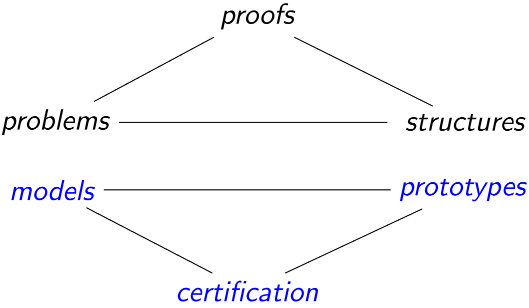
Bringing class diagrams to life

Luis S. Barbosa & Sun Meng

DI-CCTC, Minho University, Braga & CWI, Amsterdam

UML & FM Workshop 2009
Rio de Janeiro
8 December, 2009

Formal Methods



- Modelling: choose the right abstractions for a problem domain
- Calculation: express such abstractions in a mathematical framework rich enough to enable rigorous reasoning
- Prototyping: execute models to simulate systems' behavior and gather empirical evidence about their properties

UML & FM

"a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system".
OMG

- The number and diversity of diagrams expressing a UML model makes it difficult to base its semantics on a single framework.
- Some of the formalisations proposed in the literature are essentially descriptive and difficult to use in proofs.

A research agenda

The quest for a precise notion of behaviour and a calculational approach to behavioural equivalence and refinement suggested the adoption of a

$\boxed{\text{coalgebraic framework}}$

- standard notion of systems' behaviour in terms of the bisimilarity relation induced by each signature functor, upon which properties of UML models can be formulated and checked.
- built on top of a characterisation by an universal property which entails the foundations for derived calculi
- uniform setting for reasoning about the diversity of UML models and their inter-relations

Plan

- Motivation
- Why coalgebras?
- Class as coalgebras
- Composition
- Constraints
- Associations
- Future work

Coalgebras

a lens:	◯⌢◯
a tool box:	⋒ ◫
an observation structure:	◯⌢◯ universe $\xleftarrow{\alpha}$ universe
an assembly process:	artifact $\xleftarrow{d}$ ⋒ ◫ artifact

$$\alpha : \mathsf{T}U \longleftarrow U$$

- coalgebras describe transition systems
- and abstract behaviour types as (final) coalgebras
- emphasis is on observation

Coalgebras

a lens:	
a tool box:	
an observation structure:	universe $\xleftarrow{\alpha}$ universe
an assembly process:	artifact $\xleftarrow{d}$ artifact

$$\alpha : \mathsf{T}U \longleftarrow U$$

- coalgebras describe transition systems
- and abstract behaviour types as (final) coalgebras
- emphasis is on observation

Coalgebraic modelling

... the mathematics of state-based systems (Jacobs,07)

e.g., streams ($\mathsf{T}X = A \times X$) and (different types of) automata

in our own research in UML semantics

- software components (Barbosa,01): $\mathsf{T}X = \mathsf{B}(X \times O)^I$
- objects (Cruz, Barbosa, Oliveira, 05): $\mathsf{T}X = A \times \mathsf{B}(X)^I$
- statecharts (Meng, Niaxiao, Barbosa,04): $\mathsf{T}X = \mathsf{B}(X \times \mathcal{P}E)^E$
- UML sequence diagrams (Meng & Barbosa,08): $\mathsf{T}X = X^\Sigma$
- UML class diagrams (Barbosa & Meng,08): ...

Coalgebraic modelling

... the mathematics of state-based systems (Jacobs,07)

e.g., streams ($\mathsf{T}X = A \times X$) and (different types of) automata

in our own research in UML semantics

- software components (Barbosa,01): $\mathsf{T}X = \mathsf{B}(X \times O)^I$
- objects (Cruz, Barbosa, Oliveira, 05): $\mathsf{T}X = A \times \mathsf{B}(X)^I$
- statecharts (Meng, Niaxiao, Barbosa,04): $\mathsf{T}X = \mathsf{B}(X \times \mathcal{P}E)^E$
- UML sequence diagrams (Meng & Barbosa,08): $\mathsf{T}X = X^\Sigma$
- UML class diagrams (Barbosa & Meng,08): ...

