Contracts in a State-rich Timed Process Algebra

Kun Wei, Jim Woodcock and Ana Cavalcanti
Department of Computer Science
University of York

February 26, 2013
Outline

- Introduction to *Circus* family
- *Circus Time*
  - Summary
  - Example: simple protocol
  - New *Circus Time* theory: alphabet, healthiness conditions and signature
  - The relation between the new and original theory
- Reactive designs
  - Brief introduction
  - Example: sequential composition
  - Changes for some operators
- Case Study: Miracle, deadlines and interrupts
- Conclusion and future work
Circus family

Circus is a combination of Z, CSP, guarded commands and refinement calculus, and has developed into a family of languages for specification, programming and verification. Its semantics is based on UTP.

- Circus Time and OhCircus (object-orientation)
- synchrony, mobility, Control Law Diagrams, Circus/Ada
- a number of tools but still need efficient tools for both theorem-proving and model-checking

Two major projects:

- hiJaC - formal development of SCJ
- COMPASS - model-based techniques for developing and maintaining Systems of Systems
Summary of *Circus Time*

- A discrete-time model
- Semantics is based on UTP
- $\mathbb{Z}+$Timed CSP+guarded commands
- More time operators than Timed CSP such as (hard) deadlines
- The semantics has not been mechanised, and so no tool support yet
- The original *Circus Time* (Adnan Sherif) is a synthesis of a UTP theory and a framework for transformation and time separation.
Example: simple protocol
Example: simple protocol

channel \textit{in, out, send} : \mathbb{Z}; \textit{enable, disable}

process \textit{Breqs} \triangleq \textup{begin}

state \textit{APState} == [col : \mathbb{P} \text{ char}]

\textit{init} == [\textit{APState}' | col' = \emptyset]

\textit{insert} == [\triangle \textit{APState}; \textit{x}? : \text{char} | col' = col \cup \{\textit{x}\}]

\textit{InsS(w)} \triangleq (\text{wait } 0..w; \text{ Insert}) \circ (\text{send}!(\#\textit{col})@t \rightarrow \text{InsS}(w - t))

\textit{BReq1} \triangleq ((\text{in}?\textit{x}@t \rightarrow \text{InsS}(100 - t)) \circ (\text{send}!(\#\textit{col}) \rightarrow \text{Skip}); \textit{BReq1})

\textit{BReq2} \triangleq \textit{out} \rightarrow \text{enable} \rightarrow \text{send}?\textit{x} \rightarrow \textit{BReq2}

\textit{BReq3} \triangleq \text{send}?\textit{x} \rightarrow \text{disable} \rightarrow \textit{BReq3}

\bullet \text{ wait } 0..3; \text{ Init}; (\textit{BReq1}[\{\text{col}\} | \{\} \text{ send } | \{\}](\textit{BReq2}[][\{\} \text{ send } | \{\}])\textit{BReq3}))

\textup{end}

process \textit{TReqs} \triangleq \textup{begin}

\textit{TReq1} \triangleq ((\text{in}?\textit{x} \rightarrow \text{Skip}) \triangleright 5 || \text{ wait } 100); \textit{TReq1}

\textit{TReq2} \triangleq \textit{out} \rightarrow \text{wait } 0..7; \text{ enable} \rightarrow (\text{disable} \rightarrow \text{Skip}) \triangleright 15; \textit{TReq2}

\textup{end}

system \textit{AProtocol} \triangleq \textit{Breqs}[][\{\text{in, out, enable, disable}\} \textit{TReqs}]
A new *Circus Time* theory

- **Alphabet** is a set of variable names for observation.
  - ok, ok', wait, wait': *boolean*
  - tr, tr': $\text{seq}_1(\text{seq Event})$, e.g., $tr' = \langle \langle a \rangle, \langle b, c \rangle, \langle d \rangle, \langle e, f \rangle, \ldots \rangle$
  - ref, ref': $\text{seq}_1(\mathcal{P} \text{Event})$, e.g., $ref' = \langle r_1, r_2, r_3, r_4, \ldots \rangle$
  - state, state': $\mathcal{N} \rightarrow \text{value}$

- **Healthiness conditions** identify properties that characterise the theory.

