IMPROVED SEQUENTIAL SITUATION CALCULUS

http://www.formal.stanford.edu/jmc/sitcalc.html

- An action is a kind of event, e.g. $Does(Alice, Block1)$.
- Internal events are triggered by occurrence axioms and replace domain constraints.
- Circumscribe a situation at a time.
- You get improved elaboration tolerance.

STUFFY ROOM SCENARIO

There are two vents and actions that block and unblock each vent.

Domain constraint: If both vents are blocked, the room is stuffy.

$$Blocked1(s) \wedge Blocked2(s) \rightarrow Stuffy(s).$$

Problem for oversimple sitcalc: When the second vent is blocked, change can be minimized in two ways. (1) The room becomes stuffy. (2) When vent2 is blocked, vent1 becomes unblocked, also minimizing change.

FORMALIZING A BUZZER IS STRAIGHTFORWARD.

Effect axioms:

$$\begin{array}{l} On(R, Result(Onn(R), s)) \\ \neg On(R, Result(Offf(R), s)) \\ On(Sw, Result(Onn(Sw), s) \\ \neg On(Sw, Result(Offf(Sw), s)) \end{array} \quad (1)$$

Occurrence axioms:

$$\begin{array}{l} \neg On(Sw, s) \wedge On(R, s) \rightarrow Occurs(Offf(R), s) \\ On(Sw, s) \wedge \neg On(R, s) \rightarrow Occurs(Onn(R), s)) \\ On(R, s) \wedge On(Sw, s) \rightarrow Occurs(Offf(Sw), s) \\ \neg On(R, s) \wedge \neg On(Sw, s) \rightarrow Occurs(Onn(Sw), s) \end{array} \quad (2)$$

YOU CAN'T DO MUCH WITH A BUZZER

- Trace its action
- To turn it on or off requires another switch.
- Regard, "The buzzer is buzzing as a state."

CIRCUMSCRIBING IN EACH SITUATION

$$Foo' \leq_s Foo \equiv (\forall x\ y)(Foo'(x, y, s) \rightarrow Foo(x, y, s)). \quad (3)$$

Then the circumscription of $Foo(x, y, s)$ takes the form

$$Axiom(Foo) \wedge (\forall Foo')(Axiom(Foo') \rightarrow \neg(Foo' <_s Foo)), \quad (4)$$

where as is usual with circumscription,

$$(Foo' <_s Foo) \equiv (Foo' \leq_s Foo) \wedge (Foo' \neq Foo). \quad (5)$$

WHAT GETS MINIMIZED?

The present examples are too simple to fully illlustrate the formalism.

In general we minimize $Occurs$. The frame problem is solved by introducing $Changes(e, f, s)$ and minimizing $Changes$. The qualification problem is solved by introducing $Prevents(f,$ and minimizing $Prevents$. In initial situations $S0$, we minimize $Holds$ in a lengthier formalism where we write $Holds(fluent, s)$ instead of just $fluent(s)$. See the paper www.formal.stanford.edu/jmc/sitcalc.html for details.

STUFFY ROOM AXIOMS

Effect axioms:

$$\begin{array}{l} Blocked1(Result(Block1, s)) \\ Blocked2(Result(Block2, s)) \\ \neg Blocked1(Result(Unblock1, s)) \\ \neg Blocked2(Result(Unblock2, s)) \\ Stuffy(Result(Getstuffy, s)) \\ \neg Stuffy(Result(Ungetstuffy, s)) \end{array} \tag{6}$$

Occurrence axioms:

$$\begin{array}{l} Blocked1(s) \wedge Blocked2(s) \wedge \neg Stuffy(s) \\ \rightarrow Occurs(Getstuffy, s) \\ (\neg Blocked1(s) \vee \neg Blocked2(s)) \wedge Stuffy(s) \\ \rightarrow Occurs(Ungetstuffy, s) \end{array} \tag{7}$$

AN ELABORATION GIVING OSCILLATING STUFFINESS

Suppose Bob is unhappy when the room is stuffy, but Alice is unhappy when the room is cold. The stuffy room axioms tolerate adding the following axioms which makes Vent1 oscillate between open and closed.

$$\begin{array}{c} Stuffy(s) \rightarrow Occurs(Does(Bob, Unblock1), s) \\ Unblocked1(s) \rightarrow Occurs(Getcold(Alice), s), \\ Cold(Alice, (Result(Getcold, s)), \\ Cold(Alice, s) \rightarrow Occurs(Does(Alice, Block1), s). \end{array} \tag{8}$$

SETTLING DOWN—OR NOT

- $Result(e, s)$—the immediate result of an event
- $Result^*(e, s)$—the result after internal events are done.
- $Next(s)$—Result of the event that occurs in s.
- $Next^*(s)$—Next situation after internal events are done.
- When the situation doesn't settle down, $Result^*$ and $Next^*$ are undefined. The buzzer doesn't settle down, and neither does the stuffy room scenario with Alice and Bob.

REMARKS

- Only processes where the situation *settles down* after each action can be described using domain constraints.

- The buzzer can't be described at all with domain constraints, because it never settles down.

- The original stuffy room can be described inconveniently using domain constraints, but the Alice and Bob version cannot be obtained as an elaboration, because the domain constraints introduce a contradiction.

- Thanks to many people who have listened to and reacted to my harangues advocating occurrence axioms.