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

Ernesto Posse

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

August 28, 2006

Outline

McGill

Introduction

- Transforming Statecharts to DEVS

Introduction

- Transforming Statecharts to DEVS
- Transforming Statecharts to (timed) CSP...

Introduction

- Transforming Statecharts to DEVS
- Transforming Statecharts to (timed) CSP...
- ...and CSP processes to DEVS

Introduction

- Transforming Statecharts to DEVS
- Transforming Statecharts to (timed) CSP...
- ...and CSP processes to DEVS
- Analysis tools for CSP $\longrightarrow$analysis of Statecharts

McGill

Introduction

- Transforming Statecharts to DEVS
- Transforming Statecharts to (timed) CSP...
- ...and CSP processes to DEVS
- Analysis tools for CSP $\longrightarrow$analysis of Statecharts
- But CSP has some odd constructs, a horrible syntax and tools are difficult to obtain.

kiltera

- kiltera: a formalism for describing systems which are:

kiltera

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

kiltera

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

kiltera

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

kiltera

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

kiltera

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

kiltera

- kiltera: a formalism for describing systems which are:
  - dynamic
  - concurrent
  - interactive
  - timed
  - mobile
- Transform Statecharts to kiltera models ...

kiltera

- kiltera: a formalism for describing systems which are:
  - dynamic
  - concurrent
  - interactive
  - timed
  - mobile
- Transform Statecharts to kiltera models ...
- ... and kiltera models to CSP (for analysis)

kiltera

- kiltera: a formalism for describing systems which are:
  - dynamic
  - concurrent
  - interactive
  - timed
  - mobile
- Transform Statecharts to kiltera models ...
- ... and kiltera models to CSP (for analysis)
- ... and DEVS (for fast simulation)

McGill

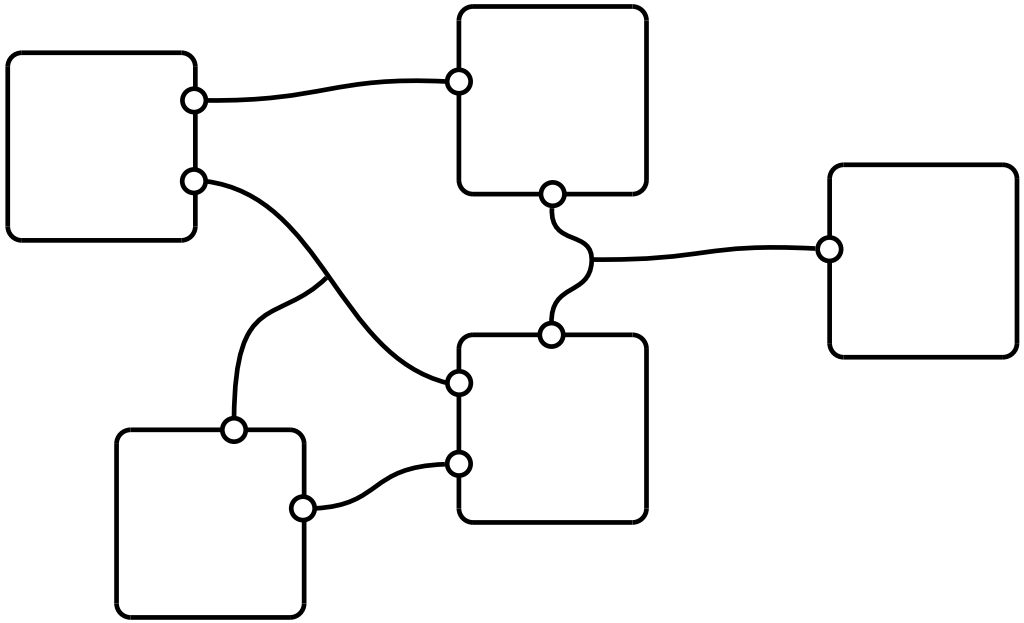
Introduction

- kiltera is not at a lower level than Statecharts:
  - Some kiltera constructs and features cannot be represented in Statecharts e.g.:
    - mobility,
    - measuring elapsed time,
    - private two-way synchronization
    - non-determinism

