CONCERNS ABOUT ASPECTS

Daniel Jackson
Computer Science & Artificial Intelligence Laboratory
MIT

Brown-Northeastern Aspects Day
Boston / March 8, 2005

what is AOP?

Afghan Online Press

http://www.aopnews.com

- Afghan News
- Today
- Yesterday
- Events
- Archives
- Weather
- AOP Links
- News Photos
- News Search
- Dari / Pashto News Services

Last update: 3/7/2005 (6:52 AM PST)

Advanced Observing Program

at Kitt Peak Visitor Center

Introduction

The Kitt Peak Visitor Center offers an advanced observing program geared towards the amateur astronomer interested in using a large telescope with state-of-the-art instruments. No previous experience in astronomy is necessary. You can observe at the world's largest optical observatory under some of North America's finest skies! Guests who participate in this program are treated as visiting astronomers, and they have complete access to the Visitor Center's considerable resources. Observe from an excellent site, dine with other astronomers, and above all enjoy exploring the universe.

Australian Optometric Panel

a nationwide panel of vision-care professionals

AUSTRALIAN OPTOMETRIC PANEL

- What is AOP?
- Member Directory
- Eye Examinations
- Frames & Lenses
- Contact Lenses
- Ortho-Keratology
- Keratoconus

What does the Australian Optometric Panel do?

AOP employs a variety of strategies to fulfil its' mission. Principal among these are:

1. Sharing of knowledge, experience and industry intelligence.
2. Problem-solving through issues analysis and creativity.
3. Use of bench-marking within the group.
4. Communication with equivalent groups overseas to know and match international best practice.
5. Use of expert guest speakers in relevant fields of finance, taxation, QA and TQM, marketing, business and human resources management, and individual personal development.
6. Training sessions for staff, using AOP members and specialist consultants.


ANCIENT ORDER OF THE PTERODACTYL
Flying Since the World Was Flat
MCMLXXVII

Action for Older Persons

Action for Older Persons (AOP) is an innovative and visionary organization that enhances the lives of adults and empowers them to prepare for the future. The mission of AOP is to assist adults in enhancing their lives by promoting financial security, physical and emotional well-being, and self-sufficiency throughout their lives. This is achieved through programs and services, education, advocacy, identifying and addressing needs, and community collaboration. AOP is a private, nonprofit, United Way member agency which was founded in 1967.


australian
orangutan
project

on further study of AOP
materials, I learned that AOP is ...

some kind of religious activity

something that involves serious pain

Envíe un correo al Presidente de la AOP

Los Beneficios de ser miembro de la AOP: <Aquí>

Santa Apolonia - Día del Odontólogo

Al conmemorarse en esta fecha el martirio de Santa Apolonia, ocurrido en el año 248 D.C., la ocasión es propicia para felicitar a todos los miembros de la AOP en su día, y hacer votos por que nuestra profesión siga por el sendero de progreso manteniendo sus más altos ideales en beneficio de la sociedad panameña. Aprovechemos la oportunidad para encomendarnos a nuestra santa patrona, y renovar así nuestro compromiso de resguardar la salud bucal de los panameños, profundizando en el aprendizaje permanente; en la dedicación a nuestros pacientes y en el esfuerzo permanente por los más nobles propósitos de la profesión odontológica.

conclusions

what problem does (or should) AOP address?

- more powerful text editing?
- overcoming limitations of OOP?
- factoring out non-functional goals?
- separating functional concerns

what's the key challenge?

- in requirements: identifying the concerns
- in design: maintaining separation -- modularity

what progress so far?

- views & problem frames

separation of concerns

Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects... It is what I sometimes have called 'the separation of concerns' which, even if not perfectly, is yet the only available technique for effective ordering of one's thoughts that I know of.

Dijkstra. On the role of scientific thought.
EWD 447, 30th August 1974

looking for concerns to separate ...

an argument
- code can be divided up in many ways
- Java only supports one way
- so let's support the others too

like the old definition of AI
- is a concern anything you can't separate now?
- are runtime assertions really a concern?

start from the concerns instead
- some are already separated (syntax/semantics)
- some are too tricky to separate (performance)

so what's left?

where do concerns come from?

from the software development problem itself!

