Web Data Modelling and Securing

Jovanka Pantović
University of Novi Sad, Serbia

joint work with

Mariangiola Dezani-Ciancaglini, University of Torino, Italy
Silvia Ghilezan, University of Novi Sad, Serbia
Svetlana Jakšić, University of Novi Sad, Serbia
Daniele Varacca, Paris VII, France

Workshop on Formal and Automated Theorem Proving and Applications
University of Belgrade, January 30,2009

Outline of the talk

- Motivation and origins
  - process calculus - π-calculus,
  - process mobility, distributed systems - $d\pi$-calculus
  - distributed systems with data - X$d\pi$-calculus
- X$d\pi$-calculus - syntax
- Operational semantics of X$d\pi$-calculus
- Type system for X$d\pi$-calculus
- Safety properties of the type system
- Example - distributed library

Origins of process algebras

- Petri nets (1962), Actor model (1973)
- Calculus of Communicating Systems (CCS), R.Milner (1973-1980)
- Communicating Sequential Processes (CSP), T.Hoare (1978-1980)
- Algebra of Communicating Processes (ACP), J.Bergstra, J.P.Klop (1982)
- π-calculus: Milner, Parrow, Walker
- Ambient calculus, Fusion calculus, Spi, dπ,...

Motivation

- interaction and mobility
- description and analysis of concurrent computation (processes) whose configuration changes over the computation
- Processes and Names (channels and variables)

π-processes

Processes:

$P \mid Q$	parallel composition
$c(x).P$	input prefixing
$\overline{c}\langle v\rangle.P$	output prefixing
$!P$	replication
$(\nu\, c)P$	creation of a new name
0	null process

π- structural congruence

The structural congruence is the smallest congruence satisfying the following:

- Alpha-conversion
- Axioms for parallel composition
  - $P|Q \equiv Q|P$
  - $(P|Q)|R \equiv P|(Q|R)$
  - $P|0 \equiv P$
- Axioms for restriction
  - $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$
  - $(\nu x)0 \equiv 0$
- Axiom for replication $!P \equiv P|!P$
- Axiom $(\nu x)(P|Q) \equiv (\nu x)P|Q$, where x is not a free name of Q

π - reduction rules

- $\overline{c}\langle v\rangle.P|c(x).Q \rightarrow P|Q\{v/x\}$
- $\dfrac{P \longrightarrow Q}{P|R \longrightarrow Q|R}$ (par)
- $\dfrac{P \longrightarrow Q}{(\nu x)P \longrightarrow (\nu x)Q}$ (res)
- $\dfrac{P \longrightarrow P'}{Q \longrightarrow Q'}$ (struct) if $P \equiv Q$ and $P' \equiv Q'$

$d\pi$ and $X d\pi$

- $d\pi$-calculus - Hennessy
  - mobile processes (agents) in a distributed world
  - Locations
  - A new process: $\mathtt{go}\ l.P$
- X$d\pi$-calculus - Gardner, Maffeis
  - localised mobile processes
  - distributed, dynamic, semi-structured web data
  - Data
  - Locations
  - Two more processes: `run`, `update`

Typed $X d\pi$

- Distributed systems - decentralised peer-to-peer networks
  - Management of semi-structured and distributed data
  - Need for security and privacy
  - Exchange of data and processes preserving security
- One solution - typed models
  - control of access
  - control of movements rights
  - control communication of values

Outline of the talk

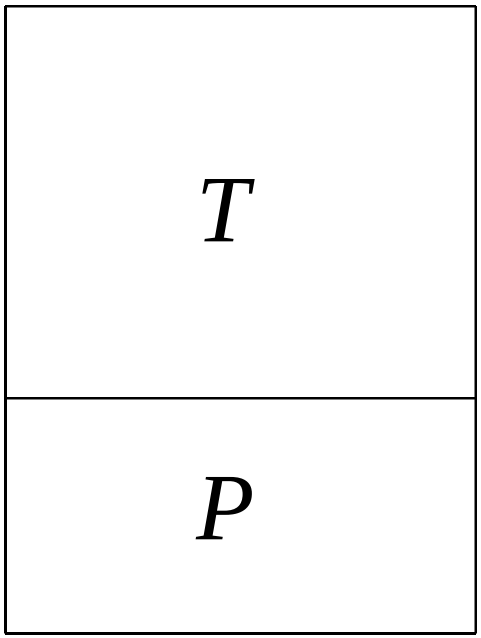
- Motivation
  - process calculus - π-calculus,
  - process mobility, distributed systems - $d\pi$-calculus
  - distributed systems with data - X$d\pi$-calculus
- X$d\pi$-calculus - syntax
- Operational semantics of X$d\pi$-calculus
- Type system for X$d\pi$-calculus
- Safety properties of the type system
- Example - distributed library

