

# Typed Nomadic Time

Andrew Hughes  
<http://www.dcs.shef.ac.uk/~andrew>

Department of Computer Science  
University of Sheffield

York - 22/11/2006

# Outline

## 1 Overview

# Outline

- 1 Overview
- 2 A Simple Example

# Outline

- 1 Overview
- 2 A Simple Example
- 3 Types

# Outline

- 1 Overview
- 2 A Simple Example
- 3 Types
- 4 Further Thoughts and Conclusions

# Overview – System Level

- Hierarchy of *components*
- Multiple levels of *abstraction*
- Behaviour *hidden* within boundaries
- Aim for *compositionality*

# Overview – Component Level

- System of *interacting processes*
- Centers on *synchronisation*
- Behaviour occurs in *synchronous cycles*

# Discrete Time

- Realised formally as a *process calculus*
- Combination of *discrete time* and *migratory mobility*
- Process behaviour can react to *clock ticks*
- *Prioritisation* ensures precedence of internal behaviour



# Clocks and Environments

- The clock appears on the bottom right to indicate that its ticks are visible within the locality, but not outside.
- Ticks become internal actions outside location boundaries.

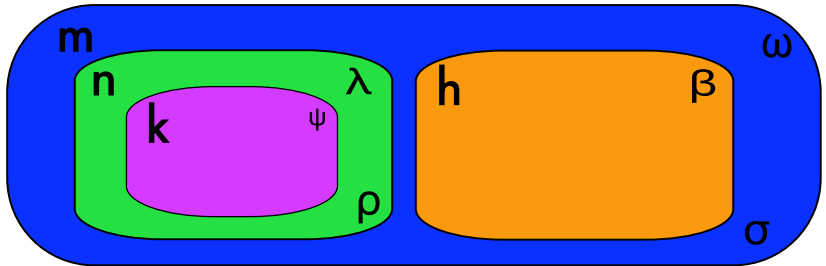
# Mobility

- Processes can *move* between components
- The hierarchy can be *changed*.
- Movement is limited by ‘bouncers’.

# Bouncers

- The locality manager.
- Dictates whether processes are allowed to enter or exit.
- Also controls whether the locality may be destroyed.

# A Sample Environment



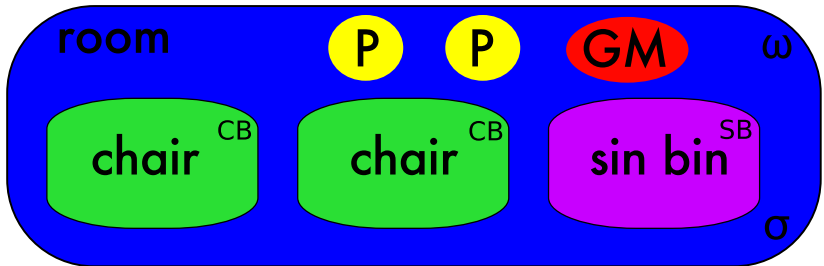
# Modelling Musical Chairs

- 1 The players begin the game standing. The number of players is initially equal to the number of chairs.
- 2 The music starts.
- 3 A chair is removed from the game.
- 4 The music stops.
- 5 Each player attempts to obtain a chair.
- 6 Players that fail to obtain a chair are out of the game.
- 7 The music restarts. Any players who are still in the game leave their chairs and the next round begins (from stage three).

# The Game Environment

- Represented using named locations (*localities*)
- Form a forest structure.
- Each chair is a *locality*.
- The 'sin bin' is also a *locality*.

# The Game Environment



# The Game Environment

## Example

$$r[c[\mathbf{0}]_{\emptyset}^{CB} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} P \mid P \mid GM1]_{\{\sigma\}}^{\Omega}.$$

- $\mathbf{0}$  is a process with no explicit behaviour.
- $\sigma$  is a clock.
- $CB$ ,  $SB$  and  $\Omega$  are *bouncers*.



# Bouncers

- The chair bouncer,  $CB$ , enforces the implicit one-person-per-chair predicate.

## Definition

$$CB \stackrel{\text{def}}{=} \mu X. (\overline{in}. \overline{out}. X + \overline{open})$$

# Bouncers

- The sin bin bouncer,  $SB$ , prevents players getting back out.

## Definition

$$SB \stackrel{\text{def}}{=} \mu X. \overline{\text{in}}. X$$

- Interactions modelled via *action sequences*.
- Synchronisation may occur when:
  - One process offers an *action*,  $a$
  - Another concurrent process offers its *co-action*,  $\bar{a}$
- It appears as an *internal action*

# Music Starts

Start of the music represented as a tick from  $\sigma$

## Example

$$r[c[\mathbf{0}]_{\emptyset}^{CB} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid P \mid P \mid GM1]_{\{\sigma\}}^{\Omega}.$$

evolves to:

## Example

$$r[c[\mathbf{0}]_{\emptyset}^{CB} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \sigma.MP \mid \sigma.MP \mid GM2]_{\{\sigma\}}^{\Omega}.$$

# Chair Removal

Gamesmaster removes a chair

## Example

$$r[c[\mathbf{0}]_{\emptyset}^{CB} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \sigma.MP \mid \sigma.MP \mid \textit{open chair.GM3}]_{\{\sigma\}}^{\Omega}.$$

evolves to:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \sigma.MP \mid \sigma.MP \mid \textit{GM3}]_{\{\sigma\}}^{\Omega}.$$

# Music Stops

The music is stopped

## Example

$$r[\mathbf{0} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \sigma.MP \mid \sigma.MP \mid GM3]_{\{\sigma\}}^{\Omega}.$$

evolves to:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid MP \mid MP \mid GM4]_{\{\sigma\}}^{\Omega}.$$

# Multiway Synchronization

- Seating a player requires:
  - The gamesmaster to perform *on sit in c*.
  - The player to synchronize with this on *sit*.
  - The chair bouncer allowing the player in, via  $\overline{in}$ .