Coalgebras as T-shaped transition structures

$$\begin{tikzcd} \mathsf{T}U & U \arrow[l, "\beta"'] \arrow[d, "h"] \\ \mathsf{T}V \arrow[from=u, "\mathsf{T}h"'] & V \arrow[l, "\alpha"] \end{tikzcd} \qquad \textit{i.e.} \qquad \mathsf{T}h \cdot \beta = \alpha \cdot h$$

$$\begin{tikzcd} U & U \arrow[l, "\beta\longleftarrow"'] \arrow[d, "h"] \\ V \arrow[from=u, "h"'] & V \arrow[l, "\alpha\longleftarrow"] \end{tikzcd} \qquad \textit{i.e.} \qquad h \cdot {}_{\beta}\!\longleftarrow = {}_{\alpha}\!\longleftarrow \cdot h$$

Coalgebras as T-shaped transition structures

$$\xleftarrow{\alpha} \quad \stackrel{def}{=} \quad \in_{\mathsf{F}} \cdot \alpha$$

where $\in_{\mathsf{T}}$ is functorial membership, a natural transformation given by

$$
\begin{array}{rcl}
\in_{\mathsf{Id}} & = & id \\
\in_{\mathsf{K}} & = & \bot \\
\in_{T_1 \times T_2} & = & (\in_{T_1} \cdot \pi_1) \cup (\in_{T_2} \cdot \pi_2) \\
\in_{T_1 + T_2} & = & [\in_{T_1}, \in_{T_2}] \\
\in_{T_1 \cdot T_2} & = & \in_{T_2} \cdot \in_{T_1} \\
\in_{T^K} & = & \bigcup_{k \in K} \in_T \cdot \beta_k \ (\text{where } \beta_k f = f\ k) \\
\in_{\mathcal{P}} & = & \in
\end{array}
$$

Morphisms preserve and reflect transitions

$$\mathsf{T}h \cdot d \;=\; c \cdot h$$

which entails

$$h \cdot \;{}_{d}\!\longleftarrow \;=\; {}_{c}\!\longleftarrow \cdot h$$

i.e., the conjunction of inclusions

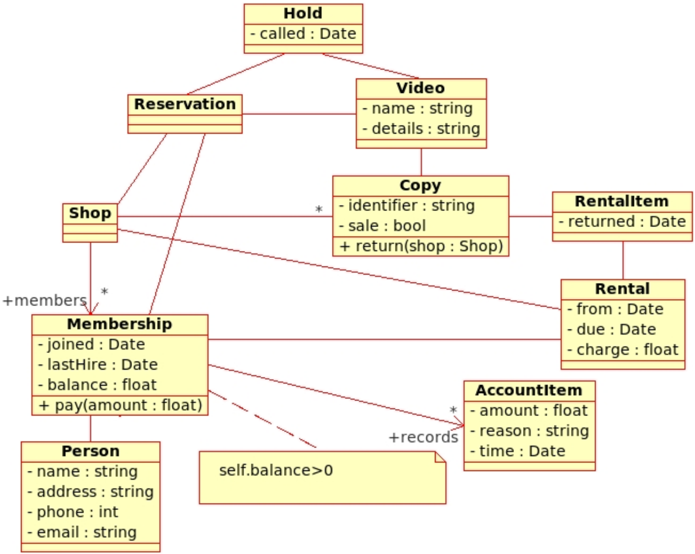
$$h \cdot \;{}_{d}\!\longleftarrow \;\subseteq\; {}_{c}\!\longleftarrow \cdot h$$
$${}_{c}\!\longleftarrow \cdot h \;\subseteq\; h \cdot \;{}_{d}\!\longleftarrow$$

or, going pointwise,

$$v' \;{}_{d}\!\longleftarrow v \;\Rightarrow\; h\,v' \;{}_{c}\!\longleftarrow h\,v$$
$$u' \;{}_{c}\!\longleftarrow h\,v \;\Rightarrow\; \exists_{v' \in V}.\; v' \;{}_{c}\!\longleftarrow v \;\wedge\; u' = h\,v'$$

$\boxed{\text{morphisms entail (T-shaped) bisimilarity}}$

Class diagrams



Class diagrams

A class diagram captures the static structure of a system, as a set of classes and relationships between them.