- **Signature** gives a set of operators and atomic components of the programming theory.

- This theory only focuses on the CSP constructs since the others are the same as those in *Circus*. 
Healthiness conditions in *Circus Time*

- $R_{1ct}(X) \triangleq X \land RT$
- $R_{2ct}(X(tr, tr')) \triangleq X(\langle\langle\rangle\rangle, \text{diff}(tr', tr))$
- $R_{3ct}(X) \triangleq \Pi_{ct} \triangleleft \text{wait} \triangleright X$
- $R_{ct} = R_{1ct} \circ R_{2ct} \circ R_{3ct}$
- $CSP_{1ct}(X) \triangleq X \lor (\neg ok \land RT)$
- $CSP_{2ct}(X) \triangleq X ; J$
- $CSP_{3ct}(X) \triangleq \text{Skip} ; X$
- $CSP_{4ct}(X) \triangleq X ; \text{Skip}$
- $CSP_{5ct}(X) \triangleq X \parallel \text{Skip}$

$RT \triangleq tr \preceq tr' \land \text{front}(\text{ref}) \leq \text{ref}' \land \#\text{diff}(tr', tr) = \#(\text{ref}' - \text{front}(\text{ref}))$

$\Pi \triangleq ok' = ok \land \text{wait}' = \text{wait} \land tr' = tr \land \text{ref}' = \text{ref} \land \text{state}' = \text{state}$

$\Pi_{ct} \triangleq (\neg ok \land RT) \lor (ok' \land \Pi)$

$J \triangleq (ok \Rightarrow ok') \land \Pi \neg ok$

Healthiness conditions in the original and new theories are isomorphic except for $R_{2ct}$
CSP constructs in *Circus Time*

\[
\text{Skip} \mid \text{Stop} \mid \text{Chaos} \mid N := e \mid c.e \rightarrow P \mid P \mid Q \mid P \triangleright b \triangleright Q \mid b&P \mid P \Box Q \\
\mid P 
\triangleright Q \mid P \mid [s_1 \mid \{ CS \} \mid s_2] \mid Q \mid P \setminus CS \mid \text{Wait } d \mid \mu N \bullet P \mid P \triangleright \{d\} Q \\
\mid \text{Miracle} \mid \text{Wait } d_1..d_2 \mid P \triangleright d \mid d \leftarrow P \mid c.e@t \rightarrow P \mid P \triangleright Q \mid P \triangleright_c Q \mid P \triangleright_d Q
\]

- **Miracle**
- **P \triangleright d:** \(P\) **must** terminate within \(d\)
- **d \leftarrow P:** observable events in \(P\) **must** happen within \(d\)
- **c.e@t \rightarrow P:** \(t\) records the relative time of the occurrence of \(c.e\)
- **P \triangleright_c Q:** event-driven timeout
- **P \triangleright_d Q:** time-driven timeout

Prefix, external choice and parallelism are changed for capturing behaviours of actions more precisely.
The relation between the new and original Circus Times.

**Predicates**

*New Circus Time*

*Original Circus Time*

Predicates constructed by *original Circus Time* signature

Predicates constructed by *new Circus Time* signature
Reactive designs

- The theory of relations
  - a predicate with an alphabet for initial and final observations

- The theory of designs
  - $P \vdash Q \triangleq ok \land P \Rightarrow ok' \land Q$
  - $H1(P) = ok \Rightarrow P$, $H2(P) = [P[false/ok'] \Rightarrow P[true/ok']]$
  - $true; P = true$, not $true; P = P$

- The theory of reactive processes
  - $ok$ and $ok'$, $wait$ and $wait'$, $tr$ and $tr'$, $ref$ and $ref'$
  - $R1(P) = P \land tr \leq tr'$, $R2(P(tr, tr')) = P(\langle \rangle, tr' - tr)$,
    $R3(P) = \Pi_R \triangleleft wait \triangleright P$

- The theory of CSP
  - $CSP1(P) = P \lor (\neg ok \land tr \leq tr')$, $CSP2(P) = P; J$

- Reactive designs for CSP processes
  - $P = R_{ct}(\neg P^f_f \vdash P^t_f)$ where $P^a_b \triangleq P[a, b/ok', wait]$
Reactive design for sequential composition

