Object Diagrams

Object Diagrams are used to visualize the structure of a system, including the classes, objects, and associations among them. They are good for documenting the static structure of a system.

When an object diagram is partial, it means that it does not include all the objects and associations that are part of the system. Partial object diagrams are good for showing incomplete or evolving parts of a system.

When an object diagram is with respect to another thing, it means that the objects and associations are defined relative to that other thing.

System states and object diagrams are related in that the object diagrams can represent different states of the system. The satisfaction of an OCL expression indicates that the expression holds true for the given object diagram.

A set of OCL constraints is consistent when there is no conflict between the constraints.

Can you think of an object diagram that violates this OCL constraint?
are labelled with $G$. Then the nodes in $\mathcal{O}$-typed attribute/value pairs only.

Let $\sigma$ be a system state. We say attribute $\sigma$ is closed if and only if no attribute has a dangling reference in $\sigma$. Any object alive in $\sigma$, i.e. if $\sigma$ has value comprises an object which is not alive in $\sigma$. We call $\sigma$ complete if and only if the attribute's value comprises an object which is not alive in $\sigma$.

$\sigma$ is then a singleton. In other words: a given (consistent) object diagram tells us what the corresponding finite system state can have which is represented by the object diagram.

$\sigma$ is complete.

Special Notation

Closed Object Diagrams vs. Dangling References

Find the 10 differences! (Both diagrams shall be complete.)
Aftermath

Weslightlydeviatefromthestandard(for reasons):

• In the course, $C_0$, $C_1$ and $C^*$-typed attributes only have sets as values. UML also considers multisets, that is, they can have $u_1$:

$C_{u_2}$:

$C_{u_3}$:

(This is not an object diagram in the sense of our definition because of the requirement on the edges $E$. Extension is straightforward but tedious.)

• We allow to give the valuation of $C_0$, $C_1$-or $C^*$-typed attributes in the values compartment.

• Allows us to indicate that a certain $r$ is not referring to another object.

• Allows us to represent "dangling references", i.e. references to objects which are not alive in the current system state.

• We introduce a graphical representation of $\emptyset$ values.

• If we only have a picture as below, we typically assume that it's meant to be an object diagram wrt. some signature and structure.

In the example, we can conclude (by "goodwill") that the author is referring to some signature $/CB = (/CC, /BV, V, \{\})$ with at least

• $\{C,D\} \subseteq /BV$, $T \in /CC$, $\{x: C^*, p: C^*, z: T\} \subseteq V$, $\{x\} \subseteq \text{atr}(C)$, $\{p, z\} \subseteq \text{atr}(D)$, and a structure with

• $\{u_1, u_2\} \subseteq /BW(C)$, $u_3 \in /BW(D)$, $0 \in /BW(T)$. 

Example: Object Diagrams for Documentation

Example: Data Structure

[Schumann et al., 2008]

Example: Illustrative Object Diagram

[Schumann et al., 2008]
Revise to:

Meeting

w. |

Meeting w. (exists \rightarrow allInstances •)

\[ \text{fragment of OCL. less rich} \]

Constrain OCL, use a

[Cabot and Claris’ o, 2008]

\[ \text{Options:} \]

And now

\[ \text{move(newStart : Date)} \]

\[ \text{implies} \]

\[ \text{name : String} \]

\[ \text{location} = \text{title} \]

[1]

m

1

m

1

\[ \text{Cabot and Claris’ o, 2008} \]

\[ \text{http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/} \]

\[ \text{(C)Prof. Dr. P. Thiemann,} \]

\[ \text{otherwise.} \]

\[ \text{true} \]

\[ \text{false} = 0 \]

\[ \text{x : inv \text{D \ context \ •}} \]

\[ \text{isEmpty} \rightarrow \]

\[ \text{p.n : inv \text{C \ context \ •}} \]

\[ \text{isEmpty} \rightarrow \]

\[ \text{n : inv} \]

\[ \text{C : w (exists \rightarrow allInstances {}} \]

\[ \text{Cabot and Claris’ o, 2008} \]

\[ \text{http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/} \]

\[ \text{(C)Prof. Dr. P. Thiemann,} \]

\[ \text{otherwise.} \]

\[ \text{true} \]

\[ \text{false} = 0 \]

\[ \text{x : inv \text{D \ context \ •}} \]

\[ \text{isEmpty} \rightarrow \]

\[ \text{p.n : inv \text{C \ context \ •}} \]

\[ \text{isEmpty} \rightarrow \]

\[ \text{n : inv} \]

\[ \text{C : w (exists \rightarrow allInstances {}} \]

\[ \text{Cabot and Claris’ o, 2008} \]

\[ \text{http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/} \]

\[ \text{(C)Prof. Dr. P. Thiemann,} \]

\[ \text{otherwise.} \]

\[ \text{true} \]

\[ \text{false} = 0 \]

\[ \text{x : inv \text{D \ context \ •}} \]

\[ \text{isEmpty} \rightarrow \]

\[ \text{p.n : inv \text{C \ context \ •}} \]

\[ \text{isEmpty} \rightarrow \]

\[ \text{n : inv} \]

\[ \text{C : w (exists \rightarrow allInstances {}} \]

\[ \text{Cabot and Claris’ o, 2008} \]

\[ \text{http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/} \]

\[ \text{(C)Prof. Dr. P. Thiemann,} \]

\[ \text{otherwise.} \]
Expressive Power:

"Pure OCL expressions only compute primitive recursive functions, but not recursive functions in general."

\cite{CengarleandKnapp,2001}

Evolution over Time:

```
finally self.x > 0
```

Proposals for fixes e.g. \cite{FlakeandMuller,2003} (Or: Sequence diagrams.)

Real-Time:

"Objects respond within 10s"

Proposals for fixes e.g. \cite{CengarleandKnapp,2002}

Reachability:

"After insert operation, nodes shall be reachable."

Fix: add transitive closure.

Concrete Syntax:

"The syntax of OCL has been criticized – e.g., by the authors of Catalysis [\ldots] – for being hard to read and write."

"OCL's expressions are stacked in the style of Smalltalk, which makes it hard to see the scope of quantified variables."

Navigations are applied to atoms and not sets of atoms, although there is a collect operation that maps a function over a set.

"Attributes, [\ldots], are partial functions in OCL, and result in expressions with undefined values."

\cite{Jackson,2002}