McGill

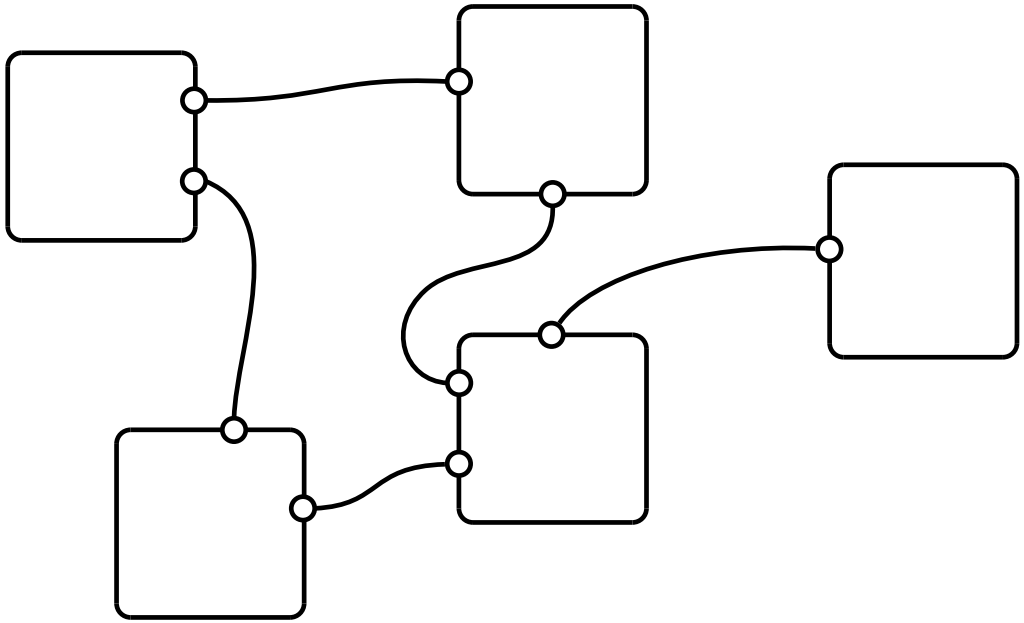
Overview of kiltera processes

A network of processes:



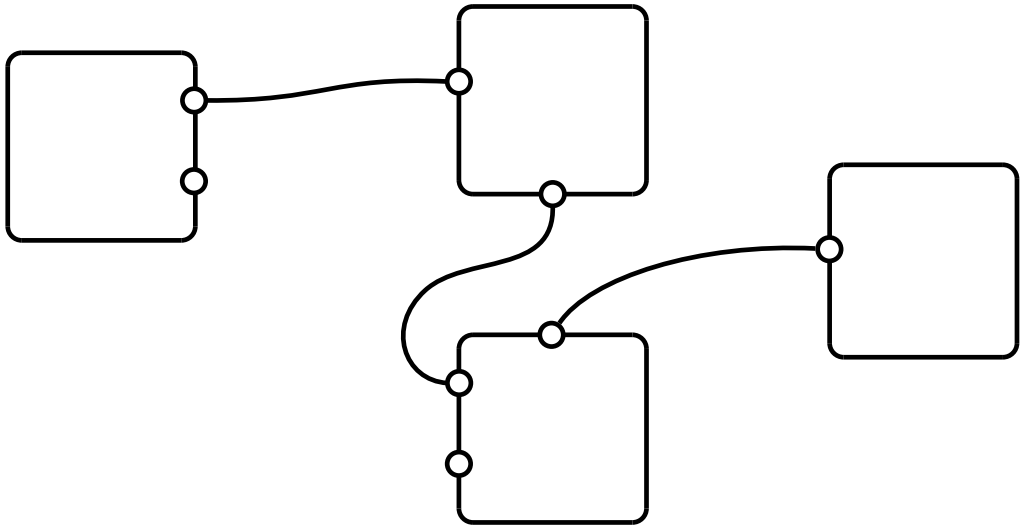
Overview of kiltera processes

Mobility: channels can move



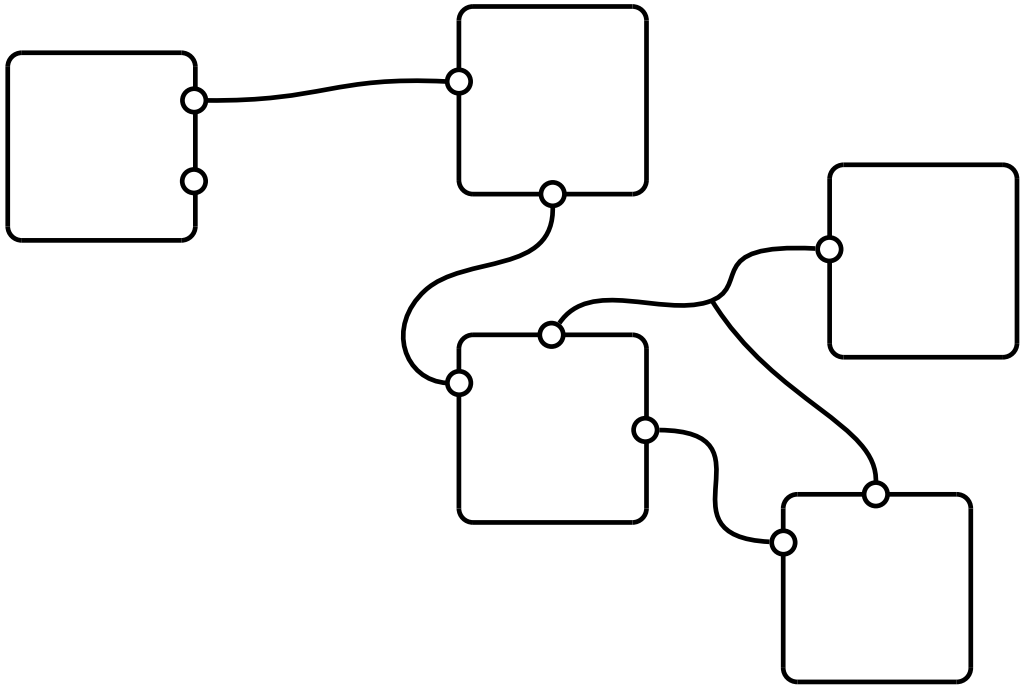
Overview of kiltera processes

Mobility: components can disapear



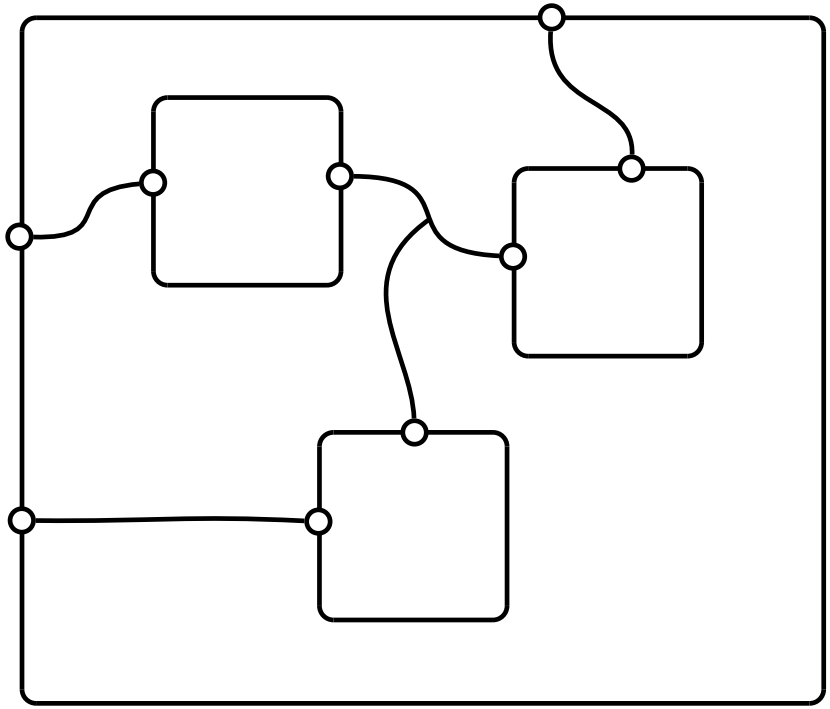
Overview of kiltera processes

Mobility: components can be created



Overview of kiltera processes

Nesting: a network of processes is a process



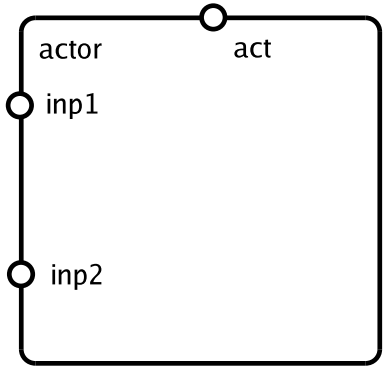
Overview of kiltera processes

- 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

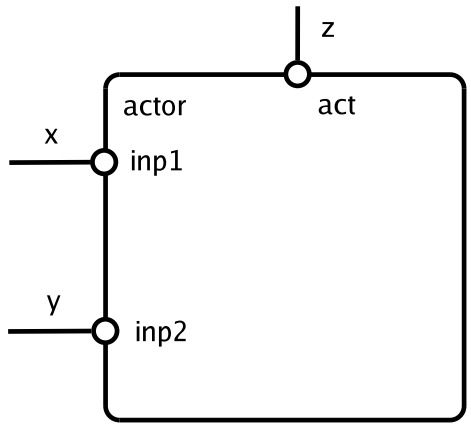
```
process actor[inp1, inp2, act]:
    <body>
```



