A An Example of Endpoint Projection

In the following we illustrate the formal notion of endpoint projection we have developed in the paper using a fairly large toy example involving five participants. First, we explain the example in English; then we introduce the description in the global calculus; finally we project the description to endpoint processes.

A.1 Global Description in English

The example is an extension of the buyer-seller example introduced in section 2. The participants involved in this protocol are

1. Buyer (B)
2. Seller (S)
3. Vendor (V)
4. CreditChecker (CC)
5. RoyalMail (RM)

The protocol proceeds as follows:

1. Buyer requests a service ch_{CC} for company check to the credit checker CreditChecker by sending its name.
2. At this point CreditChecker can either give a positive or negative answer.
3. If the answer is positive:
   (a) Buyer asks Seller for a quote about product prod;
   (b) Seller then asks Vendor for service ch_V
   (c) Seller starts recursion and asks Vendor for a quote about product prod;
   (d) Vendor replies with a quote quote;
   (e) Seller forwards quote to Buyer increasing it by 10 units (quote+10);
   (f) if the quote is reasonable ($reasonable$(quote + 10)) then:
      i. Buyer sends Seller a confirmation (quoteOK) together with the credit (cred);
      ii. Seller then contacts CreditChecker for checking the credit;
      iii. If the credit is good then:
         A. Seller contacts Shipper (service ch_{Sh});
         B. Seller sends the delivery address;
         C. Shipper sends a confirmation;
         D. Seller forwards confirmation to Buyer;
      iv. If the credit is bad:
         A. CreditChecker tells Buyer;
         B. Buyer tells Seller terminating the protocol;
   (g) if the quote is not reasonable the protocol goes back to point 3c;
4. If the answer is negative then the protocol terminates.

A.2 Global Description in the Calculus

The global description consists of several components for readability. We directly give annotated interaction. The main description is:

1. $B^1 \to CC^2 : ch_{CC}(\nu\, s)$**.** $CC^2 \to B^1 : s\langle \mathsf{ack} \rangle$**.**
2. $B^1 \to CC^2 : s\langle \mathsf{companyCheck},\ \mathsf{sellerName},\ \mathsf{compName} \rangle$**.**
3. {
4. $\quad CC^2 \to B^1 : s\langle \mathsf{good} \rangle$**.** I_{good}
5. $\quad +$
6. $\quad CC^2 \to B^1 : s\langle \mathsf{bad} \rangle$**.** $\mathbf{0}$
7. }

where I_{good} in Line 4 is:

1. $B^1 \to S^3 : ch_S(\nu\, t)$**.** $S^3 \to B^1 : r\langle \mathsf{ack} \rangle$**.**
2. $B^1 \to S^3 : t\langle \mathsf{quoteReq},\ \mathsf{prod},\ \mathsf{prod} \rangle$**.**
3. $S^3 \to V^4 : ch_V(\nu\, r)$**.**
4. $V^4 \to S^3 : r\langle \mathsf{ack} \rangle$**.**
5. μX^3**.** {
6. $\quad S^3 \to V^4 : r\langle \mathsf{quoteReq},\ \mathsf{prod},\ \mathsf{prod} \rangle$**.**
7. $\quad V^4 \to S^3 : r\langle \mathsf{quoteRes},\ \mathsf{quote},\ \mathsf{quote} \rangle$**.**
8. $\quad S^3 \to B^1 : t\langle \mathsf{quoteRes},\ \mathsf{quote} + 10,\ \mathsf{quote} \rangle$**.**
9. $\quad$ if $reasonable(\mathsf{quote})@B^1$ then
10. $\quad\quad B^1 \to S^3 : t\langle \mathsf{quoteOK},\ \mathsf{cred},\ \mathsf{cred} \rangle$**.**
11. $\quad\quad S^3 \to CC^5 : ch_{CC}(\nu\, u)$**.**
12. $\quad\quad CC^5 \to S^3 : u\langle \mathsf{ack} \rangle$**.**
13. $\quad\quad S^3 \to CC^5 : u\langle \mathsf{personalCreditCheck},\ \mathsf{cred{:}adr},\ \mathsf{cred{:}adr} \rangle$**.**
14. $\quad\quad$ {
15. $\quad\quad\quad CC^5 \to S^3 : u\langle \mathsf{good} \rangle$**.** I'_{good}
16. $\quad\quad\quad +$
17. $\quad\quad\quad CC^5 \to S^3 : u\langle \mathsf{bad} \rangle$**.**
18. $\quad\quad\quad S^3 \to B^1 : t\langle \mathsf{yourCreditIsBad} \rangle$**.** $\mathbf{0}$
19. $\quad\quad$ }
20. $\quad$ else $B^1 \to S^3 : t\langle \mathsf{quoteNotOK} \rangle$**.** X^3
21. }

where I'_{good} in Line 15 is:

1. $S^3 \to R^6 : ch_R(\nu\, p)$**.**
2. $R^6 \to S^3 : p\langle \mathsf{ack} \rangle$**.**
3. $S^3 \to R^6 : p\langle \mathsf{deliv},\ \mathsf{adr},\ \mathsf{adr} \rangle$**.**
4. $R^6 \to S^3 : p\langle \mathsf{conf} \rangle$**.**
5. $S^3 \to B^1 : t\langle \mathsf{conf} \rangle$**.** $\mathbf{0}$

We can check these descriptions are typable, strongly connected, well-threaded and coherent. For connectedness, the descritption given above uses a lot of acks. As we discussed in the long version, many of these acks are in fact unnecessary by using a relaxed notion of connectedness.

A.3 End-Point Projection of the Global Interaction

Following the definition of EPP in the paper, we first project the global descrtiption onto each thread. The first one is Buyer's only thread.

$$
\begin{aligned}
\mathsf{TP}(I,1) = \overline{ch_{CC}}(\nu s).\ & s \triangleright \mathsf{ack}().\ \overline{s} \triangleleft \mathsf{companyCheck}\langle \mathsf{sellerName}\rangle. \\
& \{\ s \triangleright \mathsf{good}().\ \overline{ch_S}(\nu t).\ t \triangleright \mathsf{ack}().\ \overline{t} \triangleleft \mathsf{quoteReq}\langle \mathsf{prod}\rangle. \\
& \ \mu X.\ \ t \triangleright \mathsf{quoteRes}(\mathsf{quote}). \\
& \quad \mathsf{if}\ \mathit{reasonable}\{\mathsf{quote}\}\ \mathsf{then}\ \overline{t} \triangleleft \mathsf{quoteOK}\langle \mathsf{cred}\rangle. \\
& \qquad\qquad\qquad\qquad \{t \triangleright \mathsf{yourCreditIsBad}() \\
& \qquad\qquad\qquad\qquad\ + \\
& \qquad\qquad\qquad\qquad\ t \triangleright \mathsf{conf}()\} \\
& \quad \mathsf{else}\ \overline{t} \triangleleft \mathsf{quoteNoteOK}\langle\rangle.\ X \\
& + \\
& \ s \triangleright \mathsf{bad}().\ \mathbf{0}\ \ \}
\end{aligned}
$$

Note this thread starts before the recursion and go through inside the (global) recursion. Thus the projected endpoint behaviour also contains recursion.

The next projection is onto the first thread of CreditChecker (note this participant has two threads, 2 and 5).

$$
\begin{aligned}
\mathsf{TP}(I,2) = !\, ch_{CC}(s).\ & \overline{s} \triangleleft \mathsf{ack}\langle\rangle.\ s \triangleright \mathsf{companyCheck}(\mathsf{compName}). \\
& \{\ \overline{s} \triangleleft \mathsf{good}\langle\rangle. \\
& \ \oplus \\
& \ \overline{s} \triangleleft \mathsf{bad}\langle\rangle.\ \mathbf{0}\ \ \}
\end{aligned}
$$

Note no recursion is involved in this thread projection, simply because the thread 2 does not occur inside the recursion.

Next we jump to Thread 5, which is another component of CreditChecker.

$$
\begin{aligned}
\mathsf{TP}(I,5) = !\, ch_{CC}(u).\ & \overline{u} \triangleleft \mathsf{ack}\langle\rangle.\ u \triangleright \mathsf{personalCreditCheck}(\mathsf{cred{:}adr}) \\
& (\overline{u} \triangleleft \mathsf{good}\langle\rangle \\
& \ + \\
& \ \overline{u} \triangleleft \mathsf{bad}\langle\rangle
\end{aligned}
$$

Note the process does not include the recursion either. This is because it is inside a recursion and it initiates a new thread there. As a result the code is identical with the projection onto Thread 2.

