On Compensation Primitives as Adaptable Processes

Jovana Dedeić
University of Novi Sad
Jovanka Pantović (Novi Sad) and
Jorge A. Pérez (Groningen)

LAP 2015 - Dubrovnik, September 24, 2015

Outline

Introduction

- Many distributed software applications exploit long-running transactions (LRTs).
- One particularly delicate aspect of LRTs management is handling (partial) failures

Introduction

- Many distributed software applications exploit long-running transactions (LRTs).
- One particularly delicate aspect of LRTs management is handling (partial) failures
- The last decade has seen the emergence of specialized constructs, such as *exceptions* and *compensations*.
- In this work, we study process calculi with constructs for compensations.

Compensable Processes (CPs)

- Several calculi with compensations have been proposed. The calculus of Lanese et al (ESOP'10) extends the π-calculus with:

$t[P\,,Q]$	$\langle Q \rangle$	$\mathtt{inst}\lfloor \lambda X.R \rfloor.P$
transaction scopes	protected blocks	compensation updates

Also, output prefixes $\bar{t}$ represent abortion signals.

Compensable Processes (CPs)

- Several calculi with compensations have been proposed. The calculus of Lanese et al (ESOP'10) extends the π-calculus with:

$t[P\,,Q]$	$\langle Q \rangle$	$\mathtt{inst}\lfloor \lambda X.R \rfloor.P$
transaction scopes	protected blocks	compensation updates

  Also, output prefixes $\bar{t}$ represent abortion signals.

- Several Labeled Transition Systems (LTS) define semantics. Example: given process

$$P = t[t_1[P_1\,,Q_1] \mid R \mid \langle P_2 \rangle\,,Q_2] \mid \bar{t}$$

In the discarding semantics, we have $P \xrightarrow{\tau}_{\mathtt{D}} \langle P_2 \rangle \mid \langle Q_2 \rangle$.

Adaptable Processes (APs)

- A process calculi approach to evolvability, in a broad sense. Proposed by Bravetti et al (FORTE'11, LMCS'12).
- Studied from several perspectives, e.g., expressiveness, decidability/verification, session types (SAC'13, WSFM'14).
- Runtime modifications to (located) process behaviors, upon exceptional circumstances – not necessarily negative.
- Simple formulation: higher-order process passing.

Adaptable Processes (APs)

- The calculus extends CCS with locations $l, l', \ldots$ and

$l\{(X).Q\}$	$l[P]$
update prefix	located process

Adaptable Processes (APs)

- The calculus extends CCS with locations $l, l', \ldots$ and

$l\{(X).Q\}$	$l[P]$
update prefix	located process

  These two constructs are meant to synchronize.
- Located processes are transparent and can be arbitrarily nested. This is useful to structure processes into hierarchies.

Adaptable Processes (APs)

- The calculus extends CCS with locations $l, l', \ldots$ and

$l\{(X).Q\}$ update prefix | $l[P]$ located process

These two constructs are meant to synchronize.

- Located processes are transparent and can be arbitrarily nested. This is useful to structure processes into hierarchies.

- Simple reduction semantics (C,D and E are evaluation contexts):

$$E\Big[C[\overline{a}.P] \mid D[a.Q]\Big] \rightarrow E\Big[C[P] \mid D[Q]\Big]$$

$$E\Big[C[l[P]] \mid D[l\{(X).Q\}.R]\Big] \rightarrow E\Big[C[Q\{^{P}/_{X}\}] \mid D[R]\Big]$$

CPs and APs: Similarities and Differences

- A transaction scope reacts to an abortion signal (an output) by removing the default behavior and running its compensation. Assuming no protected blocks in P we have:

$$\bar{t} \mid t[P \, , Q] \xrightarrow{\tau}_{\mathtt{D}} \langle Q \rangle$$

- Similarly, a process located at l reacts to a synchronization with an update prefix for l. Assuming $X \notin fv(Q)$, we have:

$$l[P] \mid l\{(X).Q\} \rightarrow Q$$

CPs and APs: Similarities and Differences

- A transaction scope reacts to an abortion signal (an output) by removing the default behavior and running its compensation. Assuming no protected blocks in P we have:

$$\bar{t} \mid t[P\,,Q] \xrightarrow{\tau}_{\mathtt{D}} \langle Q \rangle$$

- Similarly, a process located at l reacts to a synchronization with an update prefix for l. Assuming $X \notin fv(Q)$, we have:

$$l[P] \mid l\{(X).Q\} \rightarrow Q$$

Some differences:

1. In CPs, a transaction scope couples a default behavior and its associated compensation. In APs, update prefixes and located processes are defined separately.
2. In APs, there is no notion of protected block.

