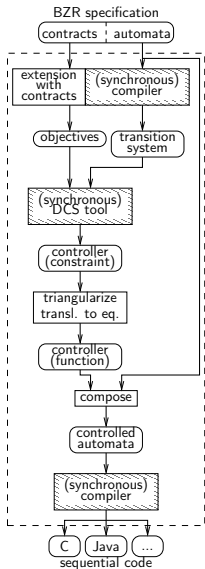
- built on top of heptagon synchronous nodes (M. Pouzet e.a.)
- contract construct :
 - assuming e_A (on the environment), enforce objective e_G
 - by constraining the additional **controllable variables**
 c_1, \dots, c_q local to the component (**with**)
- encoded as a **DCS problem (invariance)**
 computes a local controller for each component

BZR: contracts and DCS



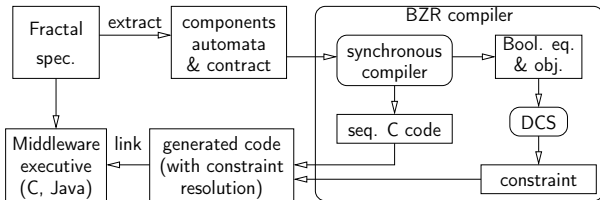
- built on top of heptagon synchronous nodes (M. Pouzet e.a.)
- contract construct :
 - assuming e_A (on the environment), enforce objective e_G
 - by constraining the additional **controllable variables**
 c_1, \dots, c_q local to the component (**with**)
- encoded as a **DCS problem (invariance)**
 computes a local controller for each component

Implementation



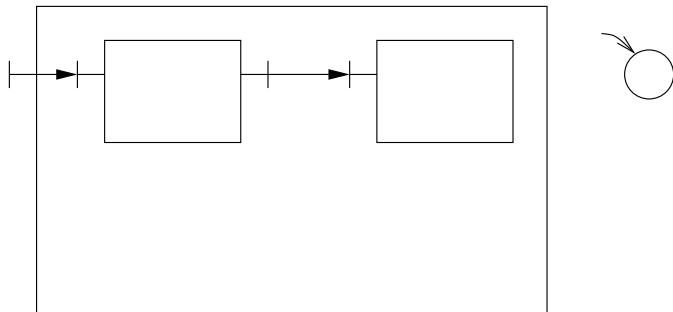
development process:

integration in adaptive system:



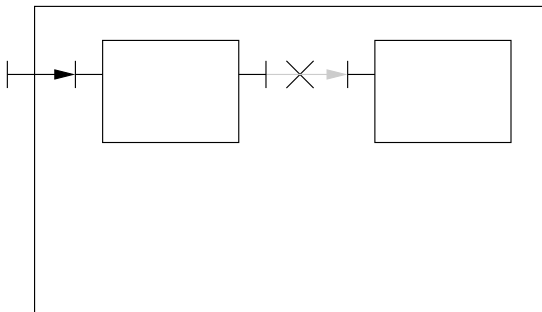
Integration in Fractal component-based design

- Configurations modelled by states



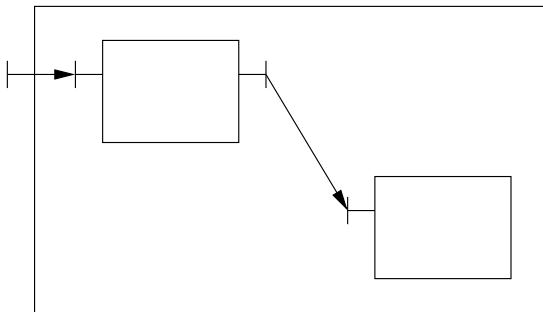
Integration in Fractal component-based design

- Configurations modelled by states
- Reconfigurations by transitions



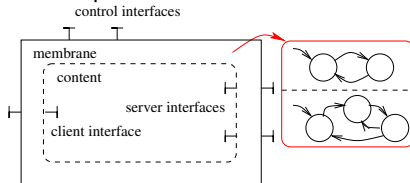
Integration in Fractal component-based design