Example

$$\begin{aligned}
\mathsf{joined} &: \mathit{Date} \longleftarrow U\\
\mathsf{lastHire} &: \mathit{Date} \longleftarrow U\\
\mathsf{balance} &: \mathbb{R} \longleftarrow U\\
\overline{\mathsf{pay}} &: U \longleftarrow U^{\mathbb{R}}
\end{aligned}$$

$$[\![\mathit{Membership}]\!] = \langle \mathsf{joined}, \mathsf{lastHire}, \mathsf{balance}, \overline{\mathsf{pay}} \rangle$$

which is a coalgebra for functor

$$\mathsf{T}\,X = \mathit{Date} \times \mathit{Date} \times \mathbb{R} \times X^{\mathbb{R}}$$

Classes as coalgebras

state space	U
initial condition	$\epsilon : \mathbf{2} \longleftarrow U$
methods	$\overline{\mathrm{md}} : (O \times U)^I \longleftarrow U$
attributes	$\mathrm{at} : A \longleftarrow U$

i.e., class is a coalgebra for functor

$$\mathsf{T}\, X \;=\; A \times (O \times X)^I$$

whose initial states verify ϵ.

Classes as coalgebras

More generally, as methods are typically partial functions or even arbitrary relations:

$$\mathsf{T}\,X \;= A \times ((O \times X) + \mathbf{1})^I$$
$$\mathsf{T}\,X \;= A \times \mathcal{P}(O \times X)^I$$

respectively. Both cases, are subsumed by

$$\langle \mathrm{at}, \overline{\mathrm{md}} \rangle : A \times \mathsf{B}(O \times U)^I \longleftarrow U$$

where functor T is parametric in a strong monad B

leading to a calculus for class composition

Classes as coalgebras

More generally, as methods are typically partial functions or even arbitrary relations:

$$\mathsf{T}\,X \;= A \times ((O \times X) + \mathbf{1})^I$$
$$\mathsf{T}\,X \;= A \times \mathcal{P}(O \times X)^I$$

respectively. Both cases, are subsumed by

$$\langle \mathsf{at}, \overline{\mathsf{md}} \rangle : A \times \mathsf{B}(O \times U)^I \longleftarrow U$$

where functor T is parametric in a strong monad B

$\boxed{\text{leading to a calculus for class composition}}$

Prototyping classes

- mechanism for registering and selecting class instances
- step-by-step interaction with each specific instance through activation of the corresponding $\langle \mathrm{at}, \overline{\mathrm{md}} \rangle$ operation
- class composition shapes a fragment of a Class Diagram as a coalgebra itself

An calculus of classes

$$\langle \mathrm{at}_p, \overline{\mathrm{md}}_p \rangle : A \times (\mathsf{B}\,(O \times U_p))^I \longleftarrow U_p$$

- Three tensor products:
  $\boxtimes$ (synchronous product), $\boxplus$ (choice) and $\circledast$ (concurrent)
- Attributes are always observable (herefore are composed in a multiplicative context)
- Initial conditions are joined by logical conjunction
- Rich calculus: properties expressed as T-bisimulation equations

Parallel

$$p \boxtimes q \;=\; \langle \gamma_{p\boxtimes q}, \langle \mathsf{at}_{p\boxtimes q}, \overline{\mathsf{md}}_{p\boxtimes q} \rangle\rangle$$

where

$$\gamma_{p\boxtimes q} \;=\; U \times V \xrightarrow{\gamma_p \times \gamma_q} \mathbf{2} \times \mathbf{2} \xrightarrow{\wedge} \mathbf{2}$$

$$\mathsf{at}_{p\boxtimes q} \;=\; U \times V \xrightarrow{\mathsf{at}_p \times \mathsf{at}_q} A \times A'$$

$$\begin{aligned}
\mathsf{md}_{p\boxtimes q} \;=\; & U \times V \times (I \times I') \xrightarrow{\mathsf{m}} (U \times I) \times (V \times I') \\
& \xrightarrow{\mathsf{md}_p \times \mathsf{md}_q} \mathsf{B}(O \times U) \times \mathsf{B}(O' \times V) \\
& \xrightarrow{\delta} \mathsf{B}((O \times U) \times (O' \times V)) \\
& \xrightarrow{\mathsf{Bm}} \mathsf{B}((O \times O') \times (U \times V))
\end{aligned}$$