X$d\pi$ calculus

Networks:

$$\mathbf{N} ::= \mathbf{0} \;\parallel\; l^{h}[T \parallel P] \;\parallel\; (\nu c^{tv})\mathbf{N} \;\parallel\; \mathbf{N} \mid \mathbf{N}$$

Location:

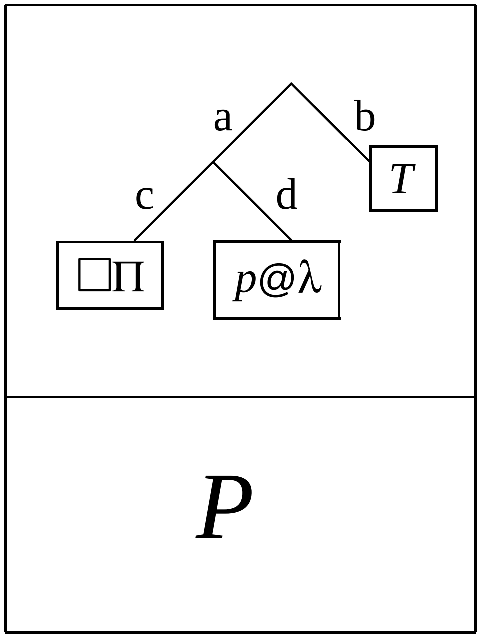


$\mathrm{X}d\pi$ calculus

Networks:

$$\mathbf{N} ::= \mathbf{0} \parallel l^h[T \parallel P] \parallel (\nu c^{tv})\mathbf{N} \parallel \mathbf{N} \mid \mathbf{N}$$

Location:



$$T := \emptyset \parallel x \parallel T \mid T \parallel \mathtt{a}[T] \parallel \mathtt{a}[\Box\Pi] \parallel \mathtt{a}[p@\lambda]$$

Syntax of processes

$Xd\pi$ calculus

Syntax of processes

P	$::=$	0	the nil process
	$\|$	$P \mid P$	composition of processes
	$\|$	$(\nu c^{tv})P$	declare new channel name c
	$\|$	$\overline{\gamma}\langle v \rangle$	output value v on a channel γ
	$\|$	$\gamma(x).P$	input parameterized by a variable x
	$\|$	$!\gamma(x).P$	replication of an input process

$\pi-$calculus

$Xd\pi$ calculus

Syntax of processes

P	$::=$	0	the nil process
	$\mid$	$P \mid P$	composition of processes
	$\mid$	$(\nu c^{tv})P$	declare new channel name c
	$\mid$	$\overline{\gamma}\langle v\rangle$	output value v on a channel γ
	$\mid$	$\gamma(x).P$	input parameterized by a variable x
	$\mid$	$!\gamma(x).P$	replication of an input process
	$\mid$	$\mathsf{go}\ \lambda.P$	migrate to location λ, continue as P
	$\mid$	$\mathsf{go}\ \circlearrowleft .P$	migrate home, continue as P

$d\pi$–calculus

Syntax of processes

P	$::=$	0	the nil process
	$\mid$	$P \mid P$	composition of processes
	$\mid$	$(\nu c^{tv})P$	declare new channel name c
	$\mid$	$\overline{\gamma}\langle v \rangle$	output value v on a channel γ
	$\mid$	$\gamma(x).P$	input parameterized by a variable x
	$\mid$	$!\gamma(x).P$	replication of an input process
	$\mid$	$\mathtt{go}\ \lambda.P$	migrate to location λ, continue as P
	$\mid$	$\mathtt{go}\ \circlearrowleft .P$	migrate home, continue as P
	$\mid$	$\mathtt{run}_p$	run the processes identified by the path expression p
	$\mid$	$\mathtt{update}_p(\chi, V).P$	update command