- Configurations modelled by states
- Reconfigurations by transitions



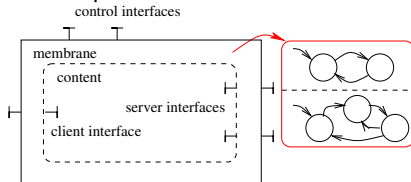
Structural behavior in Fractal

- Control associated to component
 - subcomponents controllers

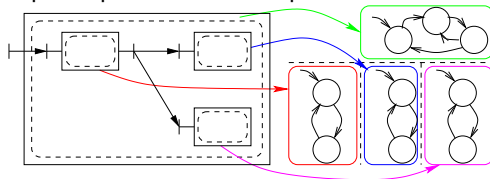


Structural behavior in Fractal

- Control associated to component
 - subcomponents controllers

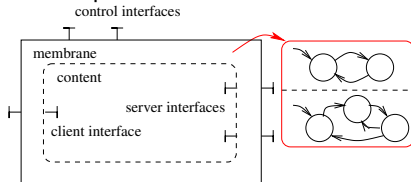


- top component-level composition

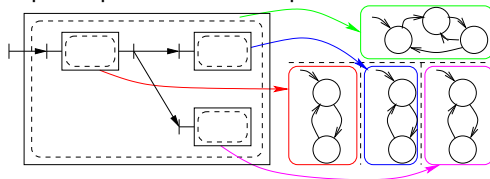


Structural behavior in Fractal

- Control associated to component
 - subcomponents controllers

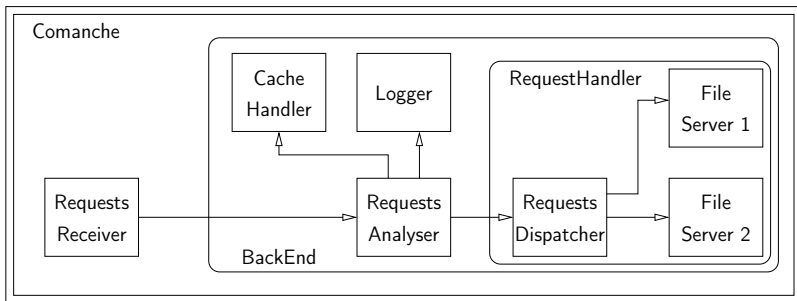


- top component-level composition



Example of a HTTP server

- variation on the Comanche HTTP server in Fractal tutorials¹
- dynamical reconfiguration: *File Servers*, *CacheHandler* and *Logger* components can be activated or deactivated

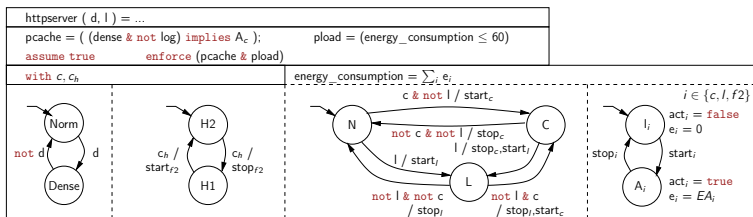


¹<http://fractal.ow2.org/tutorial/>

Components controllers & contract

Requirements: adaptation policy:

- 1 activate *CacheHandler* when high number of similar requests;
- 2 adapt number of deployed file servers w.r.t to overall load;
- 3 a logging request by the system admin. should not be denied;
- 4 logging and cache handling should be exclusive.



Ongoing work

- **data-intensive computing**, MARTE (OMG profile)
ANR project FAMOUS: reconfigurable circuits design,
validation and implementation (FPGA)
RecoMARTE; reactive control of reconfigurations

Ongoing work

- **data-intensive computing**, MARTE (OMG profile)
ANR project FAMOUS: reconfigurable circuits design,
validation and implementation (FPGA)
RecoMARTE; reactive control of reconfigurations
- **component-based middleware**, Fractal
Minalogic project MIND: new version of Fractal
control: integration in ADL, Alloy model, video case-study

Ongoing work

- **data-intensive computing**, MARTE (OMG profile)
ANR project FAMOUS: reconfigurable circuits design,
validation and implementation (FPGA)
RecoMARTE; reactive control of reconfigurations
- **component-based middleware**, Fractal
Minalogic project MIND: new version of Fractal
control: integration in ADL, Alloy model, video case-study
- **control systems programming**, Orccad
integration of controllers in the real-time executive

Ongoing work

- **data-intensive computing**, MARTE (OMG profile)
ANR project FAMOUS: reconfigurable circuits design,
validation and implementation (FPGA)
RecoMARTE; reactive control of reconfigurations
- **component-based middleware**, Fractal
Minalogic project MIND: new version of Fractal
control: integration in ADL, Alloy model, video case-study
- **control systems programming**, Orccad
integration of controllers in the real-time executive
- **virtual machines**, Synergy
control of administration loops, and their coordination
in event-based engine framework

Ongoing work

- **data-intensive computing**, MARTE (OMG profile)
ANR project FAMOUS: reconfigurable circuits design,
validation and implementation (FPGA)
RecoMARTE; reactive control of reconfigurations
- **component-based middleware**, Fractal
Minalogic project MIND: new version of Fractal
control: integration in ADL, Alloy model, video case-study
- **control systems programming**, Orccad
integration of controllers in the real-time executive
- **virtual machines**², Synergy
control of administration loops, and their coordination
in event-based engine framework

²post-doc position open on this!

Perspectives

- **more integration**
designing **controllable** runtime executives

Perspectives

- **more integration**
designing **controllable** runtime executives
- **more elaborate models**
finer grain, e.g. fault tolerance [FMSD09]

Perspectives

- **more integration**
designing **controllable** runtime executives
- **more elaborate models**
finer grain, e.g. fault tolerance [FMSD09]
- **more DCS**
costs on paths, reachability, dynamical controllers

Perspectives

- **more integration**
designing **controllable** runtime executives
- **more elaborate models**
finer grain, e.g. fault tolerance [FMSD09]
- **more DCS**
costs on paths, reachability, dynamical controllers
- **more applications**
e.g. GreenIT (sustainable)
Green4IT: energy/power consumption models
for sensor networks, servers and parallel computing
IT4Green: applying control programming techniques
to program e.g., "intelligent" buildings