Choice

$$p \boxplus q = \langle \gamma_{p\boxplus q}, \langle \mathsf{at}_{p\boxplus q}, \overline{\mathsf{md}}_{p\boxplus q} \rangle \rangle$$

where

$$
\begin{aligned}
\mathsf{md}_{p\boxplus q} = U \times V \times (I + I') &\xrightarrow{\triangle \times \mathrm{id}} (U \times V)^2 \times (I + I') \\
&\xrightarrow{\cong} (U \times I) \times V + (V \times I') \times U \\
&\xrightarrow{f} \mathsf{B}(O \times U) \times V + \mathsf{B}(O' \times V) \times U \\
&\xrightarrow{\tau_r \times \tau_r} \mathsf{B}((O \times U) \times V) + \mathsf{B}((O' \times V) \times U) \\
&\xrightarrow{\cong} \mathsf{B}(O \times (U \times V)) + \mathsf{B}(O' \times (U \times V)) \\
&\xrightarrow{g} \mathsf{B}((O + O') \times U \times V) + \mathsf{B}((O + O') \times U \times V) \\
&\xrightarrow{\nabla} \mathsf{B}((O + O') \times U \times V)
\end{aligned}
$$

Choice

where

$$f \stackrel{abv}{=} \mathsf{md}_p \times \mathsf{id} + \mathsf{md}_q \times \mathsf{id}$$
$$g \stackrel{abv}{=} \mathsf{B}(\iota_1 \times \mathsf{id}) + \mathsf{B}(\iota_2 \times \mathsf{id})$$

$$\triangle = \langle \mathsf{id}, \mathsf{id} \rangle$$
$$\triangledown = [\mathsf{id}, \mathsf{id}]$$

Wrapping

A mechanism for input/output (for class adaptation)

$$\mathrm{md}_{p[f,g]} = U_p \times I' \xrightarrow{\mathrm{id}\times f} U_p \times I \xrightarrow{\mathrm{md}_p} \mathsf{B}(U_p \times O) \xrightarrow{\mathsf{B}(\mathrm{id}\times g)} \mathsf{B}(U_p \times O')$$

where $f : I \longleftarrow I'$ and $g : O' \longleftarrow O$.

Properties

$$(p[f,g])[f',g'] \;\sim\; p[f \cdot f', g' \cdot g]$$

because

$$\begin{aligned}
& \mathsf{md}_{(p[f,g])[f',g']} \\
\sim \quad & \quad \{ \text{ wrapping definition } \} \\
& \mathsf{B}(\mathsf{id} \times g') \cdot \mathsf{md}_{p[f,g]} \cdot (\mathsf{id} \times f') \\
\sim \quad & \quad \{ \text{ wrapping definition } \} \\
& \mathsf{B}(\mathsf{id} \times g') \cdot \mathsf{B}(\mathsf{id} \times g') \cdot \mathsf{md}_p \cdot (\mathsf{id} \times f) \cdot (\mathsf{id} \times f') \\
\sim \quad & \quad \{ \times \text{ is a functor } \} \\
& \mathsf{B}(\mathsf{id} \times g' \cdot g) \cdot \mathsf{md}_p \cdot (\mathsf{id} \times f \cdot f') \\
\sim \quad & \quad \{ \text{ wrapping definition } \} \\
& \mathsf{md}_{p[f \cdot f', g' \cdot g]}
\end{aligned}$$