## Example

$$r[\mathbf{0} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \lceil \text{sit.PC} \rceil_{\sigma}(L) \mid \lceil \text{sit.PC} \rceil_{\sigma}(L) \mid \mu X.(\lceil \text{on sit in c.X} \rceil_{\sigma}(GM5))]_{\{\sigma\}}^{\Omega}$$

# Multiway Synchronization

One of the players will get a chair:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0}]_{\emptyset}^{CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \lceil \text{sit.PC} \rceil_{\sigma(L)} \mid \lceil \text{sit.PC} \rceil_{\sigma(L)} \mid \\ \mu X. (\lceil \text{on sit in c.X} \rceil_{\sigma(GM5)}) ]_{\{\sigma\}}^{\Omega}$$

evolves to:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid \overline{PC}]_{\emptyset}^{out.CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid \lceil \text{sit.PC} \rceil_{\sigma(L)} \mid \\ \mu X. (\lceil \text{on sit in c.X} \rceil_{\sigma(GM5)}) ]_{\{\sigma\}}^{\Omega}$$



# Multiway Synchronization

The other will timeout:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid PC]_{\emptyset}^{\overline{out}.CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \parallel [\textit{sit}.PC]_{\sigma}(L) \mid \mu X.([\textit{on sit in c.X}]_{\sigma}(GM5))]_{\{\sigma\}}^{\Omega}$$

evolves to:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid \sigma.PLC]_{\emptyset}^{\overline{out}.CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid L \mid GM5]_{\{\sigma\}}^{\Omega}$$

# Handling The Losers

The loser is moved to the sin bin:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid \sigma.PLC]_{\emptyset}^{\overline{out}.CB} \mid s[\mathbf{0}]_{\emptyset}^{SB} \mid leave.\mathbf{0} \mid \mu X.([\textit{on leave in s.X}] \sigma(GM1))]_{\{\sigma\}}^{\Omega}$$

evolves to:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid \sigma.PLC]_{\emptyset}^{\overline{out}.CB} \mid s[\mathbf{0} \mid \mathbf{0}]_{\emptyset}^{SB} \mid \mu X.([\textit{on leave in s.X}] \sigma(GM1))]_{\{\sigma\}}^{\Omega}$$

# Handling The Losers

The clock ticks:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid \sigma.PLC]_{\emptyset}^{\overline{out.CB}} \mid s[\mathbf{0} \mid \mathbf{0}]_{\emptyset}^{SB} \mid \mu X.([\textit{on leave in s.X}]_{\sigma}(GM1))]_{\{\sigma\}}^{\Omega}$$

evolves to:

## Example

$$r[\mathbf{0} \mid c[\mathbf{0} \mid PLC]_{\emptyset}^{\overline{out.CB}} \mid s[\mathbf{0} \mid \mathbf{0}]_{\emptyset}^{SB} \mid GM1]_{\{\sigma\}}^{\Omega}$$

# Starting Again

The surviving player leaves the chair for the next round:

## Example

$$r[0 \mid c[0 \mid \textit{on stand out c.0} \mid \textit{stand.P}]_{\emptyset}^{\overline{\textit{out.CB}}} \mid s[0 \mid \mathbf{0}]_{\emptyset}^{SB} \mid \textit{GM1}]_{\{\sigma\}}^{\Omega}$$

evolves to:

## Example

$$r[0 \mid c[0 \mid \mathbf{0}]_{\emptyset}^{\overline{\textit{out.CB}}} \mid s[0 \mid \mathbf{0}]_{\emptyset}^{SB} \mid \textit{GM1} \mid P]_{\{\sigma\}}^{\Omega}$$

# Type System: Motivation

- Two reasons:
  - Catch syntactically valid, semantically invalid constructs  
e.g. a bouncer  $in.\overline{in}.0$
  - Restrict mobility by type not cardinality

# Type System: Groups

- Latter achieved by *groups*:
  - $\mathcal{R}$  – Environments that may be *resided in*
  - $\mathcal{O}$  – Environments which may be *opened*
  - $\mathcal{L}$  – Environments which may be *left*
  - $\mathcal{E}$  – Environments which may be *entered*

# Conclusions

- Novel combination of features:
  - Synchronization with arbitrary numbers of agents
  - Movement around a dynamic topology.
- Future work
  - Detailed case studies (e.g. quorum sensing in bacteria)
  - Stochastic extensions?

## The End

Thanks for listening.  
Any questions?





CARDELLI, L., GORDON, A.D.

Mobile Ambients.

*In Proceedings of the 1st International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '98) (1998), no. 1378 in LNCS, Springer-Verlag.*



MILNER, R.

Communication and Concurrency

Prentice-Hall, London (1989)



NORTON, B., LÜTTGEN, G., AND MENDLER, M.

A Compositional Semantic Theory for Synchronous Component-based Design.

*In Proceedings of the 14th International Conference on Concurrency Theory (CONCUR '03) (2003), no. 2761 in LNCS, Springer-Verlag.*