Overview of kiltera processes

Process instantiation

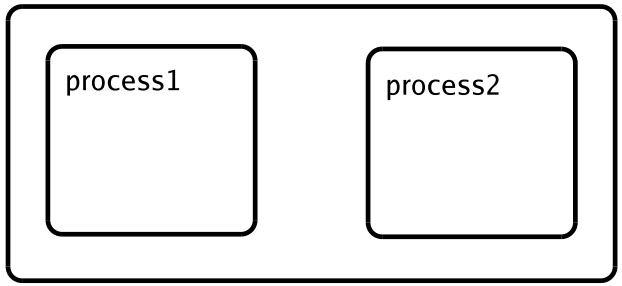
```
process actor[inp1, inp2, act]:
    <body>
in
    ...
    actor[x,y,z]
    ...
```



Overview of kiltera processes

Parallel processes

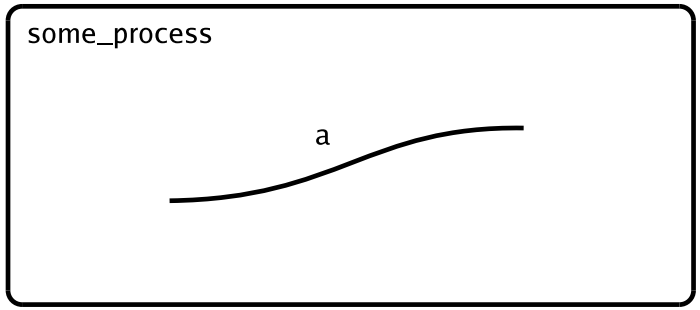
```
par
    <process1>
    <process2>
```



Overview of kiltera processes

Channel declarations

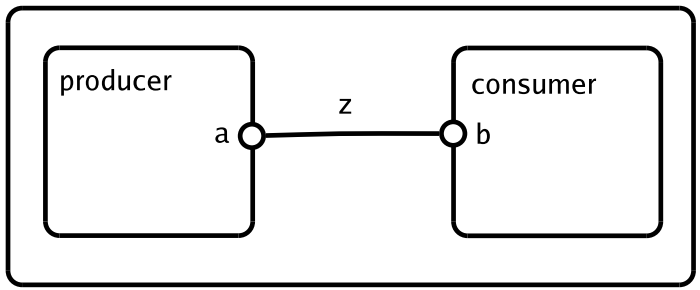
```
channel a in
    <some_process>
```



Overview of kiltera processes

Networks

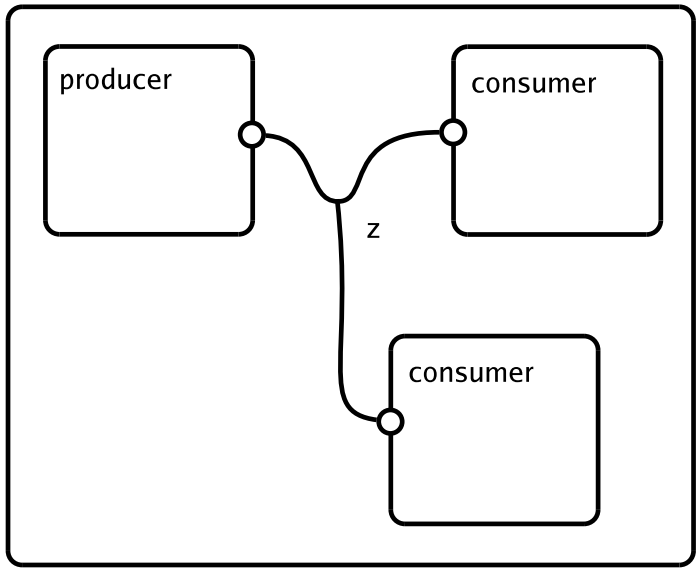
```
process producer[a]:
   <body1>
process consumer[b]:
   <body2>
in
  channel z in
    par
      producer[z]
      consumer[z]
```



