Dynamic Process/Service Composition/Combination

Ugo Montanari
Dipartimento di Informatica
Università di Pisa

March 18, 2009

Work in collaboration with
Marzia Buscemi, IMT Lucca

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

Global Computing

- new models of computation
- new programming and analysis methods
  - distribution, concurrency
  - open endness
  - process mobility, service discovery
  - structuring into sessions, transactions
  - typing, code analysis, verification also at run time
- difficult to distinguish between design, execution and reconfiguration phases
- still distinction between procedural information and declarative information

Composition vs. Combination

- local computer system
  - sequential/parallel program composition
- wide area net
  - discovering and combining processes
- choreography, orchestration, coordination methods
- two-sided or multi-party sessions
- negotiations with non-functional service level agreements
- long transactions with failures and compensations
- architectural design languages for business-to-business, telecom or health applications.

Service-Oriented Computing

- distributed information systems + distributed concurrent programming
- accessing relevant information
  - about the network
  - about data and ontology of the application
- expressive contracts and service level agreements
- guarantees about security
- deadlock avoidance
- conformance of orchestration and choreography
- existence of compensations in the presence of failures

European Project SENSORIA, I

- linguistic primitives for modelling and programming
- qualitative and quantitative analysis methods
- sound engineering and deployment techniques

Some relevant studies (see abstract on the web for links)

- CaSPiS
  - two-sided sessions and pipelining, recursion
  - handling (unexpected) termination of the partner's side of a session.
  - session types guarantee communicating entities will not block
  - session type inference is decidable
  - implemented general tool
  - MUSE multiparty sessions

European Project SENSORIA, II

- The process calculus Cc-Pi
  - name-passing calculi, concurrent constraint programming
  - requirements on service level agreements are constraints
  - soft notions of constraints
- Architectural Design Rewriting
  - software architectures development & reconfiguration with term-rewriting
  - proof that a design was constructed according to the style
  - naturally supports style-preserving reconfigurations
  - MAUDE implementation
- Lambda-req for security
  - selecting and invoking services
  - behavior of services over-approximated by a type and effect system
  - the approximation is model-checked

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

About Names

Names can be:

1. channels
2. identifiers
3. values (data)
4. objects
5. pointers
6. references
7. locations
8. encryption keys
9. ...

Names can:

1. be created and destroyed
2. sent them around to share information
3. acquired to communicate with previously unknown processes
4. used for evaluation or communication
5. be tested to take decisions based on their values
6. used as private means of communication, e.g. to share secret
7. ...

Syntax of π-Calculus

We assume a countably infinite set of names $\mathcal{N}$ is defined.

(Processes) P	$::=$		S	sum
		$\mid$	$P_1 \mid P_2$	parallel composition
		$\mid$	$(\nu x)P$	name restriction
		$\mid$	$!P$	replication
(Sums) S	$::=$		$\mathbf{0}$	inactive process (nil)
		$\mid$	$\pi.P$	prefix
		$\mid$	$S_1 + S_2$	choice
(Prefixes) π	$::=$		$\overline{x}\langle y\rangle$	sends y on x
		$\mid$	$x(z)$	substitutes for z the name received on x
		$\mid$	τ	internal action
		$\mid$	$[x = y]\pi$	matching: tests equality of x and y

Structural Congruence

$$P \mid \mathbf{0} \equiv P \qquad P_1 \mid P_2 \equiv P_2 \mid P_1 \qquad P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3$$

$$S + \mathbf{0} \equiv S \qquad S_1 + S_2 \equiv S_2 + S_1 \qquad S_1 + (S_2 + S_3) \equiv (S_1 + S_2) + S_3$$

$$!P \equiv P \mid !P \qquad [a = a]\pi.P \equiv \pi.P$$

$$(\nu a)\mathbf{0} \equiv \mathbf{0} \qquad (\nu a)(\nu b)P \equiv (\nu b)(\nu a)P \qquad \frac{a \notin fn(P)}{P \mid (\nu a)Q \equiv (\nu a)(P \mid Q)}$$

$$P \equiv P \qquad \frac{P \equiv Q}{Q \equiv P} \qquad \frac{P \equiv Q \quad Q \equiv R}{P \equiv R} \quad \text{(equivalence)}$$

$$\frac{P =_\alpha P'}{P \equiv P'} \qquad \frac{P \equiv P'}{\mathbb{C}[P] \equiv \mathbb{C}[P']} \quad \text{(congruence)}$$