Original

\[ P ; Q \equiv \exists \text{obs}_0 \cdot P[\text{obs}_0/\text{obs}'] \land Q[\text{obs}_0/\text{obs}] \]

Reactive design

\[ P ; Q = R_{ct} \left( \neg (R_{1_{ct}}(P^f_f) ; R_{1_{ct}}(true)) \land \neg (R_{1_{ct}}(P^t_f) ; R_{1_{ct}}(\neg \text{wait} \land R_{2_{ct}}(Q^f_f))) \right) \]

\[ \vdash R_{1_{ct}}(P^t_f) ; R_{1_{ct}}(\exists \text{wait} \triangleright R_{2_{ct}}(Q^i_f)) \]

For example,

\[ (c.e \to \text{Skip}); \text{Miracle} = R_{ct}(true \vdash \text{wait}' \land \triangleright/\text{tr}' = \triangleright/\text{tr} \land \text{possible}(\text{tr}, \text{tr}', c)) \]
Non prefix-closed traces

- Original prefix

\[
\begin{align*}
  \text{wait}\_\text{com}(c) & \lor \text{term}\_\text{com}(c) \lor (\text{wait}\_\text{com}(c); \text{term}\_\text{com}(c)) \\
  \text{wait}\_\text{com}(c) & \equiv \text{wait}' \land \text{possible}(\text{ref}, \text{ref}', c) \land \sim/\text{tr}' = \sim/\text{tr} \\
  \text{term}\_\text{com}(c) & \equiv \neg \text{wait}' \land \text{diff}(\text{tr}', \text{tr}) = \langle\langle c\rangle\rangle
\end{align*}
\]

which, at termination, allows refusals and local state to be arbitrary, and requires prefix-closed traces. E.g., \(\langle\langle\rangle, \langle c\rangle\rangle \in T(P) \Rightarrow \langle\langle\rangle, \langle\rangle\rangle \in T(P)\). But \(\text{Wait } 1; (c \rightarrow \text{Skip} \square \text{Miracle})\) violates this assumption.

- New prefix

\[
\begin{align*}
  \text{wait}\_\text{com}(c) & \lor \text{term}\_\text{now}\_\text{com}(c) \lor (\text{wait}\_\text{com}(c); \text{term}\_\text{next}\_\text{com}(c)) \\
  \text{wait}\_\text{com}(c) & \equiv \text{wait}' \land \text{possible}(\text{ref}, \text{ref}', c) \land \sim/\text{tr}' = \sim/\text{tr} \land \text{state}'=\text{state} \\
  \text{term}\_\text{now}\_\text{com}(c) & \equiv \left( \neg \text{wait}' \land \text{diff}(\text{tr}', \text{tr}) = \langle\langle c\rangle\rangle \land \text{front}(\text{ref}') = \text{front}(\text{ref}) \land \text{state}'=\text{state} \right) \\
  \text{term}\_\text{next}\_\text{com}(c) & \equiv \neg \text{wait} \land \text{tr}' - \text{tr} = \langle\langle c\rangle\rangle \land \text{front}(\text{ref}') = \text{ref} \land \text{state}'=\text{state}
\end{align*}
\]
Divergences are important

- External choice is the most important operator in Circus Time that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original Circus Time
  - the precondition, \( \neg P_f \land \neg Q_f \), in \( P \square Q \)
  - counterexamples, e.g., Wait 3; Chaos \( \square \) Wait2
- Divergences are comprehensively considered in the new theory.

\[ P \square Q \equiv R_{ct} \]

\[
\neg (((P_f^t \lor Q_f^t) \land tr' = tr) ; R_{ct}(true)) \land \\
\neg ((P_f^t \land ((Q_f \land \searrow tr' = \searrow tr \land wait')); tr' - tr = \langle \langle \rangle \rangle) ; R_{ct}(true)) \land \\
\neg ((Q_f^t \land ((P_f \land \searrow tr' = \searrow tr \land wait')); tr' - tr = \langle \langle \rangle \rangle) ; R_{ct}(true)) \land \\
\neg (P_f^t \land ((Q_f \land \searrow tr' = \searrow tr \land wait')); head(tr' - tr) \neq \langle \rangle) \land \\
\neg (Q_f^t \land ((P_f \land \searrow tr' = \searrow tr \land wait')); head(tr' - tr) \neq \langle \rangle) \land \\
\neg ((P_f^t \lor Q_f^t) \land head(diff(tr', tr)) \neq \langle \rangle) \\
\neg (P_f^t \land Q_f^t \land wait' \land \searrow tr' = \searrow tr) \lor (Diff(P_f^t, Q_f^t) \land (P_f^t \lor Q_f^t))
\]

