Workflow modeling

Christian Stefansen

cstef@diku.dk

2003-06-18

Self-evident, adj.
Evident to one's self and to nobody else.

– Devil's Dictionary

Agenda

- The four universal flow constructs
- Motivating example for using a calculus
- Introducing the π-Calculus
- The π-Calculus meets REA in action
- Open issues
- Bonus if time: Formal stuff to impress friends with

The Four Universal Constructs

A brief look at these flow/process languages

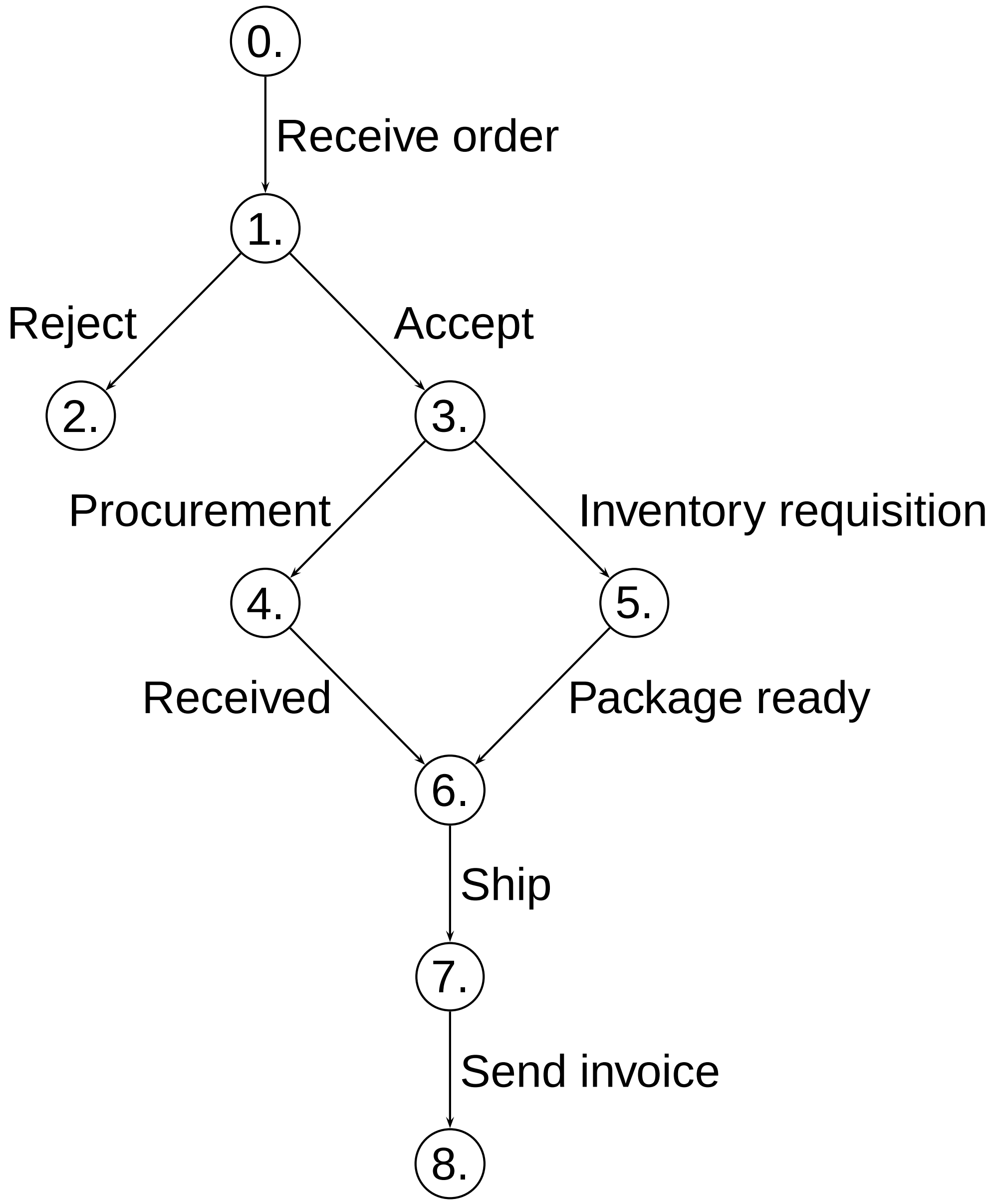
- BPEL (formerly XLANG and WSFL), ebXML, UML Activity Diagrams

supports the idea of four universal constructs:

- Sequence
- Parallelization
- Iteration/recursion
- Selection

(Taken from [9] where reference is given to Aalst [1].)