Our Contribution: Encoding CPs into APs

- We have encoded CPs (with different semantics for failure) into APs.
- Our encodings not only are a non trivial application of process mobility. They shed light on the intricate semantics of compensable processes.
- The main challenge to encodability is in representing the different failure semantics using adaptable process.

Our Contribution: Motivation

Our motivation is twofold.

First

Understanding how different semantics for compensable processes can be **uniformly** implemented as adaptable processes.

Second

Our encodings could enable the **transference** of, e.g., decidability results or type systems from adaptable processes to calculi with compensations.

Compensable Processes end Adaptable Processes

Both CPs and APs are defined as variants of CCS.

- The syntax of the calculus of compensable processes.

$$\begin{array}{rcl} \pi & ::= & a \mid \overline{a} \\ P, Q & ::= & \mathbf{0} \mid \pi.P \mid !P \mid (\nu a)P \mid P \mid Q \mid t[P, Q] \mid \langle Q \rangle \\ & \mid & X \mid \texttt{inst}\lfloor \lambda X.R \rfloor.P \end{array}$$

- The syntax of the calculus of adaptable processes.

$$\begin{array}{rcl} \pi & ::= & a \mid \overline{a} \mid l\{(X).Q\} \\ P & ::= & \mathbf{0} \mid \pi.P \mid !P \mid P \mid Q \mid (\nu a)P \mid l[P] \mid X \end{array}$$

The Encoding: Basic Intuition

We roughly encode protected blocks and transactions as:

$$\begin{array}{rcl} [\![\langle R\rangle]\!]_{t,\rho} & = & p_{t,\rho}\big[\,[\![R]\!]_{\epsilon}\,\big] \\ [\![t[P\,,Q]]\!]_{\rho} & = & \underbrace{t\big[\,[\![P]\!]_{t,\rho}\,\big]}_{\text{(a)}} \mid \underbrace{l_t.\pi_1.\cdots.\pi_k.p_t\big[\,[\![Q]\!]_{t,\rho}\,\big]}_{\text{(b)}} \mid \underbrace{t.\overline{l_t}.K}_{\text{(c)}} \end{array}$$

- Paths ρ describe the structure of nested transactions
- Protected blocks are placed in designated locations p_t.
- Part (a) is a located process encoding the default activity
- Part (b) represents the compensation activity and is protected by special prefixes $(\pi_1.\cdots.\pi_k)$.
- Part (c) handles abortion signals, collecting protected blocks.

  It is meant to consume prefixes $\pi_1,\cdots,\pi_k$.

The Encoding: Discarding Semantics

Let P be a compensable process and let ρ be a path.
The encoding $\mathtt{D}[\![\cdot]\!]_\rho$ of compensable processes into adaptable processes is defined as follows:

$$
\begin{array}{rcl}
\mathtt{D}[\![\langle P\rangle]\!]_\rho & = & p_\rho\left[\mathtt{D}[\![P]\!]_\epsilon\right] \\
\mathtt{D}[\![t[P\,,Q]]\!]_\rho & = & t\left[\mathtt{D}[\![P]\!]_{t,\rho}\right] \mid \mathtt{D}\|Q\|_{t,\rho}^{\mathtt{npb}_{\mathtt{D}}(P)} \mid t.\overline{l_t}.k_t.\mathbf{0} \\
\mathtt{D}[\![\mathbf{0}]\!]_\rho & = & \mathbf{0} \\
\mathtt{D}[\![P_1 \mid P_2]\!]_\rho & = & \mathtt{D}[\![P_1]\!]_\rho \mid \mathtt{D}[\![P_2]\!]_\rho \\
\mathtt{D}[\![\pi.P]\!]_\rho & = & \pi.\mathtt{D}[\![P]\!]_\rho \\
\mathtt{D}[\![!\,P]\!]_\rho & = & !\,\mathtt{D}[\![P]\!]_\rho \\
\mathtt{D}[\![(\nu a)P]\!]_\rho & = & (\nu a)\mathtt{D}[\![P]\!]_\rho
\end{array}
$$

The Encoding: By Example

Let $P_0 = t[R \mid \langle P \rangle \, , Q] \mid \bar{t}$ be a CP with $\mathtt{npb}_{\mathtt{D}}(R) = 0$.
Then $P_0 \xrightarrow{\tau}_{\mathtt{D}} \langle P \rangle \mid \langle Q \rangle$. We obtain:

$$\mathtt{D}[\![P_0]\!]_\epsilon \quad = \quad t\Big[\mathtt{D}[\![R \mid \langle P \rangle]\!]_{t,\epsilon}\Big] \mid \mathtt{D}\|Q\|^1_{t,\epsilon} \mid t.\overline{l_t}.k_t \mid \bar{t}$$