- This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in Circus Time that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original Circus Time
  - the precondition, \( \neg P^f \wedge \neg Q^f \), in \( P \square Q \)
  - counterexamples, e.g., Wait 3; Chaos \( \square \) Wait2
- Divergences are comprehensively considered in the new theory.

\[
P \square Q \cong R_{ct} \begin{cases} 
\neg \left( ((P^f \lor Q^f) \wedge tr' = tr) ; R_{ct}(\text{true}) \right) \land \\
\neg \left( (P^f \land ((Q^f \land \neg tr' = \neg tr \wedge wait')); tr' - tr = \langle \langle \rangle \rangle) ; R_{ct}(\text{true}) \right) \land \\
\neg \left( (Q^f \land ((P^f \land \neg tr' = \neg tr \wedge wait')); tr' - tr = \langle \langle \rangle \rangle) ; R_{ct}(\text{true}) \right) \land \\
\neg \left( (P^f \land ((Q^f \land \neg tr' = \neg tr \wedge wait')); head(tr' - tr) \neq \langle \rangle) \right) \land \\
\neg \left( (Q^f \land ((P^f \land \neg tr' = \neg tr \wedge wait')); head(tr' - tr) \neq \langle \rangle) \right) \land \\
\neg ((P^t \lor Q^t) \land head(diff(tr', tr)) \neq \langle \rangle) \\
\vdash (P^t \land Q^t \land wait' \land \neg tr' = \neg tr) \lor (Diff(P^t, Q^t) \land (P^t \lor Q^t))
\end{cases}
\]

- This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in *Circus Time* that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original *Circus Time*
  - the precondition, $\neg P_f^t \land \neg Q_f^t$, in $P \Box Q$
  - counterexamples, e.g., *Wait 3; Chaos* $\Box$ *Wait2*
- Divergences are comprehensively considered in the new theory.

\[
\begin{align*}
P \Box Q & \cong R_{ct} \\

\neg (((P_f^t \lor Q_f^t) \land tr' = tr) ; R_{ct}(\text{true})) \land \\
\neg ((P_f^t \land ((Q_f^t \land \neg \text{tr}' = \neg \text{tr} \land \text{wait}'); tr' - tr = \langle \rangle)); R_{ct}(\text{true})) \land \\
\neg ((Q_f^t \land ((P_f^t \land \neg \text{tr}' = \neg \text{tr} \land \text{wait}'); tr' - tr = \langle \rangle)); R_{ct}(\text{true})) \land \\
\neg (P_f^t \land ((Q_f^t \land \neg \text{tr}' = \neg \text{tr} \land \text{wait}'); head(tr' - tr) \neq \langle \rangle)) \land \\
\neg (Q_f^t \land ((P_f^t \land \neg \text{tr}' = \neg \text{tr} \land \text{wait}'); head(tr' - tr) \neq \langle \rangle)) \land \\
\neg ((P_f^t \lor Q_f^t) \land head(diff(\text{tr}', \text{tr})) \neq \langle \rangle) \\
\neg (P_f^t \land Q_f^t \land \text{wait}' \land \neg \text{tr}' = \neg \text{tr}) \lor (Diff(P_f^t, Q_f^t) \land (P_f^t \lor Q_f^t))
\end{align*}
\]

- This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in Circus Time that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original Circus Time
  - the precondition, $\neg P^f \land \neg Q^f$, in $P \square Q$
  - counterexamples, e.g., Wait 3; Chaos $\square$ Wait2
- Divergences are comprehensively considered in the new theory.