Properties

Moreover,

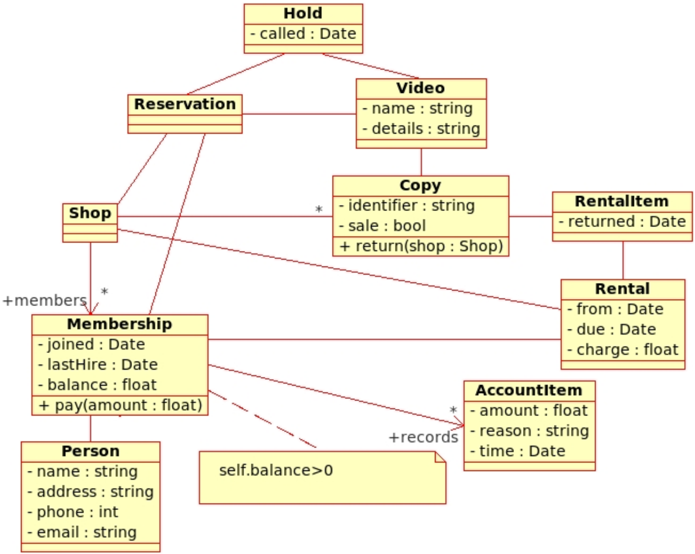
- $\boxtimes$, $\boxplus$ and $\circledast$ are associative as well as commutative (if B is a commutative monad)
- All properties are stated up to bisimilarity:

proof technique:
bisimilarity is witnessed by any morphism linking both sides

Properties

$$
\begin{array}{ll}
& \mathsf{md}_{q\boxtimes p[\mathsf{s},\mathsf{s}]} \cdot (\mathsf{s}\times\mathsf{id}) \\
= & \quad \{\ \boxtimes \text{ and wrapping definition } \} \\
& \mathsf{B}(\mathsf{id}\times\mathsf{s})\cdot\mathsf{Bm}\cdot\delta_l\cdot(a_q\times a_p)\cdot\mathsf{m}\cdot(\mathsf{id}\times\mathsf{s})\cdot(\mathsf{s}\times\mathsf{id}) \\
= & \quad \{\ \mathsf{s} \text{ natural and } \mathsf{s}\cdot\mathsf{m}=\mathsf{m}\cdot(\mathsf{s}\times\mathsf{s}) \ \} \\
& \mathsf{B}(\mathsf{id}\times\mathsf{s})\cdot\mathsf{Bm}\cdot\delta_l\cdot\mathsf{s}\cdot(a_p\times a_q)\cdot\mathsf{m} \\
= & \quad \{\ \delta_l,\ \delta_r \text{ interchangeable } \} \\
& \mathsf{B}(\mathsf{id}\times\mathsf{s})\cdot\mathsf{Bm}\cdot\mathsf{Bs}\cdot\delta_r\cdot(a_p\times a_q)\cdot\mathsf{m} \\
= & \quad \{\ \text{routine: } \mathsf{m}\cdot\mathsf{s}=(\mathsf{s}\times\mathsf{s})\cdot\mathsf{m} \ \} \\
& \mathsf{B}(\mathsf{id}\times\mathsf{s})\cdot\mathsf{B}(\mathsf{s}\times\mathsf{s})\cdot\mathsf{Bm}\cdot\delta_r\cdot(a_p\times a_q)\cdot\mathsf{m} \\
= & \quad \{\ \mathsf{B} \text{ commutative } \} \\
& \mathsf{B}(\mathsf{id}\times\mathsf{s})\cdot\mathsf{B}(\mathsf{s}\times\mathsf{s})\cdot\mathsf{Bm}\cdot\delta_l\cdot(a_p\times a_q)\cdot\mathsf{m} \\
= & \quad \{\ \mathsf{s}=\mathsf{s}^\circ,\ \boxtimes \text{ and wrapping definition } \} \\
& \mathsf{B}(\mathsf{s}\times\mathsf{id})\cdot\mathsf{md}_{p\boxtimes q}
\end{array}
$$

Constraints as invariants



Constraints as invariants

OCL constraint

$$\text{balance} > 0$$

in CD is supposed to be preserved along the system life.
Formally, it is incorporated in the semantics as an invariant

An invariant [Jac, 06] for a coalgebra $c : X \to \mathrm{T}(X)$ is a predicate $P \subseteq X$ satisfying for all $x \in X$,

$$x \in P \quad \Rightarrow \quad c(x) \in Pred(\mathrm{T})(P).$$

where $Pred(\mathrm{T})(P)$ stands for the *lifting* of predicate P via T

Constraints as invariants

OCL constraint

$$\text{balance} > 0$$

in CD is supposed to be preserved along the system life.
Formally, it is incorporated in the semantics as an invariant