The Encoding: By Example

Let $P_0 = t[R \mid \langle P \rangle , Q] \mid \overline{t}$ be a CP with $\mathtt{npb}_{\mathtt{D}}(R) = 0$.
Then $P_0 \xrightarrow{\tau}_{\mathtt{D}} \langle P \rangle \mid \langle Q \rangle$. We obtain:

$$\begin{aligned}
\mathtt{D}[\![P_0]\!]_\epsilon &= t\Big[\mathtt{D}[\![R \mid \langle P \rangle]\!]_{t,\epsilon}\Big] \mid \mathtt{D}\|Q\|^1_{t,\epsilon} \mid t.\overline{l_t}.k_t \mid \overline{t} \\
&= t\Big[\mathtt{D}[\![R]\!]_{t,\epsilon} \mid p_{t,\epsilon}\big[\mathtt{D}[\![P]\!]_\epsilon\big]\Big] \\
&\quad \mid l_t.p_{t,\epsilon}\Big\{(X).z\{p_\epsilon[X] \mid \overline{m_t}.p_\epsilon\big[\mathtt{D}[\![Q]\!]_\epsilon\big]\}\Big\} \\
&\quad .(z[\mathbf{0}] \mid m_t.\overline{k_t}.t\{\dagger\}) \mid t.\overline{l_t}.k_t \mid \overline{t}
\end{aligned}$$

The Encoding: By Example

Let $P_0 = t[R \mid \langle P \rangle, Q] \mid \bar{t}$ be a CP with $\mathtt{npb}_{\mathtt{D}}(R) = 0$.
Then $P_0 \xrightarrow{\tau}_{\mathtt{D}} \langle P \rangle \mid \langle Q \rangle$. We obtain:

$$
\begin{aligned}
\mathtt{D}[\![P_0]\!]_\epsilon &= t\Big[\mathtt{D}[\![R \mid \langle P \rangle]\!]_{t,\epsilon}\Big] \mid \mathtt{D}\|Q\|^1_{t,\epsilon} \mid t.\overline{l_t}.k_t \mid \bar{t} \\
&= t\Big[\mathtt{D}[\![R]\!]_{t,\epsilon} \mid p_{t,\epsilon}\big[\mathtt{D}[\![P]\!]_\epsilon\big]\Big] \\
&\quad \mid l_t.p_{t,\epsilon}\Big\{(X).z\{p_\epsilon[X] \mid \overline{m_t}.p_\epsilon\big[\mathtt{D}[\![Q]\!]_\epsilon\big]\}\Big\} \\
&\quad .(z[\mathbf{0}] \mid m_t.\overline{k_t}.t\{\dagger\}) \mid t.\overline{l_t}.k_t \mid \bar{t} \\
&\to^* t\Big[\mathtt{D}[\![R]\!]_{t,\epsilon} \mid z\{p_\epsilon[\mathtt{D}[\![P]\!]_\epsilon] \mid \overline{m_t}.p_\epsilon\big[\mathtt{D}[\![Q]\!]_\epsilon\big]\}\Big] \\
&\quad \mid z[\mathbf{0}] \mid m_t.\overline{k_t}.t\{(Y).\mathbf{0}\} \mid k_t
\end{aligned}
$$

The Encoding: By Example

Let $P_0 = t[R \mid \langle P \rangle, Q] \mid \overline{t}$ be a CP with $\mathtt{npb}_{\mathtt{D}}(R) = 0$.
Then $P_0 \xrightarrow{\tau}_{\mathtt{D}} \langle P \rangle \mid \langle Q \rangle$. We obtain:

$$
\begin{array}{rcl}
\mathtt{D}\llbracket P_0 \rrbracket_{\epsilon} & = & t\Big[\mathtt{D}\llbracket R \mid \langle P \rangle \rrbracket_{t,\epsilon}\Big] \mid \mathtt{D}\|Q\|^{1}_{t,\epsilon} \mid t.\overline{l_t}.k_t \mid \overline{t} \\
& = & t\Big[\mathtt{D}\llbracket R \rrbracket_{t,\epsilon} \mid p_{t,\epsilon}\big[\mathtt{D}\llbracket P \rrbracket_{\epsilon}\big]\Big] \\
& & \mid l_t.p_{t,\epsilon}\Big\{(X).z\big\{p_{\epsilon}[X] \mid \overline{m_t}.p_{\epsilon}\big[\mathtt{D}\llbracket Q \rrbracket_{\epsilon}\big]\big\}\Big\} \\
& & .(z[\mathbf{0}] \mid m_t.\overline{k_t}.t\{\dagger\}) \mid t.\overline{l_t}.k_t \mid \overline{t} \\
& \rightarrow^{*} & t\Big[\mathtt{D}\llbracket R \rrbracket_{t,\epsilon} \mid z\big\{p_{\epsilon}[\mathtt{D}\llbracket P \rrbracket_{\epsilon}] \mid \overline{m_t}.p_{\epsilon}\big[\mathtt{D}\llbracket Q \rrbracket_{\epsilon}\big]\big\}\Big] \\
& & \mid z[\mathbf{0}] \mid m_t.\overline{k_t}.t\big\{(Y).\mathbf{0}\big\} \mid k_t \\
& \rightarrow^{*} & p_{\epsilon}\big[\mathtt{D}\llbracket P \rrbracket_{\epsilon}\big] \mid p_{\epsilon}\big[\mathtt{D}\llbracket Q \rrbracket_{\epsilon}\big] = \mathtt{D}\llbracket \langle P \rangle \mid \langle Q \rangle \rrbracket_{\epsilon}
\end{array}
$$

The Encoding: Operational Correspondence

Theorem

Let P be a compensable process and let ρ be a path.

a) If $P \xrightarrow{\tau}_{\mathtt{D}} P'$ then $\mathtt{D}[\![P]\!]_\rho \rightarrow^* \mathtt{D}[\![P']\!]_\rho$

b) If $\mathtt{D}[\![P]\!]_\rho \rightarrow Q$ then $\exists P'$ s.t. $P \xrightarrow{\tau}_{\mathtt{D}} P'$ and $Q \rightarrow^* \mathtt{D}[\![P']\!]_\rho$.

Further Results in the paper

We have described CPs with static recovery (no compensation updates $\texttt{inst}\lfloor \lambda X.R \rfloor .P$) and discarding semantics.

Further Results in the paper

We have described CPs with static recovery (no compensation updates $\texttt{inst}\lfloor \lambda X.R \rfloor .P$) and discarding semantics.

- In the paper we also consider encodings for CPs with two further failure semantics: preserving and aborting.

Further Results in the paper

We have described CPs with static recovery (no compensation updates $\texttt{inst}\lfloor \lambda X.R \rfloor .P$) and discarding semantics.

- In the paper we also consider encodings for CPs with two further failure semantics: preserving and aborting.
- Example: let P be a CP $t[t_1[P_1\,,Q_1] \mid R \mid \langle P_2 \rangle\,,Q_2] \mid \bar{t}$. In the preserving and aborting semantics, we have the internal transitions:

$$P \xrightarrow{\tau}_{\mathtt{P}} t_1[P_1\,,Q_1] \mid \langle P_2 \rangle \mid \langle Q_2 \rangle$$
$$P \xrightarrow{\tau}_{\mathtt{A}} \langle Q_1 \rangle \mid \langle P_2 \rangle \mid \langle Q_2 \rangle$$

The encodings into APs and proofs follow similar principles.

Further Results in the paper

We have described CPs with static recovery (no compensation updates $\texttt{inst}\lfloor\lambda X.R\rfloor.P$) and discarding semantics.

- In the paper we also consider encodings for CPs with two further failure semantics: preserving and aborting.
- Example: let P be a CP $t[t_1[P_1\,,Q_1] \mid R \mid \langle P_2\rangle\,,Q_2] \mid \bar{t}$.
  In the preserving and aborting semantics, we have the internal transitions:

$$P \xrightarrow{\tau}_{\texttt{P}} t_1[P_1\,,Q_1] \mid \langle P_2\rangle \mid \langle Q_2\rangle$$

$$P \xrightarrow{\tau}_{\texttt{A}} \langle Q_1\rangle \mid \langle P_2\rangle \mid \langle Q_2\rangle$$

  The encodings into APs and proofs follow similar principles.
- We also cover dynamic recovery, which includes compensation updates $\texttt{inst}\lfloor\lambda X.R\rfloor.P$.

The paper

- Jovana Dedeic, Jovanka Pantovic, Jorge A. Pérez: On Compensation Primitives as Adaptable Processes. EXPRESS/SOS 2015: 16-30

Future Plans

- We plan to consider the reverse direction of encoding.
- Cast our encodability results into a setting with session types:
  - The source language could be the typed calculus with interactional exceptions (Carbone et al, CONCUR'08)
  - The target language could be recently proposed extensions of adaptable processes with session types (SAC'13, WSFM'14).

On Compensation Primitives as Adaptable Processes

Jovana Dedeić
University of Novi Sad
Jovanka Pantović (Novi Sad) and
Jorge A. Pérez (Groningen)

LAP 2015 - Dubrovnik, September 24, 2015