Overview of kiltera processes

Networks

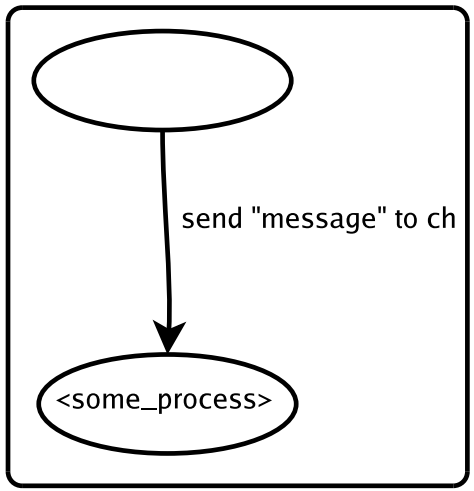
```
process producer[a]:
   <body1>
process consumer[b]:
   <body2>
in
  channel z in
    par
      producer[z]
      consumer[z]
      consumer[z]
```



Overview of kiltera processes

Sending messages

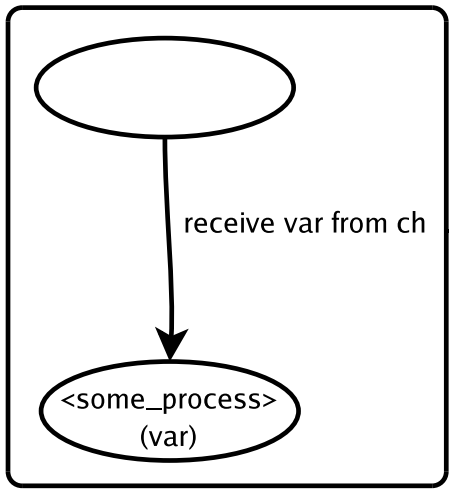
```
send "message" to ch ->
    <some_process>
```



Overview of kiltera processes

Receiving messages

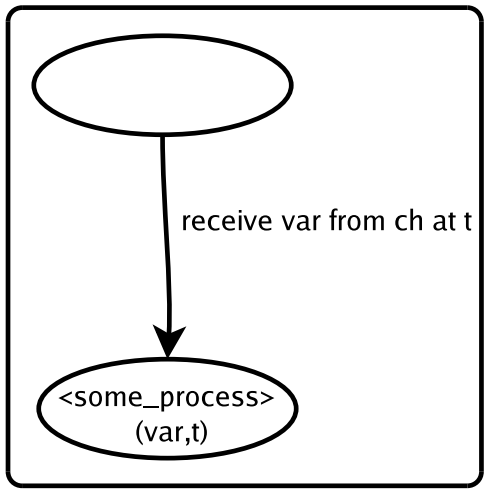
```
receive var from ch ->
    <some_process>
```



Overview of kiltera processes

Timing

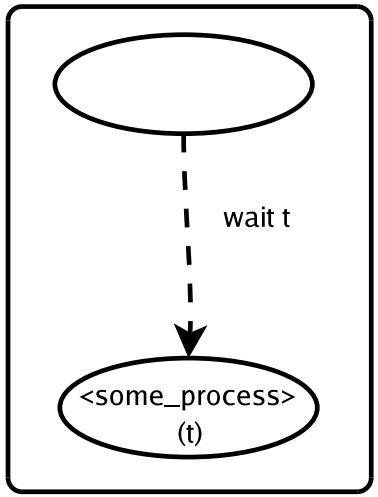
```
receive var from ch at t ->
    <some_process>
```



Overview of kiltera processes

Waiting

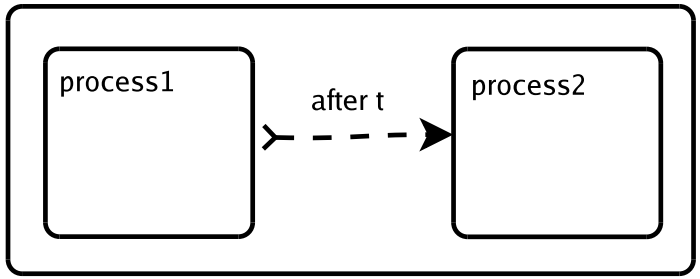
```
wait t ->
    <some_process>
```



Overview of kiltera processes

Timeout

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



Overview of kiltera processes

Other constructs:

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

Demo

Cross your fingers!

Future work

- Simulation/Execution:
  - Event traces
  - Logical time simulation

Future work

- Simulation/Execution:
  - Event traces
  - Logical time simulation
- Language features:
  - Full mobility: moving "live" processes
  - Multi-way synchronization
  - Deterministic parallel composition
  - Himesis graphs as first-class values

Future work

- Simulation/Execution:
  - Event traces
  - Logical time simulation
- Language features:
  - Full mobility: moving "live" processes
  - Multi-way synchronization
  - Deterministic parallel composition
  - Himesis graphs as first-class values
- Code generation:
  - kiltera2CSP
  - kiltera2DEVS
  - kiltera2C