IMPROVED SEQUENTIAL SITUATION CALCULU

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

- An action is a kind of event, e.g. $Does(Alice, Bloc$

- Internal events are triggered by occurrence axioms
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 unbl
each vent.

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

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

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

FORMALIZING A BUZZER IS STRAIGHTFORWAR

Effect axioms:

$$On(R, Result(Onn(R), s))$$
$$\neg On(R, Result(\mathit{Offf}(R), s))$$
$$On(Sw, Result(Onn(Sw), s)$$
$$\neg On(Sw, Result(\mathit{Offf}(Sw), s))$$

Occurrence axioms:

$$\neg On(Sw, s) \wedge On(R, s) \rightarrow Occurs(\mathit{Offf}(R), s)$$
$$On(Sw, s) \wedge \neg On(R, s) \rightarrow Occurs(Onn(R), s))$$
$$On(R, s) \wedge On(Sw, s) \rightarrow Occurs(\mathit{Offf}(Sw), s)$$
$$\neg On(R, s) \wedge \neg On(Sw, s) \rightarrow Occurs(Onn(Sw), s)$$

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

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

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

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

where as is usual with circumscription,

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

WHAT GETS MINIMIZED?

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

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

STUFFY ROOM AXIOMS

Effect axioms:

$$Blocked1(Result(Block1, s))$$
$$Blocked2(Result(Block2, s))$$
$$\neg Blocked1(Result(Unblock1, s))$$
$$\neg Blocked2(Result(Unblock2, s))$$
$$Stuffy(Result(Getstuffy, s))$$
$$\neg Stuffy(Result(Ungetstuffy, s))$$

Occurrence axioms:

$$Blocked1(s) \wedge Blocked2(s) \wedge \neg Stuffy(s)$$
$$\rightarrow Occurs(Getstuffy, s)$$
$$(\neg Blocked1(s) \vee \neg Blocked2(s)) \wedge Stuffy(s)$$
$$\rightarrow Occurs(Ungetstuffy, s)$$

AN ELABORATION GIVING OSCILLATING STUFFINESS

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

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

SETTLING DOWN—OR NOT

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

REMARKS

- Only processes where the situation *settles down* af-
ter each action can be described using domain con-
straints.

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

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

- Thanks to many people who have listened to and acted to my harangues advocating occurrence axio