Reduction Rules

The so-called *reduction semantics* focuses on *internal* moves $P \longmapsto Q$ only.

$$(RTAU) \quad \frac{}{(\tau.P + S) \longmapsto P}$$

$$(RCOM) \quad \frac{}{(a(x).P_1 + S_1) | (\overline{a}\langle b \rangle.P_2 + S_2) \longmapsto P_1[{}^{b}/_{x}] | P_2}$$

$$(RPAR) \quad \frac{P \longmapsto P'}{P \mid Q \longmapsto P' \mid Q}$$

$$(RRES) \quad \frac{P \longmapsto P'}{(\nu a)P \longmapsto (\nu a)P'}$$

$$(RSTRUCT) \quad \frac{P \equiv Q \quad Q \longmapsto Q' \quad Q' \equiv P'}{P \longmapsto P'}$$

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

Constraint Semirings

Definition

A c-semiring is a tuple $\langle A, +, \times, 0, 1\rangle$ s.t.:

- A a set and $0, 1 \in A$ 1 (0) identity, absorbing on x, + (+, x)
- $+$ commutative, associative, idempotent ($a + b$ is the worst constraint that is best than a and b)
- $\times$ associative, commutative, distributes over $+$ ($a \times b$ combines a and b).

Partial ordering $\leq$ on c-semirings

$a \leq b$ iff $a + b = b$ (intuitively, a is more constrained than b, alias $a \vdash b$).

Examples

- Classical CSPs: $\langle\{\text{False}, \text{True}\}, \vee, \wedge, \text{False}, \text{True}\rangle$
- Fuzzy CSPs: $\langle[0, 1], max, min, 0, 1\rangle$
- Weighted CSPs: $\langle[0, \ldots, +\infty], \text{min}, +, +\infty, 0\rangle$

Named Constraint Semirings

- A *named c-semiring* is a c-semiring equipped with:
  - name fusions $x = y$ for all names x, y
  - a notion of support $\mathrm{supp}(c)$ for each element c
  - a hiding operator $(\nu x.)$ that makes x local in c
  - a set of axioms (ruling how to combine operations)
- A *named constraint* is just an element of the named c-semiring.

Example: functional constraints

- Let D be a domain for $\mathcal{N}$, a *functional constraint* is a function $c = (\mathcal{N} \to D) \to \{\text{True}, \text{False}\}$ (es. $x\eta = a, y\eta = b$)
- A named c-semiring for functional constraints is such that:
  - the elements are all functional constraints over $\mathcal{N}$ and D
  - $(c + d)\eta = c\eta \vee d\eta$ and $(c \times d)\eta = c\eta \wedge d\eta$
  - $0\eta = \mathit{False}$ and $1\eta = \mathit{True}$
  - $(\nu x.c)$ and $\rho\, c$ are as expected

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

Aims I

1. Providing a formal model for defining SLA contracts and for validating contracts at service execution.
2. Modelling complex negotiation scenarios (not just XML-templates).
3. Studying mechanisms for resource allocation and for combining different SLA requirements.

Aims II

Main Ingredients

The CC-Pi calculus is simple process calculus that:

- extends Pi_F by generalising explicit fusions to named constraints
- integrates cc-programming primitives (`ask` , `tell`)
- introduces new primitives for constraint handling (`retract` , `check`)

SLA Contract Scenario

- A server and client willing to reach an agreement are specified as cc-pi processes that add their own requirements and guarantees as constraints to (possibly, local) stores.
- The synchronisation of two processes results in the combination of their respective stores of constraints and may succeed or be stuck.

CcPi-Calculus (syntax)

- Cc-pi is parametric wrt named c-semirings (assume c ranges over constraints of an arbitrary named c-semiring)
- $x, y, z, \ldots$ range over $\mathcal{N}$; K ranges over a set of process identifiers.

PREFIXES	π	$::=$	$\tau \mid \overline{x}\langle\tilde{y}\rangle \mid x\langle\tilde{y}\rangle \mid \texttt{tell}\ c \mid$
			$\texttt{ask}\ c \mid \texttt{retract}\ c \mid \texttt{check}\ c$
UNCONSTRAINED PROC.	U	$::=$	$\mathbf{0} \mid U\vert U \mid \sum_i \pi_i.U_i \mid (x)U \mid K(\tilde{y})$
CONSTRAINED PROC.	P	$::=$	$U \mid c \mid P\vert P \mid (x)P$

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

CcPi-Calculus (semantics)

The structural axioms allow to put processes into a normal form

$$(x_1)\dots(x_n)(C \,|\, U)$$

with C a parallel composition of constraints and U an unconstrained process.

