AnaCon: Case Studies

John J. Camilleri

john.j.camilleri@chalmers.se

16 February 2012

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-247914.

AnaCon: Case studies

1. Airline check-in process

   Fenech, Pace & Schneider, 2009

2. Internet Service Provider (ISP) contract

   Pace, Prisacariu & Schneider, 2007

Case study: ISP contract

Contract extract

1. The **Client** shall not supply false information to the Client Relations Department of the **Provider**.
2. Whenever the Internet Traffic is **high** then the **Client** must pay $[price]$ immediately, or the **Client** must notify the **Provider** by sending an e-mail specifying that he will pay later.
3. If the **Client** delays the payment as stipulated in Clause 2, after notification he must immediately lower the Internet traffic to the **normal** level, and pay later double $(2 * [price])$.
4. If the **Client** does not lower the Internet traffic immediately, then the **Client** will have to pay $3 * [price]$.
5. The **Client** shall, as soon as the Internet Service becomes operative, submit within seven (7) days the personal data form from his account on the **Provider**'s web page to the Client Relations Department of the **Provider**.
6. The **Provider** may, at its sole discretion, without notice or giving any reason or incurring any liability for doing so, suspend Internet Services immediately if the **Client** is in breach of Clause 1.

Original

1. The **Client** shall not supply false information to the Client Relations Department of the **Provider**.
6. The **Provider** may, at its sole discretion, without notice or giving any reason or incurring any liability for doing so, suspend Internet Services immediately if the **Client** is in breach of Clause 1.

CNL

```
{the Client} shall not provide {false information},
otherwise {the Provider} may suspend {the service
immediately}
```

$\mathcal{CL}$

$F(false)\_P(suspend)$

Original

2. Whenever the Internet Traffic is **high** then the **Client** must pay $[price]$ immediately, or the **Client** must notify the **Provider** by sending an e-mail specifying that he will pay later.
3. If the **Client** delays the payment as stipulated in Clause 2, after notification he must immediately lower the Internet traffic to the **normal** level, and pay later double $(2 * [price])$.

CNL

```
if {the traffic} becomes {high} then either
  - {the Client} must pay {price P}
  - first {the Client} must notify {the Provider by
  e-mail}, {the Client} must lower {the traffic to normal
  level}, then {the Client} must pay {price 2P}
```

$\mathcal{CL}$

$$[high]\,(\,O(pay1)\oplus O(notify.lower.pay2)\,)$$

Original

3. If the **Client** delays the payment as stipulated in Clause 2, after notification he must immediately lower the Internet traffic to the **normal** level, and pay later double $(2 * [price])$.
4. If the **Client** does not lower the Internet traffic immediately, then the **Client** will have to pay $3 * [price]$.

CNL

```
if {the traffic} becomes {high} then either
  - {the Client} must pay {price P}
  - both
    - {the Client} must notify {the Provider by e-mail}
    - first {the Client} must lower {the traffic to normal
    level}, then {the Client} must pay {price 2P},
    otherwise {the Client} is required to pay {price 3P}
```

$\mathcal{CL}$

$$[high]\Big(O(pay1) \oplus \big(O(notify) \wedge O(lower.pay2) \_ O(pay3)\big)\Big)$$

CNL

```
always each of
  - {the Client} shall not provide {false information},
  otherwise {the Provider} may suspend {the service
  immediately}
  - if {the traffic} becomes {high} then either
    - {the Client} must pay {price P}
    - both
      - {the Client} must notify {the Provider by e-mail}
      - first {the Client} must lower {the traffic to
      normal level}, then {the Client} must pay {price 2P},
      otherwise {the Client} is required to pay {price 3P}
  - if {the service} becomes {operative} then {the Client}
  shall submit {the personal data form within seven days}
```

CNL

```
always each of
  - {the Client} shall not provide {false information},
  otherwise {the Provider} may suspend {the service
  immediately}
  - if {the traffic} becomes {high} then either
    - {the Client} must pay {price P}
    - both
      - {the Client} must notify {the Provider by e-mail}
      - first {the Client} must lower {the traffic to
      normal level}, then {the Client} must pay {price 2P},
      otherwise {the Client} is required to pay {price 3P}
  - if {the service} becomes {operative} then {the Client}
  shall submit {the personal data form within seven days}
```

Problem

- $\mathcal{CL}$ does not allow disjunction over clauses.

CNL

```
always each of
  - {the Client} shall not provide {false information},
  otherwise {the Provider} may suspend {the service
  immediately}
  - if {the traffic} becomes {high} then {the Client} must
  pay {price P}, otherwise both
    - {the Client} must notify {the Provider by e-mail}
    - first {the Client} must lower {the traffic to normal
    level}, then {the Client} must pay {price 2P},
    otherwise {the Client} is required to pay {price 3P}
  - if {the service} becomes {operative} then {the Client}
  shall submit {the personal data form within seven days}
```

$\mathcal{CL}$

$$
\begin{array}{l}
[1*]\big(\\
\quad F(false) \_ P(suspend) \\
\quad \wedge\, [high]\, O(pay1) \_ \big(O(notify) \wedge O(lower.pay2) \_ O(pay3) \big) \\
\quad \wedge\, [oper]\, O(submit) \\
\big)
\end{array}
$$

XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contract>
  <clauses>
    <clause>[1*]((F(a7)_((P(a1))))^[a4]((O(a8)_((O(a2.b1.a9)
    _((O(a3)))))))^[a5]((O(a6))))</clause>
  </clauses>
  <concurrentActions>
    <action>a4#a5</action>
    <action>a1#a4</action>
    <action>a7#a4</action>
    <action>a8#a9</action>
  </concurrentActions>
</contract>
```

Dictionary