Outline of the talk

- Motivation
  - process calculus - π-calculus,
  - process mobility, distributed systems - $d\pi$-calculus
  - distributed systems with data - X$d\pi$-calculus
- X$d\pi$-calculus - syntax
- Operational semantics of X$d\pi$-calculus
- Type system for X$d\pi$-calculus
- Safety properties of the type system
- Example - Distributed library

Operational semantics

(com) $l^h[\, T \parallel \overline{c}^{tv}\langle v\rangle \mid c^{tv}(z).P \mid Q \,] \longrightarrow l^h[\, T \parallel P\{v/z\} \mid Q \,]$

(com!) $l^h[\, T \parallel \overline{c}^{tv}\langle v\rangle \mid !c^{tv}(z).P \mid Q \,] \longrightarrow l^h[\, T \parallel !c^{tv}(z).P \mid P\{v/z\} \mid Q \,]$

Operational semantics

(com) $l^h[\, T \parallel \overline{c}^{tv}\langle v\rangle \mid c^{tv}(z).P \mid Q \,] \longrightarrow l^h[\, T \parallel P\{v/z\} \mid Q \,]$

(com!) $l^h[\, T \parallel \overline{c}^{tv}\langle v\rangle \mid !c^{tv}(z).P \mid Q \,] \longrightarrow l^h[\, T \parallel !c^{tv}(z).P \mid P\{v/z\} \mid Q \,]$

(stay) $l^h[\, T \parallel \mathsf{go}\ l^h.P \mid Q \,] \longrightarrow l^h[\, T \parallel P \mid Q \,]$

(go) $l^h[\, T_1 \parallel \mathsf{go}\ m^j.P \mid Q \,] \mid m^j[\, T_2 \parallel R \,] \longrightarrow l^h[\, T_1 \parallel Q \,] \mid m^j[\, T_2 \parallel P \mid R \,]$

Operational semantics

(com) $l^h[\, T \,||\, \overline{c}^{tv}\langle v\rangle \mid c^{tv}(z).P \mid Q \,] \longrightarrow l^h[\, T \,||\, P\{v/z\} \mid Q \,]$

(com!) $l^h[\, T \,||\, \overline{c}^{tv}\langle v\rangle \mid !c^{tv}(z).P \mid Q \,] \longrightarrow l^h[\, T \,||\, !c^{tv}(z).P \mid P\{v/z\} \mid Q \,]$

(stay) $l^h[\, T \,||\, \text{go } l^h.P \mid Q \,] \longrightarrow l^h[\, T \,||\, P \mid Q \,]$

(go) $l^h[\, T_1 \,||\, \text{go } m^j.P \mid Q \,] \mid m^j[\, T_2 \,||\, R \,] \longrightarrow l^h[\, T_1 \,||\, Q \,] \mid m^j[\, T_2 \,||\, P \mid R \,]$

(run)
$$\frac{p(T) \leadsto_{p,l^h,\Box x,\Box x} T, \{\{\Box P_1/\Box x\},\ldots,\{\Box P_n/\Box x\}\}}{l^h[\, T \,||\, \mathtt{run}_p \mid Q \,] \longrightarrow l^h[\, T \,||\, P_1 \mid \ldots \mid P_n \mid Q \,]}$$

(update)
$$\frac{p(T) \leadsto_{p,l^h,\chi,V} T', \{\mathsf{s}_1,\ldots,\mathsf{s}_n\}}{l^h[\, T \,\|\, \mathtt{update}_p(\chi,V).P \mid Q \,] \longrightarrow l^h[\, T' \,\|\, P\mathsf{s}_1 \mid \ldots \mid P\mathsf{s}_n \mid Q \,]}$$