SOS rules

$$(\textsc{Tau})\quad C\,|\,\tau.U \to C\,|\,U \qquad (\textsc{Tell})\ C\,|\,\mathtt{tell}\ d.U \to C\,|\,d\,|\,U \text{ if } C\,|\,d \text{ consistent}$$

$$(\textsc{Ask})\ C\,|\,\mathtt{ask}\ d.U \to C\,|\,U \quad \text{if } C \vdash d \quad (\textsc{Retract})\ C\,|\,\mathtt{retract}\ d.U \to (C - d)\,|\,U$$

$$(\textsc{Check})\quad C\,|\,\mathtt{check}\ d.U \to C\,|\,U \text{ if } C\,|\,d \text{ consistent}$$

$$(\textsc{Com})\quad C\,|\,(\bar{x}\langle\tilde{y}\rangle.U + \textstyle\sum \pi_i.U_i)\,|\,(z\langle\tilde{w}\rangle.V + \textstyle\sum \pi'_j.V_j) \longrightarrow (C\,|\,\tilde{y} = \tilde{w})\,|\,U\,|\,V$$

$$\text{if } |\tilde{y}| = |\tilde{w}|,\ C\,|\,\tilde{y} = \tilde{w} \text{ consistent and } C \vdash x = z$$

$$(\textsc{Sum})\ \frac{C\,|\,\pi_i.U_i \to P}{C\,|\,\sum \pi_i.U_i \to P} \qquad (\textsc{Par})\ \frac{P \to P'}{P\,|\,U \to P'\,|\,U}$$

$$(\textsc{Res})\ \frac{P \to P'}{(x)P \to (x)P'} \qquad (\textsc{Struct})\ \frac{P \equiv P' \quad P' \to Q' \quad Q' \equiv Q}{P \to Q}$$

Roadmap

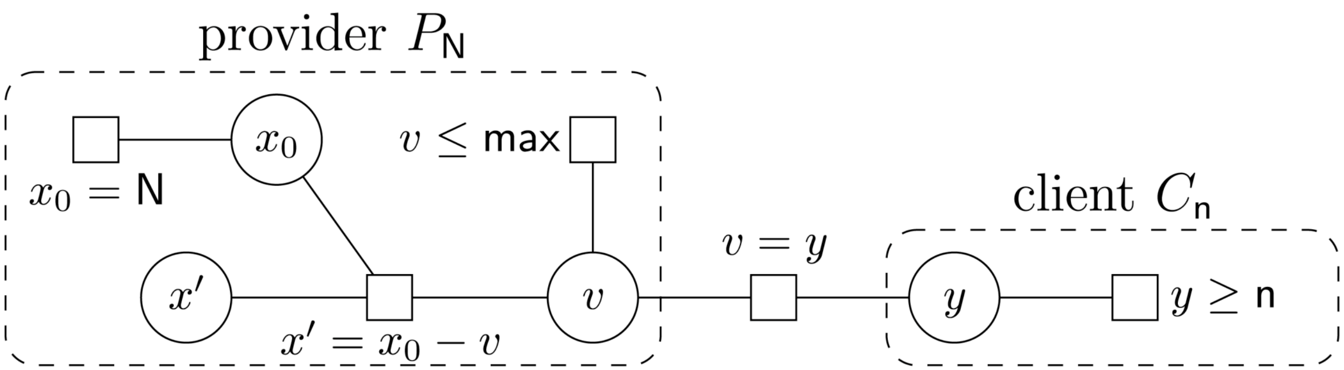
- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

CcPi-Calculus (example I)

Example 1

- Consider a service offering computing resources (e.g. units of CPUs)
- The provider P and a client C want to conclude a SLA contract.
- P_{N} (N available resources) and C_{n} (at least n resources) are as below