An Example Workflow



Each state may be an abstraction over an invocation.

π-Calculus Syntax[1]

The π-Calculus has the following syntax

P	$::=$	$\mathbf{0}$	(inaction)
	$\mid$	$P_1 + P_2$	(summation *i.e. choice*)
	$\mid$	$\overline{y}x.P$	(output prefix)
	$\mid$	$y(x).P$	(input prefix)
	$\mid$	$\tau.P$	(silent prefix)
	$\mid$	$P_1 \mid P_2$	(parallel composition)
	$\mid$	$(x)P$	(restriction)
	$\mid$	$!P$	(replication)

where

$x, y \in \mathcal{N}$	(names)
$P, Q, R, \ldots \in \mathcal{P}$	(process expressions)

$$\left.\begin{array}{r}\text{Restriction}\\ \text{Prefix}\\ \text{Replication}\end{array}\right\} > \text{Composition} > \text{Summation}$$

[1] This particular version of the syntax was shamelessly ganked from [5] and adapted slightly.

The Example Expressed in the π-Calculus

We use channels `order`, `inventory`, `procurement`, `shipping`, `invoice`.

Assume x carries customer identification, line items etc. and assume that appropriate subsystems to handle inventory, procurement, shipping, and invoicing exist.

$$
\begin{aligned}
\text{order}(x) \quad . \quad & (\overline{\text{order}}\,\texttt{decline} + \\
& (\overline{\text{order}}\,\texttt{accept} \\
& \quad .\,((\overline{\text{inventory}}\,x \mid \overline{\text{procurement}}\,x) \\
& \qquad \mid \text{inventory}(x').\,\text{procurement}(x'') \\
& \qquad\quad .\,\overline{\text{shipping}}\,x'.\,\overline{\text{shipping}}\,x'' \\
& \qquad\quad .\,\overline{\text{invoice}}\,x'.\,\overline{\text{invoice}}\,x'')))
\end{aligned}
$$

The calculus focuses on what is *observable* i.e. communication. Every transition produces *data*, whereas the state reflects *commitments* (the future).

Types of Accounting

Financial Accounting The process of producing financial statements for external constituencies—people outside the organization, such as shareholders, creditors, and governmental authorities. This process is heavily constrained by standard-setting, regulatory, and tax authorities and the auditing requirements of independent accountants (contrast with management accounting).

Management Accounting A value-adding improvement process of planning, designing, measuring, and operating nonfinancial and financial information systems that guides management action, motivates behavior, and supports and creates the cultural values necessary to achieve an organization's strategic, tactical, and operating objectives.

Atkinson, Banker, Kaplan, and Young [2]

The economic events and scarce economic ressources of the REA model lie primarily within the sphere of financial accounting.

Here every *observable* event in the firm is registered so the system lends itself very well to the broader concept of management accounting.

The Trivial Firm

Consider a company that only permits four simple events:

- Sell goods
- Receive money
- Buy goods
- Pay money

Setting

σ	sales	π	purchases
γ	goods	$\$$	money

yields the process expression

$$F = ((\overline{\sigma}_1\gamma \mid \sigma_2(\$)) \;.\; F) \mid ((\overline{\pi}_1\$|\pi_2(\gamma)) \;.\; F).$$

But something is fishy—what?

The Trivial Firm—Correct Version

Perhaps with $+$ instead? The expression

$$F = ((\overline{\sigma}_1\gamma \mid \sigma_2(\$)) + F) \mid ((\overline{\pi}_1\$|\pi_2(\gamma)) + F)$$

blocks. Once $\overline{\sigma}_1\gamma$, say, has occured, the inference rule of the operational semantics for sum

$$\frac{P \stackrel{\alpha}{\to} P'}{P + Q \stackrel{\alpha}{\to} P'}$$

states that the remaining $+F$ is discarded. So no other sale can be initiated. A more sound approach is

$$F = (\overline{\sigma}_1\gamma \mid \sigma_2(\$) + \overline{\pi}_1\$|\pi_2(\gamma)) \mid F$$

which allows an arbitrary number of simultaneous transactions in progress. By the definition of replication

$$!F \stackrel{\text{new}}{\curvearrowright\!\!=} F \mid !F$$

it simplifies to

$$F = !(\overline{\sigma}_1\gamma \mid \sigma_2(\$) + \overline{\pi}_1\$|\pi_2(\gamma)).$$

The Fundamental Theorem of Accounting[2]

Assets = Liabilities + Equity

This is our minimal demand: To derive an income statement and a balance that obey this invariant.