Outline of the talk

- Motivation
  - process calculus - π-calculus,
  - process mobility, distributed systems - $d\pi$-calculus
  - distributed systems with data - X$d\pi$-calculus
- X$d\pi$-calculus - syntax
- Operational semantics of X$d\pi$-calculus
- Type system for X$d\pi$-calculus
- Safety properties of the type system
- Examples
  - Distributed library
  - Remote voting system

Type system

The main goals:

- to control communication of values
- to control migration of processes
- to control access to data

Type system

Net	$Path$	$PathLocal$
$ch(tv)$	$Pointer(i)$	$PointerLocal(i)$
$Loc(i)$	$Tree(i)$	$TreeLocal(i)$
$Script(i)$	$Proc(i)$	$ProcLocal(i)$

$$(\mathcal{L}, \leq), \quad i \in \mathcal{L}$$

$$tv ::= ch(tv) \parallel Loc(i) \parallel Script(i) \parallel Path^{\star} \parallel Tree^{\star}(i)$$

Communication of values:

$$\frac{\Sigma \vdash_{\ominus} v : tv \quad \Sigma \vdash \gamma : ch(tv) \quad |tv| \leq i \leq h \quad \ominus \in \{\varepsilon, h\}}{\Sigma \vdash_{h} \overline{\gamma}\langle v \rangle : Proc^{\star}(i)}$$

$$\frac{\Sigma, x : tv \vdash_{h} P : Proc^{\star}(i) \quad \Sigma \vdash \gamma : ch(tv) \quad |tv| \leq i}{\Sigma \vdash_{h} \gamma(x).P : Proc^{\star}(i)}$$

Initial type system

Migration of processes:

$$\frac{\Sigma \vdash_h P : Proc^\star(j) \quad \Sigma \vdash \lambda : Loc(j) \quad j \le i \le h}{\Sigma \vdash_h \mathsf{go}\ \lambda.P : Proc^\star(i)}$$

$$\frac{\Sigma \vdash_h P : Proc^\star(h)}{\Sigma \vdash_h \mathsf{go}\ \circlearrowleft .P : Proc^\star(i)}$$

Initial type system

Access to data:

$$\frac{\Sigma \vdash p : Path^\star \quad i \leq h}{\Sigma \vdash_h \mathtt{run}_p : Proc^\star(i)}$$

$$\frac{\Sigma \vdash p : Path^\star \quad \Sigma \cup \Sigma_0 \vdash_h P : Proc^\star(i) \quad \Sigma_0 \vdash_i V : SPT(j) \quad j \leq i}{\Sigma \vdash_h \mathtt{update}_p(\chi, V).P : Proc^\star(i)}$$

$$\Sigma_0 = \begin{cases} x : Tree(j) \text{ if } \chi = x^j, \\ x : Loc(j), y : Path^\star \text{ if } \chi = y^\star @ x^j, \\ x : ProcLocal(j) \text{ if } \chi = \Box x^j \end{cases}$$

Ongoing type system

- replacing 2 rules for migration of processes by:

$$\frac{\Sigma \vdash P : Proc^{\star}(j)}{\Sigma \vdash \mathsf{go}\ l^{j}.P : Proc^{\star}(i)}$$

- rewriting each initial rule for processes without the decoration on the turn-style in the process judgements and without the relative conditions

- leaving the decoration for data

Networks:

$$\frac{\vdash_i T : Tree(i) \qquad \vdash_i P : Proc(i)}{\vdash l^i[\, T \parallel P\{l^i/\circlearrowleft\}\,] : Net}\ (netIloc)$$

$$\frac{\vdash_i T : Tree(i) \qquad \vdash P : Proc(i)}{\vdash l^i[\, T \parallel P\,] : Net}\ (netOloc)$$

Outline of the talk

- Motivation
  - process calculus - π-calculus,
  - process mobility, distributed systems - $d\pi$-calculus
  - distributed systems with data - X$d\pi$-calculus
- X$d\pi$-calculus - syntax
- Operational semantics of X$d\pi$-calculus
- Type system for X$d\pi$-calculus
- Safety properties of the type system
- Distributed library

Safety

- *Subject reduction - type preservation under reduction*
  If $\vdash \mathbf{N} : Net$ and $\mathbf{N} \rightarrow \mathbf{N}'$, then $\vdash \mathbf{N}' : Net$.