```
a1 = {the Provider} suspend {the service immediately}
a3 = {the Client} pay {price 3P}
a2 = {the Client} notify {the Provider by e-mail}
a5 = {the service} become {operative}
a4 = {the traffic} become {high}
a7 = {the Client} provide {false information}
a6 = {the Client} submit {the personal data form within
     seven days}
a9 = {the Client} pay {price 2P}
a8 = {the Client} pay {price P}
b1 = {the Client} lower {the traffic to normal level}
```

Output

```
473 counter examples found (only showing first)
Trace:
  1. traffic become high
  2. the Client provide false information and the service
  become operative
  3. the Client notify the Provider and the service become
  operative and the Client submit the personal data form
  4. the Client notify the Provider and the service become
  operative and the Client submit the personal data form
  5. the traffic become high and the Client lower traffic
  to the normal level and the Client submit the personal
  data form
```

Issues

- Natural sequence of events not reflected anywhere:

  Submit information $\rightarrow$ *Service operational* $\rightarrow$ *Traffic becoming high*

- Huge number of counter examples

CNL

```
{the Client} must submit {the data} ;
if {the Client} submits {the data} then each of
  - {the Provider} must check {the data}
  - if first {the Provider} checks {the data}, then {the
  Provider} disapproves {the data} then {the Provider} may
  cancel {the contract}
  - if first {the Provider} checks {the data}, then {the
  Provider} approves {the data} then each of
    - {the service} must become {operative}
    - if {the service} becomes {operative} then always if
    {the traffic} becomes {high} then {the Client} must pay
    {price P}, otherwise both
      - {the Client} must notify {the Provider by e-mail}
      - first {the Client} must lower {the traffic to
      normal level}, then {the Client} must pay {price 2P},
      otherwise {the Client} is required to pay {price 3P}
```

$\mathcal{CL}$

$$
\begin{array}{l}
O(submit) \wedge [submit](\\
\quad O(check) \\
\quad \wedge [check.dis]\, P(cancel) \\
\quad \wedge [check.app](\\
\quad\quad O(op) \\
\quad\quad \wedge [op][1*][high]\, O(pay1) \_ \big(O(notify) \wedge O(lower.pay2) \_ O(pay3)\big) \\
\quad) \\
)
\end{array}
$$

- Explicit handling of sequence
- Possibly different meaning to original contract
- Cross-product of all contradictory actions, minus key few

Output

```
18 counter examples found (only showing first)
Trace:
  1. the Client submit the data
  2. the Provider check the data
  3. the Provider approve the data
  4. the service become operative
  5. traffic become high
  6. traffic become high
  7. traffic become high and the Client pay price P and
     the Client notify the Provider by e-mail
  8. traffic become high and the Client pay price P and
     the Client notify the Provider by e-mail
  9. traffic become high and the Client pay price P and
     the Client lower traffic to the normal level
```

Issues

- *traffic become high* fires repeatedly

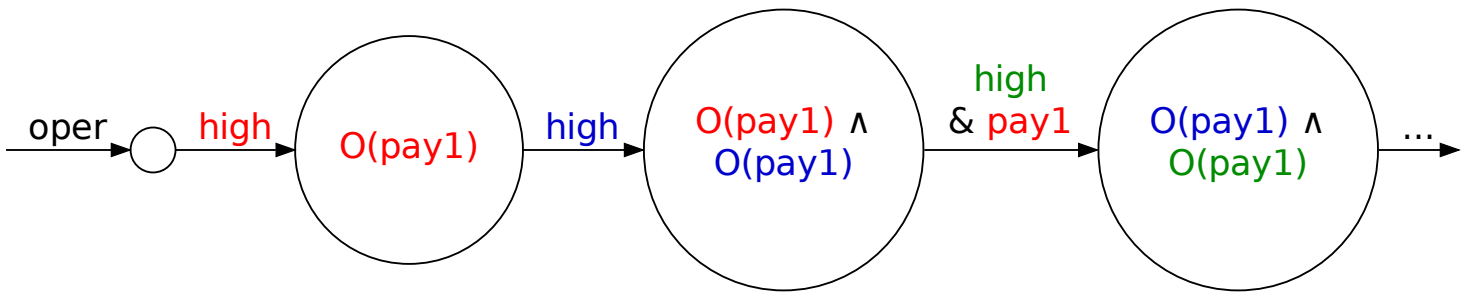
CNL

```
... always if {the traffic} becomes {high} then {the Client}
must pay {price P} ...
```

$\mathcal{CL}$

$$[1*][high]\,O(pay1)$$

Automaton



Issues

- Underspecification of *traffic become high*
- Lack of temporal conditions
- Repeat vs. sustained occurances/instances

Observations

- Knowledge of CNL and corresponding $\mathcal{CL}$ formula
- Declaration of causal/temporal relationships
- Only interested in *minimal subset* of contradictory traces
  - e.g. $high$, $high \,\&\, notify$, $high \,\&\, notify \,\&\, pay3$
  - ...eliminate $\&$ operator?

References

- S. Fenech, G. Pace, G. Schneider
  Automatic Conflict Detection on Contracts.
  ICTAC'09, Vol. 5684 of LNCS, Springer, 2009, pp. 200–214.
- G. Pace, C. Prisacariu, G. Schneider
  Model Checking Contracts — A Case Study.
  ATVA'07, Vol. 4762 of LNCS, Springer-Verlag, 2007, pp. 82–97.