$$P \square Q \cong R_{ct}$$

This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in \textit{Circus Time} that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original \textit{Circus Time}
  - the precondition, \( \neg P_f \land \neg Q_f \), in \( P \Box Q \)
  - counterexamples, e.g., \( \text{Wait} ~ 3; \text{Chaos} \Box \text{Wait2} \)
- Divergences are comprehensively considered in the new theory.

\[
P \Box Q \cong R_{ct}
\]

\[
\begin{align*}
\neg (((P_f^t \lor Q_f^t) \land tr' = tr) ; R_{ct}(\text{true})) \land \\
\neg ((P_f^f \land ((Q_f^f \land \lnot tr' = \lnot tr \land \text{wait}'); tr' - tr = \langle\rangle)); R_{ct}(\text{true})) \land \\
\neg ((Q_f^f \land ((P_f^f \land \lnot tr' = \lnot tr \land \text{wait}'); tr' - tr = \langle\rangle)); R_{ct}(\text{true})) \land \\
\neg (P_f^f \land ((Q_f^f \land \lnot tr' = \lnot tr \land \text{wait}'); \text{head}(tr' - tr) \neq \langle\rangle)) \land \\
\neg (Q_f^f \land ((P_f^f \land \lnot tr' = \lnot tr \land \text{wait}'); \text{head}(tr' - tr) \neq \langle\rangle)) \land \\
\neg ((P_f^t \lor Q_f^t) \land \text{head}(\text{diff}(tr', tr)) \neq \langle\rangle) \\
\neg (P_f^t \land Q_f^t \land \text{wait}'; \land \lnot tr' = \lnot tr) \lor (\text{Diff}(P_f^t, Q_f^t) \land (P_f^t \lor Q_f^t))
\end{align*}
\]

- This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in *Circus Time* that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original *Circus Time*
  - the precondition, $\neg P^f_f \land \neg Q^f_f$, in $P \Box Q$
  - counterexamples, e.g., *Wait 3; Chaos* $\Box$ *Wait2*

Divergences are comprehensively considered in the new theory.

This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in *Circus Time* that can define other operators such as timeout and deadline.

- Very 'loose' definition for divergences in original *Circus Time*
  - the precondition, $\neg P_f^t \land \neg Q_f^t$, in $P \square Q$
  - counterexamples, e.g., *Wait 3; Chaos \square Wait2*

- Divergences are comprehensively considered in the new theory.

This reactive design looks complex but intuitive.
Divergences are important

- External choice is the most important operator in *Circus Time* that can define other operators such as timeout and deadline.
- Very 'loose' definition for divergences in original *Circus Time*;
  - the precondition, $\neg P_f \land \neg Q_f$, in $P \Box Q$
  - counterexamples, e.g., *Wait* 3; *Chaos* $\Box$ *Wait*2
- Divergences are comprehensively considered in the new theory.

\[
P \Box Q \cong R_{ct} \left\{ \begin{array}{l} \neg ((P_f^t \lor Q_f^t) \land tr' = tr); R_{ct}(\text{true}) \land \\
\neg ((P_f^t \land (Q_f^t \land \neg tr' = \neg tr \land wait')); (tr' - tr = \langle\rangle); R_{ct}(\text{true}) \land \\
\neg ((Q_f^t \land (P_f^t \land \neg tr' = \neg tr \land wait')); (tr' - tr = \langle\rangle); R_{ct}(\text{true}) \land \\
\neg (P_f^t \land ((Q_f^t \land \neg tr' = \neg tr \land wait')); \text{head}(tr' - tr) \neq \langle\rangle) \land \\
\neg (Q_f^t \land ((P_f^t \land \neg tr' = \neg tr \land wait')); \text{head}(tr' - tr) \neq \langle\rangle) \land \\
\neg ((P_f^t \lor Q_f^t) \land \text{head}(\text{diff}(tr', tr)) \neq \langle\rangle) \\
\vdash (P_f^t \land Q_f^t \land wait' \land \neg tr' = \neg tr) \lor (\text{Diff}(P_f^t, Q_f^t) \land (P_f^t \lor Q_f^t)) \end{array} \right. \]