An invariant [Jac, 06] for a coalgebra $c : X \to \mathrm{T}(X)$ is a predicate $P \subseteq X$ satisfying for all $x \in X$,

$$x \in P \quad \Rightarrow \quad c(x) \in \mathit{Pred}(\mathrm{T})(P).$$

where $\mathit{Pred}(\mathrm{T})(P)$ stands for the *lifting* of predicate P via T

Constraints as invariants

Making the definition more amenable to formal calculation we

- represent predicates as coreflexives, i.e., fragementes of id:

$$y\ \Phi_X\ x \quad \equiv \quad y = x \wedge x \in X$$

- identify the lifting $Pred(\mathsf{T})(P)$ of predicate P with its image through relator T

cf *Calculating invariants as coreflexive bisimulations* (BOS, 08)

Constraints as invariants

$$\langle \forall\ x\ ::\ x \in P\ \Rightarrow\ c(x) \in\ \mathit{Pred}(\mathsf{T})(P)\rangle$$

$\equiv$ $\{$ $\forall$-one point rule $\}$

$$\langle \forall\ y, x\ :\ y = x :\ x \in P\ \Rightarrow\ c(y) = c(x) \wedge c(x) \in\ \mathit{Pred}(\mathsf{T})(P)\rangle$$

$\equiv$ $\{$ $\forall$-trading $\}$

$$\langle \forall\ y, x\ ::\ y = x \wedge x \in P\ \Rightarrow\ c(y) = c(x) \wedge c(x) \in\ \mathit{Pred}(\mathsf{T})(P)\rangle$$

$\equiv$ $\{$ predicates as coreflexives $\}$

$$\langle \forall\ y, x\ ::\ y\ \Phi_P\ x\ \Rightarrow\ c(y)\ \Phi_{\mathit{Pred}(\mathsf{T})(P)}\ c(x)\rangle$$

$\equiv$ $\{$ rule $(f\ b)R(g\ a)\ \equiv\ b(f^\circ \cdot R \cdot g)a$ $\}$

$$\langle \forall\ y, x\ ::\ y\ \Phi_P\ x\ \Rightarrow\ y(c^\circ \cdot \Phi_{\mathit{Pred}(\mathsf{T})(P)} \cdot c)x\rangle$$

$\equiv$ $\{$ inclusion $\}$

$$\Phi_P\ \subseteq c^\circ \cdot \Phi_{\mathit{Pred}(\mathsf{T})(P)} \cdot c$$

$\equiv$ $\{$ shuntig rule and relator definition $\}$

$$c \cdot \Phi_P\ \subseteq \mathsf{T}\,\Phi_P \cdot c$$

Constraints as invariants

Therefore

$$
\begin{array}{l}
[\![\text{balance} > 0]\!] \;\; = \\
[\![\mathit{Membership}]\!] \cdot \Phi_{\text{balance}>0} \subseteq \mathsf{T}\,\Phi_{\text{balance}>0} \cdot [\![\mathit{Membership}]\!]
\end{array}
$$

- Constraints stand for proof obligations when transforming Class Diagrams

A calculus of CD constraints

Instantiating the proof obligation to T, yields

$$
\begin{array}{rl}
& \langle \mathsf{at}, \overline{\mathsf{md}} \rangle \cdot \Phi_P \subseteq A \times \mathsf{B}(O \times \Phi_P)^I \cdot \langle \mathsf{at}, \overline{\mathsf{md}} \rangle \\
\equiv & \quad \{ \text{ split fusion and absorption } \} \\
& \langle \mathsf{at} \cdot \Phi_P, \overline{\mathsf{md}} \cdot \Phi_P \rangle \subseteq \langle \mathsf{at}, \mathsf{B}(O \times \Phi_P)^I \cdot \overline{\mathsf{md}} \rangle \\
\equiv & \quad \{ \text{ structural equality } \} \\
& \quad \mathsf{at} \cdot \Phi_P \subseteq \mathsf{at} \\
& \quad \overline{\mathsf{md}} \cdot \Phi_P \subseteq \mathsf{B}(O \times \Phi_P)^I \cdot \overline{\mathsf{md}}
\end{array}
$$