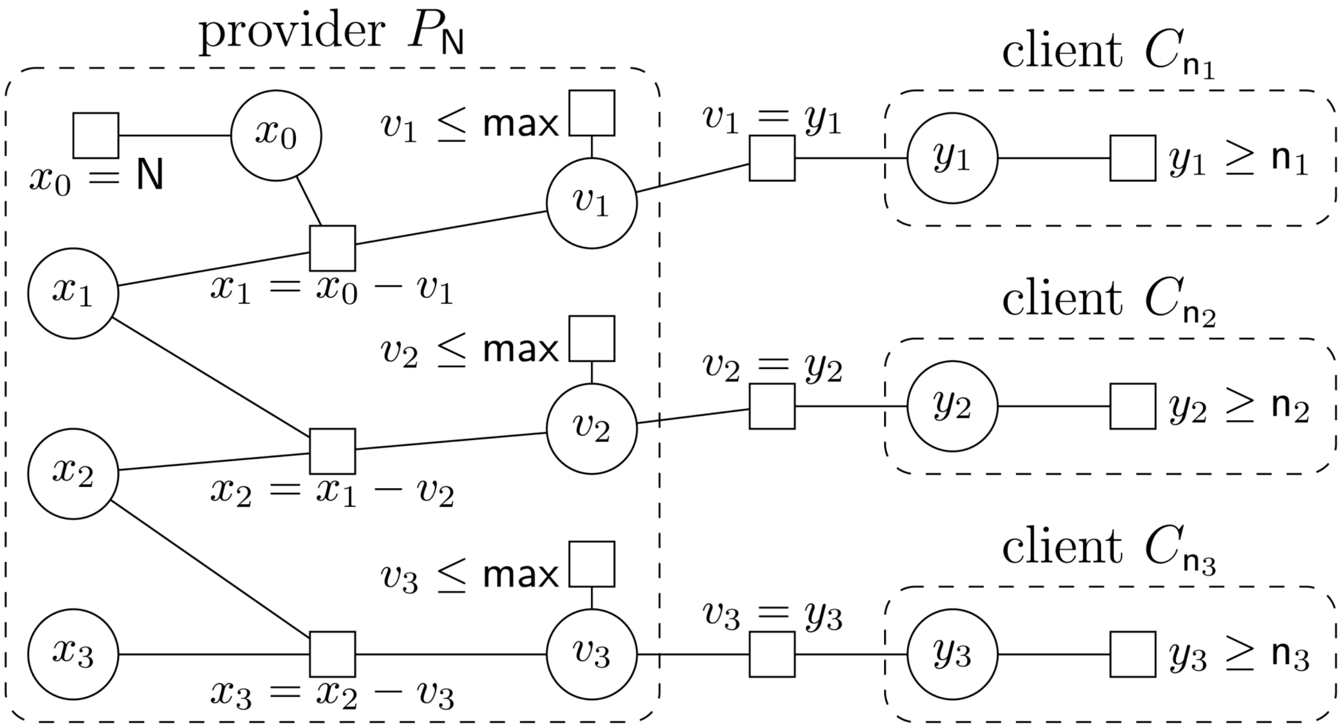
$$P_{\mathsf{N}} = (x_0)(\mathtt{tell}\ (x_0 = \mathsf{N}).Q(x_0))$$
$$Q(x) = (v)(x')(\mathtt{tell}\ (x' = x - v).\mathtt{tell}\ (v \leq \mathsf{max}).c\langle v\rangle.Q(x')).$$
$$C_{\mathsf{n}} = (y)(\mathtt{tell}\ (y \geq n).\overline{c}\langle y\rangle.\tau.\mathtt{retract}\ (y \geq n).\mathtt{tell}\ (y = 0)).$$



