

kiltera: a language for concurrent, interacting, mobile, timed systems

Ernesto Posse

Modelling, Simulation and Design Lab
School of Computer Science
McGill University

August 28, 2006

Outline

- 1 Introduction
- 2 Overview of kiltera processes
- 3 Future work

Introduction

- kiltera: a formalism for describing systems which are:

Introduction

- kiltera: a formalism for describing systems which are:
 - dynamic

Introduction

- kiltera: a formalism for describing systems which are:
 - dynamic
 - concurrent

Introduction

- kiltera: a formalism for describing systems which are:
 - dynamic
 - concurrent
 - interactive

Introduction

- kiltera: a formalism for describing systems which are:
 - dynamic
 - concurrent
 - interactive
 - timed

Introduction

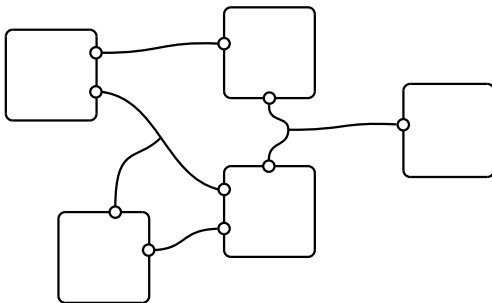
- kiltera: a formalism for describing systems which are:
 - dynamic
 - concurrent
 - interactive
 - timed
 - discrete-event

Introduction

- kiltera: a formalism for describing systems which are:
 - dynamic
 - concurrent
 - interactive
 - timed
 - discrete-event
 - mobile

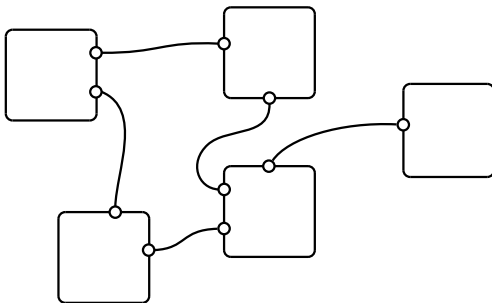
Overview of kiltera processes

A network of processes:



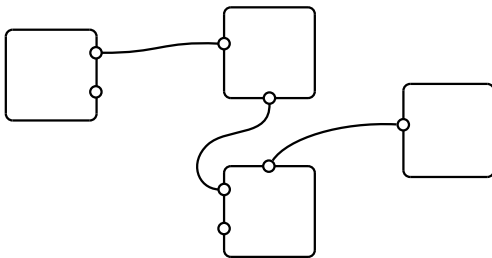
Overview of kiltera processes

Mobility: channels can move



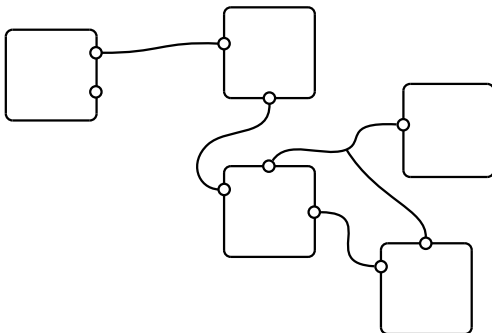
Overview of kiltera processes

Mobility: components can disappear



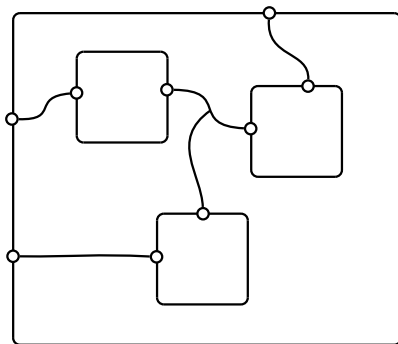
Overview of kiltera processes

Mobility: components can be created



Overview of kiltera processes

Nesting: a network of processes is a process



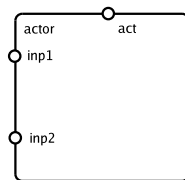
Overview of kiltera processes

- Time
- Global clock
- Time-base: real numbers
- Time consistency
- Processes can observe and measure the passage of time between events
- Process behaviour can be determined by the timing of events
- Processes can schedule events in the future

Overview of kiltera processes

Process definitions

```
process actor[inp1, inp2, act]:  
  <body>
```

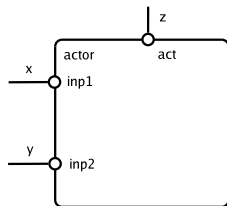


This defines a class of processes with a given interface.

