From PFA to HJJ: known unknowns

Cliff B Jones

School Computing Science
Newcastle University
UK

ICSE 2009-05-19

Contents

1. Background

2. Successes

3. Open issues

4. Direction

Trace the *evolving* HJJ ideas

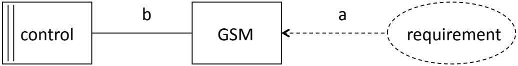
Hayes/Jackson/Jones

Starting point

- shared view that software development not perfect!
- ... for "closed systems", theory is available
- recognition that "open systems" present issues not resolved by:
  - pre/post conditions
  - data reification
  - (sequential) operation decomposition
- Problem Frame Approach [Jac03, Jac00]
- Hayes/Mahony notation for continuous time reasoning
- rely/guarantee "thinking"

An unlikely union?

Intuitive PFA diagrams



a: {pos: Height}

b: control ! {dir: up | down, motor: on | off}
GSM ! {top: Bool, bot: Bool}

Hayes/Mahony notation

often face reality where values vary continuously — cf. Duration Calculus

$$
\begin{array}{l}
SluiceGateRequirement \triangleq \\
\quad \lambda T : Interval(Time) \cdot \\
\qquad \forall I : Interval(T) \cdot \\
\qquad\qquad \# I \geq 6 hours \;\; \Rightarrow \\
\qquad\qquad\qquad \int_I (pos = \textbf{open}) \in \frac{1}{6} * \# I \pm (\textbf{max\_open} + \# I * \textbf{Error}) \wedge \\
\qquad\qquad\qquad \int_I (pos = \textbf{closed}) \in \frac{5}{6} * \# I \pm (\textbf{max\_closed} + \# I * \textbf{Error})
\end{array}
$$

Rely/guarantee proof rules

. . . compositional

$$\boxed{\textit{Par-I}}\ \frac{\begin{array}{l}\{P, R \lor Gr\}\ sl\ \{Gl, Ql\} \\ \{P, R \lor Gl\}\ sr\ \{Gr, Qr\} \\ Gl \lor Gr\ \Rightarrow\ G \\ \overleftarrow{P} \land Ql \land Qr \land (R \lor Gl \lor Gr)^{*}\ \Rightarrow\ Q\end{array}}{\{P, R\}\ sl\ ||\ sr\ \{G, Q\}}$$

"R/G Approach" — not a specific set of rules

Contents

1. Background
2. Successes
3. Open issues
4. Direction

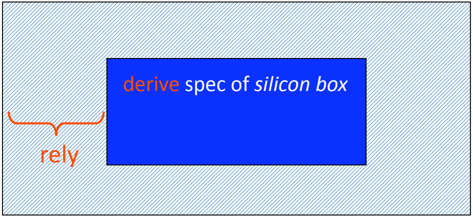
The good news

HJJ outline

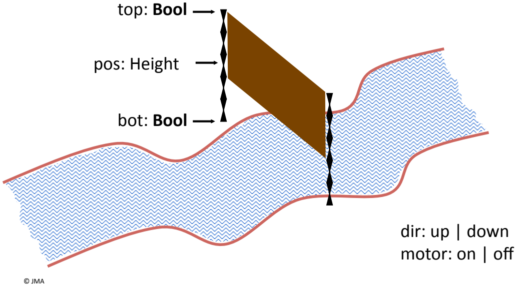
- examples of “deducing specifications” from the wider system
- ground understanding in external world (as in PFA)
- make (agree and *record*) assumptions about physical components
- . . . but *do not model them in detail*
- derive the specification of “silicon box”
- . . . and certainly don’t just jump into specifying the silicon box
- we’ll come back to: when assumptions not satisfied

Simple message

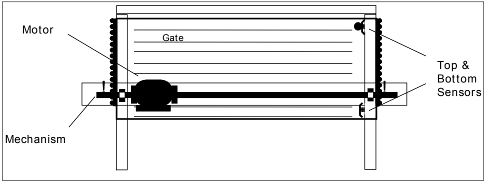
specify *overall* system



Intuition of Sluice Gate System



No water!



© MAJ

Sluice Gate System

- relatively simple system
- no difficulty (given above notations) to specify normal behaviour
- SGS presents two interesting fault-tolerance issues
  - slow (stuck?) gate
  - top/bottom sensors “open circuit” (or flickering)
- at one level, just weaker R/G specs
- some help from “Deontic implications”

Contents

1 Background

2 Successes

3 Open issues

4 Direction

Known unknowns

- there is no fixed way of saying how wide a "system" is
- Manuel's objection: where is the "method"
- fault tolerance (FT): in one sense, just weaker rely conditions
- view behaviour in "layers"
- ... not just a big conjunction of implications
- some traction with "faults as interference"
- ... dates back to ISAT study
- but this leaves the issue of describing "phase changes"
- technical questions about (timing) handover
- "discussion" about separating FT behaviour from normal
- probabilities (R/G-A?)

Contents

1 Background

2 Successes

3 Open issues

4 Direction

Method

- erudite discussions (on Descartes et al.) twixt Michael and Manuel
- see some steps:
  1. choose a system perimeter
  2. specify/agree ideal (physical world) requirements
  3. define/agree assumptions
  4. repeat steps 2–3 with weaker assumptions
  5. handle “composition issue”
- Manuel talks of “LFTS”
- I recall Michael (about JSP) asking that “methods be normative”!
- Michael’s “operational principles” (radical design)
  - . . . more later?

TimeBands

Alan Burns and Ian Hayes

- powerful idea
- view different granularities of time
- . . . add “precision”, etc.
- certainly promising
- like the way exceptions seen as crossing TimeBands
- but (for my taste) getting overly complicated
- “events” vs. “activities”
- “timeless TimeBands” (fuzzy everything)
- now suspect that we only need “precision”

Teleo-Reactive notation

Nilsson (via Keith Clark) to Ian Hayes

- originally for programming robots
- state changes cause guards to be re-evaluated
- (these are not "action systems")
- neat way of showing the interrupts for exceptions
- in one sense, shifts the question to semantics of TR
- . . . but puts it in one place

Thank yous

- thank you (Michael) for (JSP, ...) PFA
- . . . and the challenges
- . . . and the discussions
- *Dubium Sapientiae initium* Descartes

References

Ian Hayes, Michael Jackson, and Cliff Jones.
Determining the specification of a control system from that of its environment.
In Keijiro Araki, Stefani Gnesi, and Dino Mandrioli, editors, *FME 2003: Formal Methods*, volume 2805 of *Lecture Notes in Computer Science*, pages 154–169. Springer Verlag, 2003.

Michael Jackson.
Problem Frames: Analyzing and structuring software development problems.
Addison-Wesley, 2000.

Michael Jackson.
Aspects of system description.
In Annabelle McIver and Carroll Morgan, editors, *Programming Methodology*, pages 137–160. Springer Verlag, 2003.

Cliff B. Jones, Ian J. Hayes, and Michael A. Jackson.
Deriving specifications for systems that are connected to the physical world.
In Cliff B. Jones, Zhiming Liu, and Jim Woodcock, editors, *Formal Methods and Hybrid Real-Time Systems: Essays in Honour of Dines Bjørner and Zhou Chaochen on the Occassion of Their 70th Birthdays*, volume 4700 of *Lecture Notes in Computer Science*, pages 364–390. Springer Verlag, 2007.