Initial conditions also satisfies constraints, *i.e.*,

$$\forall_{u \in U} \,.\; \gamma\, u \Rightarrow P\, u$$

or, in a point-free format

$$\Phi_\gamma \subseteq \Phi_P$$

A calculus of CD constraints

Constraints are preserved by combinators, $\boxplus$, $\boxtimes$ and $\circledast$, ie,

$$\begin{aligned} \mathsf{at}_{p \boxtimes q} \cdot \Phi_{(P \times P')} &\subseteq \mathsf{at}_{p \boxtimes q} \\ \overline{\mathsf{md}}_{p \boxtimes q} \cdot \Phi_{(P \times P')} &\subseteq \mathsf{B}((O \times O') \times \Phi_{(P \times P')})^{I \times I'} \cdot \overline{\mathsf{md}}_{p \boxtimes q} \end{aligned}$$

- the first inequality holds because $\Phi_{(P \times P')}$ is a coreflexive.
- to prove the second we reason

A calculus of CD constraints

$$
\begin{array}{ll}
 & \mathsf{md}_{p \boxtimes q} \cdot (\Phi_{(P \times P')} \times (\mathsf{id} \times \mathsf{id})) \\
= & \qquad \{ \text{ definition of } \boxtimes \text{ and } \Phi_{(P \times P')} = \Phi_P \times \Phi_{P'} \} \\
 & \mathsf{Bm} \cdot \delta \cdot (\mathsf{md}_p \times \mathsf{md}_q) \cdot \mathsf{m} \cdot ((\Phi_P \times \Phi_{P'}) \times (\mathsf{id} \times \mathsf{id})) \\
= & \qquad \{ \text{ m is a natural transformation and } \times \text{ is a functor} \} \\
 & \mathsf{Bm} \cdot \delta \cdot ((\mathsf{md}_p \cdot (\Phi_P \times \mathsf{id})) \times (\mathsf{md}_q \cdot (\mathsf{id} \times \Phi_{P'}))) \cdot \mathsf{m} \\
\subseteq & \qquad \{ \text{ } p \text{ (resp., } q\text{) preserves } P \text{ (resp., } P'\text{)} \} \\
 & \mathsf{Bm} \cdot \delta \cdot ((\mathsf{B}(O \times \Phi_P) \cdot \mathsf{md}_p) \times (\mathsf{B}(O' \times \Phi_{P'}) \cdot \mathsf{md}_q)) \cdot \mathsf{m} \\
= & \qquad \{ \text{ } \delta \text{ is a natural transformation and } \times \text{ is a functor} \} \\
 & \mathsf{Bm} \cdot \mathsf{B}((O \times \Phi_P) \times (O' \times \Phi_{P'})) \cdot \delta \cdot (\mathsf{md}_p \times \mathsf{md}_q) \cdot \mathsf{m} \\
= & \qquad \{ \text{ m is a natural transformation} \} \\
 & \mathsf{B}((O \times O') \times (\Phi_P \times \Phi_{P'})) \cdot \mathsf{Bm} \cdot \delta \cdot (\mathsf{md}_p \times \mathsf{md}_q) \cdot \mathsf{m} \\
= & \qquad \{ \text{ definition of } \boxtimes \} \\
 & \mathsf{B}((O \times O') \times (\Phi_P \times \Phi_{P'})) \cdot \mathsf{md}_{p \boxtimes q}
\end{array}
$$

Associations

- Types of relationships between sets of instances
- Invariants over a coalgebra representing the whole diagram dynamics (the diagram engine) over Pop × Assocs, where

$$
\begin{aligned}
\mathsf{Pop} &= \mathcal{P}(\mathsf{Ref})^{\mathsf{ClassId}} \\
\mathsf{Assocs} &= \mathcal{P}(\mathsf{Assoc})^{\mathsf{AId}} \\
\mathsf{Assoc} &= \mathsf{ClassId} \times \mathsf{ClassId} \times \mathcal{P}(\mathsf{Ref} \times \mathsf{Ref})
\end{aligned}
$$

Fundamental property

Associations are total with respect to the actual sets of instances of the classes involved.

How can this be expressed?