Overview of kiltera processes

Process instantiation

```
process actor[inp1, inp2, act]:  
  <body>  
in  
  ...  
  actor[x,y,z]  
  ...
```



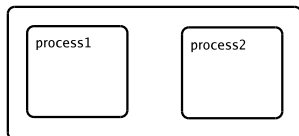
Overview of kiltera processes

Parallel processes

`par`

`<process1>`

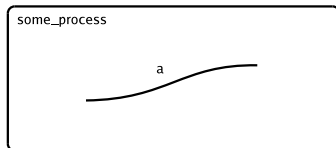
`<process2>`



Overview of kiltera processes

Channel declarations

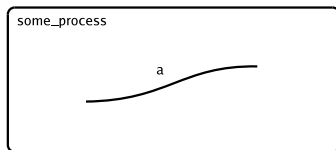
```
channel a in  
  <some_process>
```



Overview of kiltera processes

Channel declarations

```
async channel a in  
  <some_process>
```

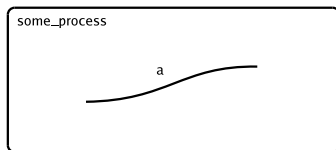


Creates a channel for asynchronous communication (unbounded buffer.) This is the default.

Overview of kiltera processes

Channel declarations

```
sync channel a in  
  <some_process>
```



Creates a channel for synchronous communication (rendez-vous.)

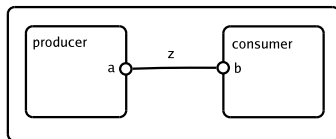
Overview of kiltera processes

Networks

```

process producer[a] :
  <body1>
process consumer[b] :
  <body2>
in
  channel z in
    par
      producer[z]
      consumer[z]

```



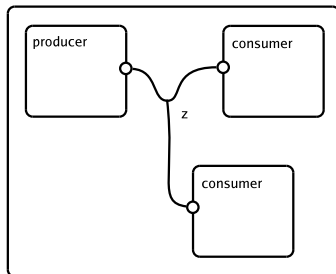
Overview of kiltera processes

Networks

```

process producer[a] :
  <body1>
process consumer[b] :
  <body2>
in
  channel z in
    par
      producer[z]
      consumer[z]
      consumer[z]

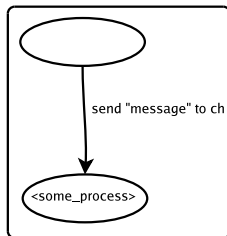
```



Overview of kiltera processes

Sending messages

```
send "message" to ch ->  
  <some_process>
```

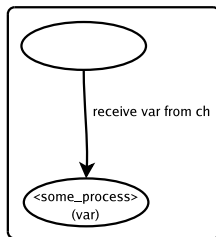


Sending is blocking if the channel is synchronous.

Overview of kiltera processes

Receiving messages

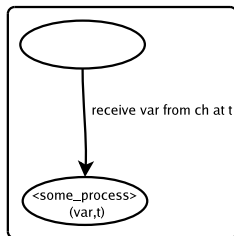
```
receive var from ch ->  
  <some_process>
```



Overview of kiltera processes

Timing

```
receive var from ch at t ->
  <some_process>
```

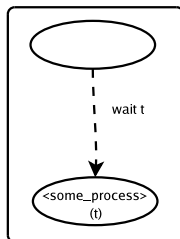


The variable t becomes bound to the time elapsed since the process arrived at the receiving state, until it synchronizes.

Overview of kiltera processes

Waiting

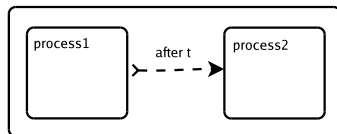
```
wait t ->  
  <some_process>
```



Overview of kiltera processes

Timeout

```
timeout  
  <process1>  
after t ->  
  <process2>
```

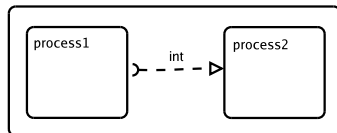


The timeout will take effect if *<process1>* has not finished *or* has not engaged in any external interaction before *t* time units.

Overview of kiltera processes

Interrupt

```
do
  <process1>
interrupt
  <process2>
```



<process1> will be interrupted if *<process2>* engages in an external interaction.

Overview of kiltera processes

Other constructs:

- Data structures
- Conditionals
- Local variable declarations
- Assignment
- Pattern-matching
- Loops
- Recursion
- Channel arrays
- Process arrays
- Auxiliary function definitions

Future work

- Language features:
 - Full mobility: moving “live” processes; ambients
 - Multi-way synchronization
 - Deterministic parallel composition

Future work

- Language features:
 - Full mobility: moving “live” processes; ambients
 - Multi-way synchronization
 - Deterministic parallel composition
- Simulation/Execution:
 - Event traces
 - Logical time simulation

Future work

- Language features:
 - Full mobility: moving “live” processes; ambients
 - Multi-way synchronization
 - Deterministic parallel composition
- Simulation/Execution:
 - Event traces
 - Logical time simulation
- Code generation:
 - kiltera2CSP
 - kiltera2C
 - kiltera2DEVS