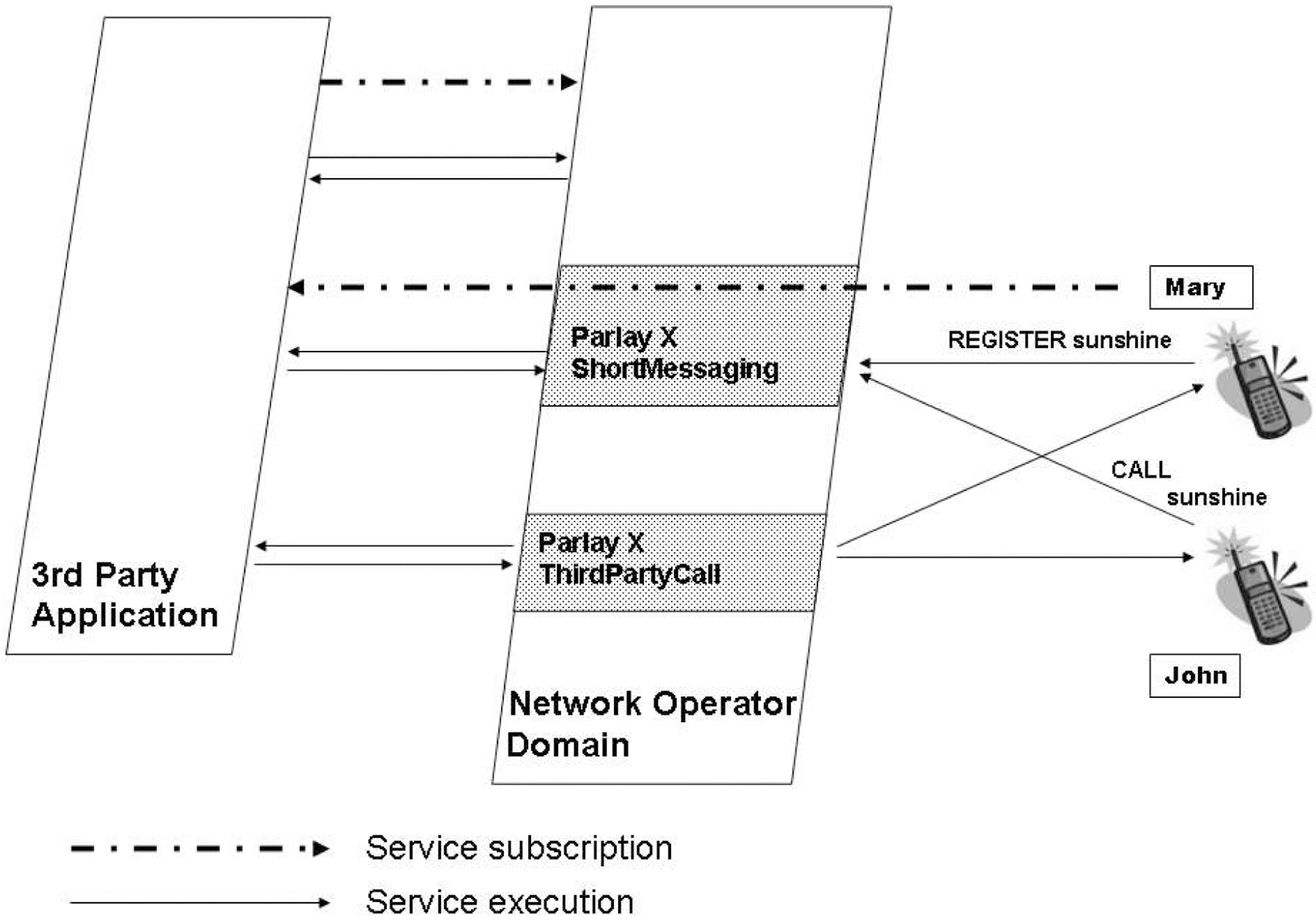
CcPi-Calculus (example II)

Example 2

A slightly more complex scenario with one provider P_{N} and three clients C_{n_1}, C_{n_2}, and C_{n_3}.



A CallBySms Service Scenario, I



A CallBySms Service Scenario, II

1. The Third Party application subscribes the services that are used by the CallBySms service and signs a SLA contract with the Network Operator;
2. The CallBySMS service is activated and the Third Party application receives a service number, e.g. 11111;
3. Mary sends an SMS "REGISTER sunshine" to the service number 11111;
4. The service associates "sunshine" to the opaque-id of Mary;
5. John sends an SMS "CALL sunshine" to the service number 11111;
6. The service retrieves the opaque-id associated to "sunshine" and set-up a call;
7. John's phone rings; John answers and gets the ringing tone;
8. Mary's phone rings; Mary answers;
9. John and Mary are connected.