- *Safety properties - consequences of subject reduction*

  P0 a channel in a location of level h can communicate only values whose security level is less than or equal to h;

  P1 a pointer from a tree in a location of level h to a tree in a location of level j implies $j \leq h$;

  P2 a process can migrate from a location of level h to a location of level j only if either $j \leq h$ or it “goes home” (i.e. it is the “descendent” of a process originating from that location).

Distributed library

Central library

$$Library^1 \,[\; \texttt{Management}\,[\; \texttt{WorkingHours}[HourPlan^2] \mid \ldots\,] \mid$$
$$\texttt{Catalog}\,[\; \ldots \mid \texttt{Pierce}[\texttt{Types}[\texttt{Pierce/Types}@LICS^1] \mid$$
$$\texttt{Category}[\texttt{Pierce/Category}@LICS^1] \ldots] \mid$$
$$\mid \texttt{Cohn}[\texttt{Universal}[\texttt{Cohn/Universal}@ALGEBRA^1] \mid \ldots]|| \ldots]$$

Field libraries

$$LICS^1 [\ldots \mid \texttt{Pierce}\,[\,\texttt{Types}\,[\,Book.pdf^1\,] \mid \texttt{Category}\,[\,Book.pdf^1] \mid \ldots\,] \,||\, \ldots],$$

$$Reader^1,\; Staff^2\;, Manager^3$$

$$Reader^1[\texttt{Book}[\texttt{Pierce}[\emptyset] \mid \ldots] \,||\, \texttt{go}\; Library^1.\texttt{copy}_{\texttt{Catalog/Pierce/Types}}(y@x^1).$$
$$\texttt{go}\; x.\texttt{copy}_{\texttt{Pierce/Types}}(z^1).\texttt{go}\; Reader^1.\texttt{paste}_{\texttt{Book/Pierce}}\langle \texttt{Types}[z] \rangle\,] \rightarrow \ldots \rightarrow$$
$$Reader^1[\texttt{Book}[\texttt{Pierce}[Book.pdf] \mid \ldots] \,||\, 0]$$

goes to the library, reads in the catalogue the location of the book, goes to the sublibrary, copies the book and pastes the copy in the tree of his location.

Distributed library

Central library

$$Library^1[\ \texttt{Management}[\ \texttt{WorkingHours}[HourPlan^2] \mid \ldots] \mid$$
$$\texttt{Catalog}[\ \ldots \mid \texttt{Pierce}[\texttt{Types}[\texttt{Pierce/Types}@LICS^1] \mid$$
$$\texttt{Category}[\texttt{Pierce/Category}@LICS^1]\ldots] \mid$$
$$\mid \texttt{Cohn}[\texttt{Universal}[\texttt{Cohn/Universal}@ALGEBRA^1] \mid \ldots]] \mid \ldots]$$

Field libraries

$$LICS^1[\ldots \mid \texttt{Pierce}[\texttt{Types}[Book.pdf^1] \mid \texttt{Category}[Book.pdf^1] \mid \ldots] \parallel \ldots],$$

$$Reader^1,\ Staff^2\ ,Manager^3$$

- $Reader^1$ - security level 1, cannot modify anything, but can copy the book
- $Staff^2$ - security level 2, cannot modify the $HourPlan$, but can copy the $HourPlan$ and can update the catalogue
- $Manager^3$ - security level 3, can update all the data in the *Library*

Future work

- study modifications of our type system
  - different security levels to different branches in the tree
  - role based access control
- use the behavioural equivalencies in order to compare networks.

More details in:

- M.Dezani-Ciancaglini, S. Ghilezan, J. Pantovic: Security types for dynamic web data, TGC'06 (Lucca). *Lecture Notes in Computer Science* 4661: 263-280 (2007)

- M.Dezani-Ciancaglini, S. Ghilezan, J. Pantovic, D. Varacca: Security types for dynamic web data. *Theoretical Computer Science* 402(2-3): 156-171 (2008)

The very beginning

Thank you for your attention!