Let's take a few transaction examples for the firm:

1. Receives a cookie worth $1
2. Receives an icecream worth $3
3. Sells a cookie for $2
4. Sells an icecream for $4
5. Receives money for cookie $2
6. Pays for icecream $3

   (Income period ends)

[2]Often equity is considered part of the liabilities giving the simpler expression Assets = Liabilities

Operational Semantics [6]

TAU-ACT : $\dfrac{-}{\tau.P \xrightarrow{\tau} P}$ OUTPUT-ACT : $\dfrac{-}{\overline{x}y.P \xrightarrow{\overline{x}y} P}$

INPUT-ACT : $\dfrac{-}{x(z).P \xrightarrow{x(w)} P\{w/z\}}$ $w \notin \mathrm{fn}((z)P)$

SUM : $\dfrac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$ MATCH : $\dfrac{P \xrightarrow{\alpha} P'}{[x=x]P \xrightarrow{\alpha} P'}$

IDE : $\dfrac{P\{\tilde{y}/\tilde{x}\} \xrightarrow{\alpha} P'}{A(\tilde{y}) \xrightarrow{\alpha} P'}$ $A(\tilde{x}) \overset{\mathrm{def}}{=} P$

PAR : $\dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$ $\mathrm{bn}(\alpha) \cap \mathrm{fn}(Q) = \emptyset$

COM : $\dfrac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{y/z\}}$ CLOSE : $\dfrac{P \xrightarrow{\overline{x}(w)} P' \quad Q \xrightarrow{x(w)} Q'}{P \mid Q \xrightarrow{\tau} (w)(P' \mid Q')}$

RES : $\dfrac{P \xrightarrow{\alpha} P'}{(y)P \xrightarrow{\alpha} (y)P'}$ $y \notin \mathrm{n}(\alpha)$ OPEN : $\dfrac{P \xrightarrow{\overline{x}y} P'}{(y)P \xrightarrow{\overline{x}(w)} P'\{w/y\}}$ $y \neq x$, $w \notin \mathrm{fn}((y)P')$

Table 2: Rules of Action. Rules involving the binary operators $+$ and $\mid$ additionally have symmetric forms.

9

Transaction Examples

Let $\gamma_c =$ cookie and $\gamma_i =$ icecream ingoing.
Let $\gamma_c' =$ cookie and $\gamma_i' =$ icecream outgoing.

$$
\begin{array}{lcl}
F & = & !((\overline{\sigma}_1\gamma \mid \sigma_2(\$)) \mid (\overline{\pi}_1\$|\pi_2(\gamma))) \\
F & \stackrel{\pi_1(\gamma_c)}{\rightarrow} & \overline{\pi}_1\$ \mid F \\
 & \stackrel{\pi_2(\gamma_i)}{\rightarrow} & \overline{\pi}_1\$ \mid \overline{\pi}_2\$ \mid F \\
 & \stackrel{\overline{\sigma}_3\gamma_c'}{\rightarrow} & \overline{\pi}_1\$ \mid \overline{\pi}_2\$ \mid \sigma_3(\$) \mid F \\
 & \stackrel{\overline{\sigma}_4\gamma_i'}{\rightarrow} & \overline{\pi}_1\$ \mid \overline{\pi}_2\$ \mid \sigma_3(\$) \mid \sigma_4(\$) \mid F \\
 & \stackrel{\sigma_3(\$)}{\rightarrow} & \overline{\pi}_1\$ \mid \overline{\pi}_2\$ \mid \sigma_4(\$) \mid F \\
 & \stackrel{\overline{\pi}_2\$}{\rightarrow} & \overline{\pi}_1\$ \mid \sigma_4(\$) \mid F
\end{array}
$$

The trace is $\pi_1(\gamma_c), \pi_2(\gamma_i), \overline{\sigma}_3\gamma_c', \overline{\sigma}_4\gamma_i', \sigma_3(\$), \overline{\pi}_2\$$. Each transition has produced REA data. The commitments are reflected in the state $\overline{\pi}_1\$ \mid \sigma_4(\$) \mid F$.

Now the income statement and the balance sheet can be derived...

Deriving the General Ledger

Every trace member gives rise to one transaction line. For now, this piece of pseudo-code for the trace consolidation function θ will suffice:

$$\begin{aligned} \theta(\_(\$)) &\rightarrow \text{Petty cash} \\ \theta(\overline{\alpha}\ x) &\rightarrow \text{Income} \\ \theta(\alpha(x)) &\rightarrow \text{Expenses} \end{aligned}$$

Notice that all outputs are credited and all inputs are debited.