CallBySms Specification in cc-pi

POLICIES

$$\begin{aligned}
c_{\text{time}} &= (\mathsf{7am} \leq i \leq \mathsf{9am}) \times (\mathsf{5pm} \leq f \leq \mathsf{9pm}) \\
c_{\text{freq}} &= nc \leq \mathsf{max_call} \\
d_{\text{time}} &= (\mathsf{6am} \leq i' \leq \mathsf{8am}) \times (\mathsf{4pm} \leq f' \leq \mathsf{6pm}) \\
d_{\text{freq}} &= (ncp' \leq \mathsf{call_per_pers}) \times (nr' \leq nc'/\mathsf{call_per_pers})
\end{aligned}$$

3RDPA-PARX NEGOTIATION

$$\begin{aligned}
\text{ParX_Neg} &= (i, f, nc, beg, end)\,(\texttt{tell}\ c_{\text{time}} \times c_{\text{freq}}.\, x\langle i, f, nc, beg, end\rangle.0) \\
\text{3rdPA_Neg} &= (i', f', ncp', nc', nr', beg', end')\,(\texttt{tell}\ d_{\text{time}} \times d_{\text{freq}}.\, \overline{x}\langle i', f', nc', beg', end'\rangle.0)
\end{aligned}$$

CLOCK

$$\begin{aligned}
\text{Clock} &= (t_0)\,(\texttt{tell}\ t = t_0.\text{Cl}(t, t_0)) \\
\text{Cl}(t, t') &= \texttt{retract}\ t = t'.\texttt{tell}\ t = t' + 1.\text{Cl}(t, t' + 1)
\end{aligned}$$

SERVICE EXECUTION

$$\begin{aligned}
\text{ParX_Ex} &= \texttt{check}\ (t = i).beg\langle\rangle.\text{ParX_Acpt_Reqst}.\texttt{check}\ (t = f).end\langle\rangle.0 \\
\text{3rdPA_Ex} &= \overline{beg'}\langle\rangle.\text{3rdPA_Acpt_Reqst}.\overline{end'}\langle\rangle.0
\end{aligned}$$

HANDLING REGISTRATION REQUESTS

$$\begin{aligned}
\text{Regist_User} &= (mary)\,(\overline{z}\langle mary, sunshine\rangle.\, mary\langle\rangle.\,\text{Wait_Calls}) \\
\text{ParX_Acpt_Reqst} &= (id, nn, ch)(z\langle id, nn\rangle.\overline{x}\langle nn, ch\rangle.\overline{id}\langle\rangle.(\text{ParX_Acpt_Reqst} \mid \text{ParX_Acpt_Call})) \\
\text{3rdPA_Acpt_Reqst} &= (nn', ncp, ch')\,(\texttt{check}\ (nr' \leq \mathsf{max_call}/\mathsf{call_per_pers}).\texttt{tell}\ (nr' = nr' + 1).x\langle nn', ch'\rangle). \\
&\quad (\text{3rdPA_Acpt_Reqst} \mid \text{3rdPA_Acpt_Call})
\end{aligned}$$

HANDLING CALL REQUESTS

$$\begin{aligned}
\text{Wait_Calls} &= (cal')\,(mary\langle cal'\rangle.cal'\langle\rangle.\text{Wait_Calls}) \\
\text{Caller} &= (john)\overline{sunshine}\langle john\rangle.\overline{john}\langle\rangle.0 \\
\text{ParX_Acpt_Call} &= (cal)\,(\texttt{check}\ (nc \leq \mathsf{max_call}).\texttt{tell}\ (nc = nc + 1).nn\langle cal\rangle.\overline{ch}\langle\rangle.\overline{id}\langle john\rangle.\text{ParX_Acpt_Call}) \\
\text{3rdPA_Acpt_Call} &= \texttt{check}\ (ncp' \leq \mathsf{call_per_pers}).\texttt{tell}\ (ncp' = ncp' + 1).ch'\langle\rangle.\text{3rdPA_Acpt_Call}
\end{aligned}$$

SYSTEM

$$S = (t, x, z)(\text{3rdPA\_Neg} \mid \text{ParX\_Neg} \mid \text{3rdPA\_Ex} \mid \text{ParX\_Ex} \mid \text{Caller} \mid \text{Regist\_User} \mid \text{Clock})$$

Roadmap

- Global computing
- Pi-calculus
- Constraint semirings
- Cc-pi: Syntax
- Cc-pi: Reduction semantics
- Cc-pi: Examples
- Conclusion and future work

Conclusion and Future Work

- Cc-Pi part of EU FET GC2 project Sensoria
- Reduction semantics at ESOP 2007 and symbolic semantics at ESOP 2008
- Names as keys for secure retract
- Efficient evaluation of constraints via locality restrictions and dynamic programming
- Extension to include behavioral types?
- Extension to handle assume - guarantee?
- Extension to handle ontologies?