We now move to the projection onto the unique thread of Seller, which is Thread 3.

$$
\begin{aligned}
\mathsf{TP}(I,3) = {} & !\, ch_S(t).\ \overline{t} \lhd \mathsf{ack}\langle\rangle.\ t \rhd \mathsf{quoteReq(prod)}.\ \overline{ch_V}(\nu r).\ t \rhd \mathsf{ack}().\\
& \mu X.\ \ \overline{r} \lhd \mathsf{quoteReq}\langle\mathsf{prod}\rangle.\ r \rhd \mathsf{quoteRes(quote)}.\\
& \quad \overline{t} \lhd \mathsf{quoteRes}\langle\mathsf{quote} + 10\rangle.\\
& \quad \{\ t \rhd \mathsf{quoteOK(cred)}.\ \overline{ch_C C}(\nu u).\ u \rhd \mathsf{ack}().\\
& \quad\ \ \overline{u} \lhd \mathsf{personalCreditCheck}\langle\mathsf{cred{:}adr}\rangle.\\
& \quad\ \ \ \{u \rhd \mathsf{good}().\ \overline{ch_R}(\nu p).\ p \rhd \mathsf{ack}()\\
& \quad\ \ \ \ \overline{p} \lhd \mathsf{deliv}\langle\mathsf{adr}\rangle.\ p \rhd \mathsf{conf}()\overline{t} \lhd \mathsf{conf}\langle\rangle\\
& \quad\ \ \ \ +\\
& \quad\ \ \ \ u \rhd \mathsf{bad}().\ \overline{t} \lhd \mathsf{CreditIsBad}\langle\rangle\ \}\\
& \quad\ \ +\\
& \quad\ \ t \rhd \mathsf{quoteNoteOK}().\ X
\end{aligned}
$$

As before, this thread starts outside of the recursion in the global description and is also used inside, so that both the recursion and the recursion variable are used as they are, leading to the recursive behaviour of the process. Note how the use of session functions as a way to handle recursion appropriately in EPP.

The projection onto the unique thread of Vendor follows.

$$
\begin{aligned}
\mathsf{TP}(I,4) = {} & !\, ch_V(r).\ \overline{t} \lhd \mathsf{ack}\langle\rangle.\\
& \mu X.\ r \rhd \mathsf{quoteReq(prod)}.\ \overline{r} \lhd \mathsf{QuoteRes}\langle\mathsf{quote}\rangle.\ X
\end{aligned}
$$

Finally we end with the projection onto Thread 6, giving the simple behaviour of RoyalMail.

$$
\mathsf{TP}(I,6) = !\, ch_R(p).\ \overline{p} \lhd \mathsf{ack}\langle\rangle.\ p \rhd \mathsf{deliv(adr)}.\ \overline{p} \lhd \mathsf{conf}\langle\rangle
$$

As before, Thread 6 does noont contain recursion since it is fully inside the (global) recursion, initiating a thread there.

As noted, there are two threads (2 and 5) that belong to the same class of equivalence i.e. they are part of the same service channel ch_{CC}. This means that we must merge the two threads in the final EPP. By applying the merge operator, and noting they are evidently mergeable, we get the following process:

$$
\begin{aligned}
& !\, ch_{CC}(u).\ \overline{u} \lhd \mathsf{ack}\langle\rangle.\\
& \qquad u \ \rhd \left(\begin{array}{l}
\mathsf{personalCreditCheck(cred{:}adr)}.\ (\overline{u} \lhd \mathsf{good}\langle\rangle \ \oplus\ \mathsf{bad}\langle\rangle)\\
\qquad\qquad +\\
\mathsf{companyCheck(compName)}.\ (\overline{u} \lhd \mathsf{good}\langle\rangle \ \oplus\ \mathsf{bad}\langle\rangle)
\end{array}\right)
\end{aligned}
$$

By which we have arrived at the endpoint behaviours of all participants realising the original global description.

The projection works because of the linear usage of channels inside each session and service channel principle, as well as the three well-structuredness conditions. We

believe many business protocols conform to these conditions (modulo relaxation of connectedness we discussed in the long version). How these conditions can be extended in disciplined ways to allow more "untamed" protocols (such as those involving exceptions) to be treated in the theory, is an interesting subject of further studies.