Let $S = (\rho, \alpha)$ be the state space of the *diagram engine* coalgebra, and for all association identifier a, let $\alpha(a) = (c, d, r)$. The association is total iff

$$\mathrm{id}_{\rho(c)} \subseteq \ker r$$

where

$$\begin{aligned} \ker R &= R^\circ \cdot R \\ \mathrm{img}\, R &= R \cdot R^\circ \end{aligned}$$

(cf, the pointfree calculus of binary relations (BH93))

Specification of associations

- one-to-one: $\ker r \subseteq id_{\rho(c)}$ (injectivity), $\mathrm{img}\, r \subseteq id_{\rho(d)}$ (simplicity), and totality leads to

$$\ker r = id_{\rho(c)} \ \wedge \ \mathrm{img}\, r \subseteq id_{\rho(d)}$$

- many-to-one: $\mathrm{img}\, r \subseteq id_{\rho(d)}$, which combined with totality yields

r is a total function

- one-to-many: $\mathrm{img}\, r^{\circ} \subseteq id_{\rho(c)}$ which is equivalent, by duality, to $\ker r \subseteq id_{\rho(c)}$. Together with totality yields

$$\ker r = id_{\rho(c)}$$

- many-to-many: any relation does the job.

Specification of associations

- at most m in the source class

$$\forall_{p \in \mathrm{dom}\, r} \cdot \quad \#(r \cdot \{p\}) \leq m$$

- at most n in the target class

$$\forall_{q \in \mathrm{rng}\, r} \cdot \quad \#(\{q\} \cdot r) \leq n$$

Note:
p is a coreflexive pair, composition $r \cdot \{p\}$ corresponds to relation $\{(y, \pi_1\, p) |\, (y, \pi_1\, p) \in r\}$.

The diagram engine

- associations are properties of binary relations between class instances
- to prototype a CD entails the need to refer explicitly to the sets of class instances as well as to the actual relations between instances
- … leads to a coalgebra over $\mathrm{Pop} \times \mathrm{Assocs}$ to represent the dynamics of the whole diagram
- properties of associations should be regarded as invariants for such a coalgebra

$\boxed{\text{But what is its shape?}}$

The diagram engine

Basic operations required:

- create new instances:
  $\mathsf{new} : U \times \mathsf{Ref} \longleftarrow U \times \mathsf{ClassId}$
- remove instances:
  $\mathsf{del} : U \longleftarrow U \times \mathsf{Ref}$
- connect a class instance to another in the context of a declared association:
  $\mathsf{connect} : U \longleftarrow U \times \mathsf{AId} \times (\mathsf{ClassId} \times \mathsf{Ref})^2$
- disconnect a class instance from an association:
  $\mathsf{disconnect} : U \longleftarrow U \times \mathsf{Ref} \times \mathsf{AId}$

The diagram engine

... which leads to

$$(U, \delta : (\mathrm{Ref} + \mathbf{1}) \times U)^{IP} \longleftarrow U)$$

where

$$IP = \mathrm{ClassId} + \mathrm{Ref} + (\mathrm{AId} \times \mathrm{ClassId} \times \mathrm{Ref})^2 + (\mathrm{Ref} \times \mathrm{AId})$$

represents the input parameters for the four operations.

- Initial conditions can be specified to characterize δ initial valid states (for example, forcing initially all sets of instances to be empty).

The diagram engine

Haskell prototyping

- available at both class and diagram levels
- requires prototyping srategies to deal with unstable states wrt δ invariants (ie, association properties): for example, after the creation of a new instance and before its addition to the relevant associations;
- in particular, on creating or removing a class instance, this forces δ to be not observabed until the associated operations of connecting or disconnecting terminate,

Future work

Where shall I go from here?, asked Alice.
That depends a great deal on where you would like to get to, said the Cat.

Lewis Caroll

Current & future work

- Current work (at the prototyping level):
  - plan significative case studies to assess empirically the merits of the approach and library
  - introduce coalgebraic refinement as another dimension in the calculus of class diagrams and an option at prototyping level;
- Long term: the quest for effective calculi
  - deriving coalgebraic calculi for different types of UML diagrams (e.g. class diagrams, statecharts and sequence diagrams);
  - combining them and their "canonical" prototypers.