Combining discovery, privacy and symmetry using pattern matching

Thomas Given-Wilson, University of Technology, Sydney

February 10, 2009
Outline

- Introduction
  - Motivating problem
  - Discovery, privacy and symmetry
- Background (in 3 examples)
  - $\pi$ calculus
  - Fusion calculus
  - Concurrent pattern calculus
- New calculus (overview)
  - The basics
  - Match rule/reduction
- Solving the motivating problem
- Conclusions
Motivating problem

The problem is for a buyer to find a particular performance that is being advertised by a seller and purchase a ticket to attend.

**Buyer:** find a performance, purchase a ticket

**Seller:** advertise a performance, sell a ticket

Three key concepts:

1. Buyer and seller *discover* each other
2. Exchange information *privately*
3. Treat both parties equally, i.e. *symmetry*
Buyer and seller need to discover each other:

- No prior knowledge of each other
- No third party/broker
- Some agreed data format (e.g. XML)
- Only communicate if discovery (matching) successful

Known problem in web services (Benatallah, Hacid, Leger, Rey and Toumani 2005).
Buyer and seller communicate privately:

- Other processes cannot:
  - see the communication
  - participate in the communication
  - interfere with the communication

- Use a process calculus approach
  - Logical/provable
  - Style of Gordon and Abadi (1997)
Symmetry

Buyer and seller exchange information:

- Both parties want a symmetric exchange:
  - Buyer: only provide payment details if they receive a ticket
  - Seller: only provide a ticket if they receive payment details
- Communicate in both directions in a single transaction
- Symmetry varies:
  - Newton: for every action there is an equal and opposite reaction
  - Milner: communication is a handshake interaction between two parties
\( \pi \) calculus

Developed by Milner, Parrow and Walker (1992) as a simple model for communicating systems.

- *Names* as channels, variables and identifiers.
- Communication is uni-directional
- All communication via named channels
  \[ n(x).P | n\langle y \rangle.Q \Rightarrow \{y/x\}P|Q \]
- Generalises \( \lambda \) calculus

**Discovery:** unable to express  
**Privacy:** achieved with channel name  
**Symmetry:** every communication is between two processes
Fusion calculus

Created as “a step towards a canonical calculus of concurrency” by Parrow and Victor (1998, p. 176).

- Expresses shared state and reduction strategies
- Communication is an equivalence relation:
  \[ n \ x. P | \overline{n} \ y. Q \overset{x \equiv y}{\leftrightarrow} P | Q \]
- “Structured” communication (as in polyadic \( \pi \) calculus)
- Generalises (polyadic) \( \pi \) calculus (and so \( \lambda \) calculus)

**Discovery:** unable to express

**Privacy:** achieved with channel name

**Symmetry:** equivalence relation
Concurrent pattern calculus

The concurrent pattern calculus of Gorla and Jay (2007) as a concurrent adaptation of pattern calculus.

- Inspired by pattern calculus of Jay and Kesner (2006)
- Uses symmetric pattern matching for communication:
  \[
  ([\theta]p \rightarrow s) | ([\varphi]q \rightarrow t) \rightarrow_\gamma \{ p[\theta|\varphi]q \}_\gamma (s|t)
  \]
- Can express (and match) data structures
- Generalises Linda (Gelernter 1985)

**Discovery:** achieved through matching data structures

**Privacy:** unable to express

**Symmetry:** communication in both directions
New calculus I

Terms:

\[ t ::= x \quad \text{names} \]
\[ t|t \quad \text{parallel composition} \]
\[ (\nu x)t \quad \text{restriction} \]
\[ t \bullet t \quad \text{compounding} \]
\[ \langle t \rangle \quad \text{local} \]
\[ [\theta]t \rightarrow t \quad \text{case} \]

Free names of a term \( t = \text{fn}(t) \)
Local terms of a term \( t = \text{lt}(t) \)
New calculus II

Match rule:

\[
\begin{align*}
\{x[\theta \parallel \varphi]q\} &= \{q/x\} & x \in \theta, \ \fn(q) \cap \varphi &= \{\} = \lt(q) \\
\{p[\theta \parallel \varphi]x\} &= \{p/x\} & x \in \varphi, \ \fn(p) \cap \theta &= \{\} = \lt(p) \\
\{x[\theta \parallel \varphi]x\} &= \{\} & x \notin \theta \cup \varphi \\
\{\langle p \rangle[\theta \parallel \varphi]q\} &= \{p[\|]q\} \\
\{p[\theta \parallel \varphi]\langle q \rangle\} &= \{p[\|]q\} \\
\{p_1 \cdot p_2[\theta \parallel \varphi]q_1 \cdot q_2\} &= \{p_1[\theta \parallel \varphi]q_1\} \cup \{p_2[\theta \parallel \varphi]q_2\} \\
\{p[\theta \parallel \varphi]q\} &= \text{undefined otherwise.}
\end{align*}
\]

Reduction:

\[
([\theta]p \rightarrow s) | ([\varphi]q \rightarrow t) \Rightarrow \{p[\theta \parallel \varphi]q\}(s \mid t) \quad \fn(s) \cap \varphi = \{\} = \fn(t) \cap \theta
\]
Solving the motivating problem

Represent the performance information by some data structure denoted Perf.

**Buyer process** (with credit card information CreditC):

\[
[\text{chan}] \text{Perf} \bullet \text{chan} \rightarrow [\text{tn}]\langle \text{chan} \rangle \bullet \text{CreditC} \bullet \text{tn} \rightarrow B
\]

**Seller process** (with ticket number 849):

\[
(\nu \text{ priv}) \text{Perf} \bullet \text{priv} \rightarrow [\text{cc}]\langle \text{priv} \rangle \bullet \text{cc} \bullet 849 \rightarrow S
\]

**Reductions** (\(\Rightarrow\)):

\[
[\text{chan}] \text{Perf} \bullet \text{chan} \rightarrow [\text{tn}]\langle \text{chan} \rangle \bullet \text{CreditC} \bullet \text{tn} \rightarrow B
\]

| (\nu \text{ priv}) \text{Perf} \bullet \text{priv} \rightarrow [\text{cc}]\langle \text{priv} \rangle \bullet \text{cc} \bullet 849 \rightarrow S
\]

\[
\Rightarrow (\nu \text{ priv})([\text{tn}]\langle \text{chan} \rangle \bullet \text{CreditC} \bullet \text{tn} \rightarrow B
\]

| [\text{cc}]\langle \text{priv} \rangle \bullet \text{cc} \bullet 849 \rightarrow S)
\]

\[
\Rightarrow (\nu \text{ priv})(B \mid S)
\]
Conclusions

- The motivating problem
  - demonstrates the desirability of discovery, privacy and symmetry
  - is typical of a class of examples
  - cannot be expressed in existing calculi
- The calculus overviewed here
  - expresses discovery, privacy and symmetry
  - expresses a solution to the motivating problem
  - (hopefully) subsumes/generalises pattern calculus (Jay and Kesner 2009, Jay 2009)