- This reactive design looks complex but intuitive.
Miracle

- \[ \text{Miracle} \triangleq R_{ct}(true \vdash false) = R_{1ct}(ok) \lor (ok' \land I \land wait) \]
- Miracle is the top element in the complete lattice by the refinement relation.
- Miracle is an unstarted action that must not be seen during any execution.
- The combination between Miracle and other operators violates some assumptions of the standard CSP.
  - \( a \rightarrow \text{Miracle} \)
  - \( a \rightarrow \text{Skip} \land \square \text{Miracle} \)
  - \( \text{Wait } d; \text{Miracle} \)

- Miracle is used to define a deadline operator
Deadlines

- Deadlines impose requirements on a system's environment.
- Missing a deadline leads to a timestep (infeasibility) in Circus Time.
- Two kinds of deadlines: an action must terminate (▶), or must execute (◀) external events within a deadline.

\[
P ▶ d ≡ R_{ct} \left( \begin{array}{l}
- ((P^f \land tr' = tr) \land R_{1ct}(true)) \\
- ((P^f \land \#tr' - \#tr \leq d) \land R_{1ct}(true)) \\
\quad \vdash (P^t \land \#tr' - \#tr \leq d)
\end{array} \right)
\]

\[
d ◁ P = R_{ct} \left( \begin{array}{l}
- ((P^f \land tr' = tr) \land R_{1ct}(true)) \\
- ((P^f \land \#tr' - \#tr \leq d \land \neg (\neg tr' = tr) \land head(tr' - tr) \neq \langle \rangle)) \\
\quad \vdash (P^t \land head(diff(tr', tr)) \neq \langle \rangle)
\end{array} \right)
\]

\[
(P^t \land (\#tr' - \#tr < d \land \neg (\neg tr' = tr) \lor (P^t \land (((\#tr' - \#tr < d \land \neg tr' = tr \land wait' = tr' \land wait')) \lor (P^t \land (((\#tr' - \#tr < d \land \neg tr' = tr \land wait')) \lor (P^t \land (((\#tr' - \#tr < d \land \neg tr' = tr \land wait') \land term_next) \lor term_now))))
\]

\[
\quad = P \square (Wait d; Miracle)
\]

- E.g., 3 ◁ (a → b → Skip) ▶ 5
Interrupts

- Event-driven: Seminar △ (Wait 1; close → Skip)
- Catastrophe: (Seminar △ (close → Skip)) || {close} || (Wait 1; close → Skip)

\[(P \triangle_c Q) = \]

\[
\begin{align*}
& \neg ((P^f \land possible(ref, ref', c)); R_{ct}(true)) \land \\
& \neg ((P^t \land possible(ref, ref', c)); (wait \land term\_now\_com(c)); (\neg wait \land Q^f_t)) \\
& \vdash (P^t \land \#tr' = \#tr \land \neg wait') \lor \\
& (P^t \land \#tr' \neq \#tr \land possible(ref, front(ref'), c) \land \neg wait') \lor \\
& ((P^t \land possible(ref, ref', c)); (wait \land wait\_com(c))) \lor \\
& ((P^t \land possible(ref, ref', c)); (wait \land term\_now\_com(c)); (\neg wait \land Q^t_t))
\end{align*}
\]

- Generic interrupt is constructed by parallel-by-merge, but cannot resolve the updating of local state.

- Time-driven: Seminar △₁ Skip

\[
P \triangle_d Q \equiv \begin{align*}
& \neg ((P^f \land \#tr' - \#tr \leq d); R_{ct}(true)) \land \\
& \neg ((P^t \land \#tr' - \#tr = d); (\neg wait \land \neg wait')) \land \\
& \vdash (P^t \land \#tr' - \#tr \leq d) \lor \\
& ((P^t \land \#tr' - \#tr = d); (\neg wait \land \neg wait'))
\end{align*}
\]

Contracts in a State-rich Timed Process Algebra
Conclusion and future work

- We have developed a new *Circus Time* theory that, compared with the original theory, provides a simpler mathematical model, more time operators and more constrained operators.
- Each action is expressed as a reactive design that exposes the pre-postcondition semantics and simplifies proofs.
- Reactive designs give more concise, readable and uniform UTP semantics, and also support contract-based reasoning.

Future work:
- Mechanisation of the semantics of *Circus Time* in a theorem prover.