Using a valuation function υ, every commitment contributes to the balance sheet. The states are mapped in a similar fashion to A/R and A/P respectively.

$$\begin{aligned} \upsilon(\overline{\alpha}\ x) &\rightarrow \text{A/P} \\ \upsilon(\alpha(x)) &\rightarrow \text{A/R} \end{aligned}$$

Income Statement and Balance Sheet

Very simple, stylized general ledger. Balance sheet:

Assets

A/R	
(st) $4	

Petty cash	
(5) $2	
	(6) $3

Liabilities

A/P	
	(st) $1

Equity	

Income statement:

Expenses	
(1) $1	
(2) $3	

Income	
	(3) $2
	(4) $4

The gross profit of $2 is transferred to equity, and we have Assets$(\$3)$ $=$ Liabilities$(\$1)$ $+$ Equity$(\$2)$.

Open Issues

What calculus is best suited? The π-Calculus lacks good constructs for sequentiality, leaves “stuck states”/deadlocks, is data-unaware, and perhaps too general.

What is a channel? Ontologically speaking

Soundness For what subset of process expressions do θ and υ produce valid financial statements?

Duality How can the duality constraint be enforced?

Timing How should timing be introduced?

Policies How does one express policies? Model checking?

Valuation How should valuation be done? What is $\upsilon(P + Q)$?

Exceptions What if a customer never pays? Should all such contingencies be expressed explicitly in the process expression?

Contact

Christian Stefansen
N208, DIKU
cstef@diku.dk
http://www.stefansen.dk

Bonus Stuff—The Expansion Law (simplified version [8])

For the mathematically inclined it may be interesting to notice that

$$P|Q \quad = \quad \sum_{P \stackrel{\mu}{\to} P'} \mu.(P'|Q) + \sum_{Q \stackrel{\mu}{\to} Q'} \mu.(P|Q') + \sum_{\substack{P \stackrel{\alpha}{\to} P' \\ Q \stackrel{\overline{\alpha}}{\to} Q'}} \tau.(P'|Q').$$

Hence composition can be expressed using summation by the expansion law. It is sometimes easier to realize that this identity actually holds by considering the simply indentity $a|b = a.b + b.a$ first.

Bonus Stuff—π-Calculus Variations

One may elect to restrict the syntax to *guarded* sums i.e. $\sum_{i \in I} \alpha_i.P_i$ because it reduces the practical problem of omniscient global decisions and thus simplifies the theory somewhat ([7] p. 17). Match can be expressed using the other constructs [5]. If we take π to range over prefixes a considerably simpler syntax can be obtained as:

$$
\begin{array}{rcll}
\pi & ::= & x(y) & \text{(input)} \\
 & | & \overline{x}y & \text{(output)} \\
 & | & \tau & \text{(silent)}
\end{array}
$$

$$
\begin{array}{rcll}
P & ::= & \sum_{i \in I} \alpha_i.P_i & \text{(summation } \textit{i.e. choice}\text{)} \\
 & | & P_1 | P_2 & \text{(parallel composition)} \\
 & | & (x)P & \text{(restriction)} \\
 & | & !P & \text{(replication)}
\end{array}
$$

The historic development of the π-Calculus:

Relabelling $\rightarrow$ Defined agents $\rightarrow$ Replication

References

[1] W. van der Aalst and K. van Hee: *Workflow Management*, MIT Press, 2002

[2] Atkinson, Banker, Kaplan, Young: *Management Accounting*, third international ed., Prentice Hall, 2001

[3] *BPEL (Business Process Execution Language)* `ftp://www6.software.ibm.com/software/developer/library/ws-bpel11.pdf`

[4] *ebXML* `http://www.ebxml.org/specs/ebBPSS.pdf`

[5] Robin Milner and Joachim Parrow and David Walker: *A Calculus of Mobile Processes, Part I*, `http://citeseer.nj.nec.com/article/ milner89calculus.html`

[6] Robin Milner and Joachim Parrow and David Walker: *A Calculus of Mobile Processes, Part II*, `http://citeseer.nj.nec.com/article/ milner89calculus.html`

[7] Joackim Parrow: *An Introduction to the π-Calculus*, `http://www.ifi.uio.no/dbsem/foils2003vaar/introPiCalculus.pdf`

[8] David Sands: *Models of Concurrent Computation*, handwritten lecture notes from a course given at DIKU

[9] Ole Bue Lond and Jakob Rauhe: *Accommodating*

Change in Enterprise Applications, thesis, IT-University of Copenhagen