- viewpoints of stakeholders?
- decomposition for simplest description?
- division into recognizable subproblems?

viewpoints

observation

- stakeholders have different perspectives
- often mutually inconsistent

so

- encourage, don't suppress, separate descriptions
- use tools to reconcile
- or tolerate inconsistency

examples

- viewpoints [Finkelstein et al, 1992]
- manage inconsistency [Easterbrook & Nuseibeh, 1995]
- reasoning with inconsistency [Chechik, 2001]
- reasoning with viewpoints in Z [Ainsworth, 1994]

views in declarative specification

basic idea

- exploit conjunction in declarative specs
- select the state representation that suits the operation
- group operations by representation into views

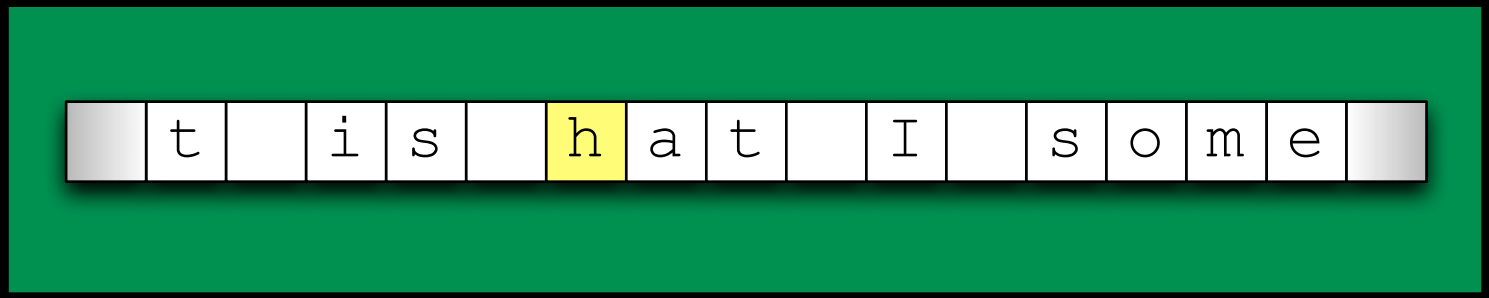
Structuring Z Specifications with Views
Daniel Jackson, TOSEM 1995

example: an editor

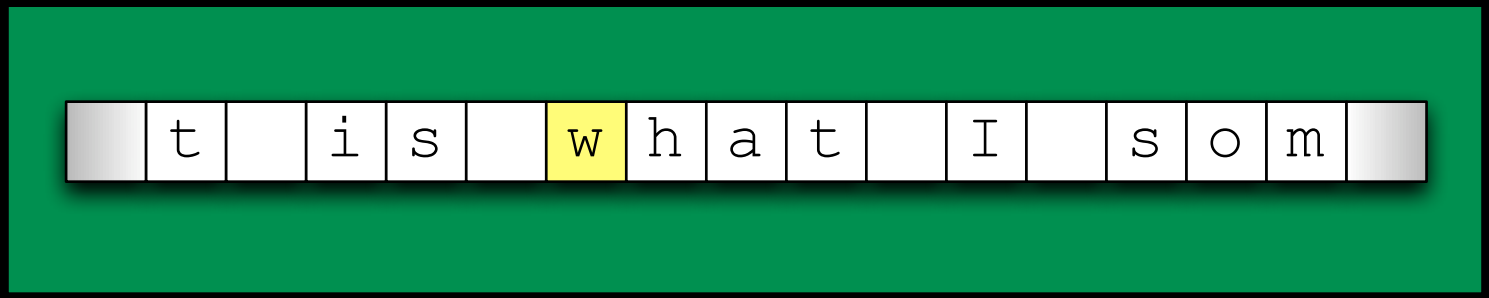
It is what I sometimes have called ‘the separation of concerns’

text insertion

before insertion:

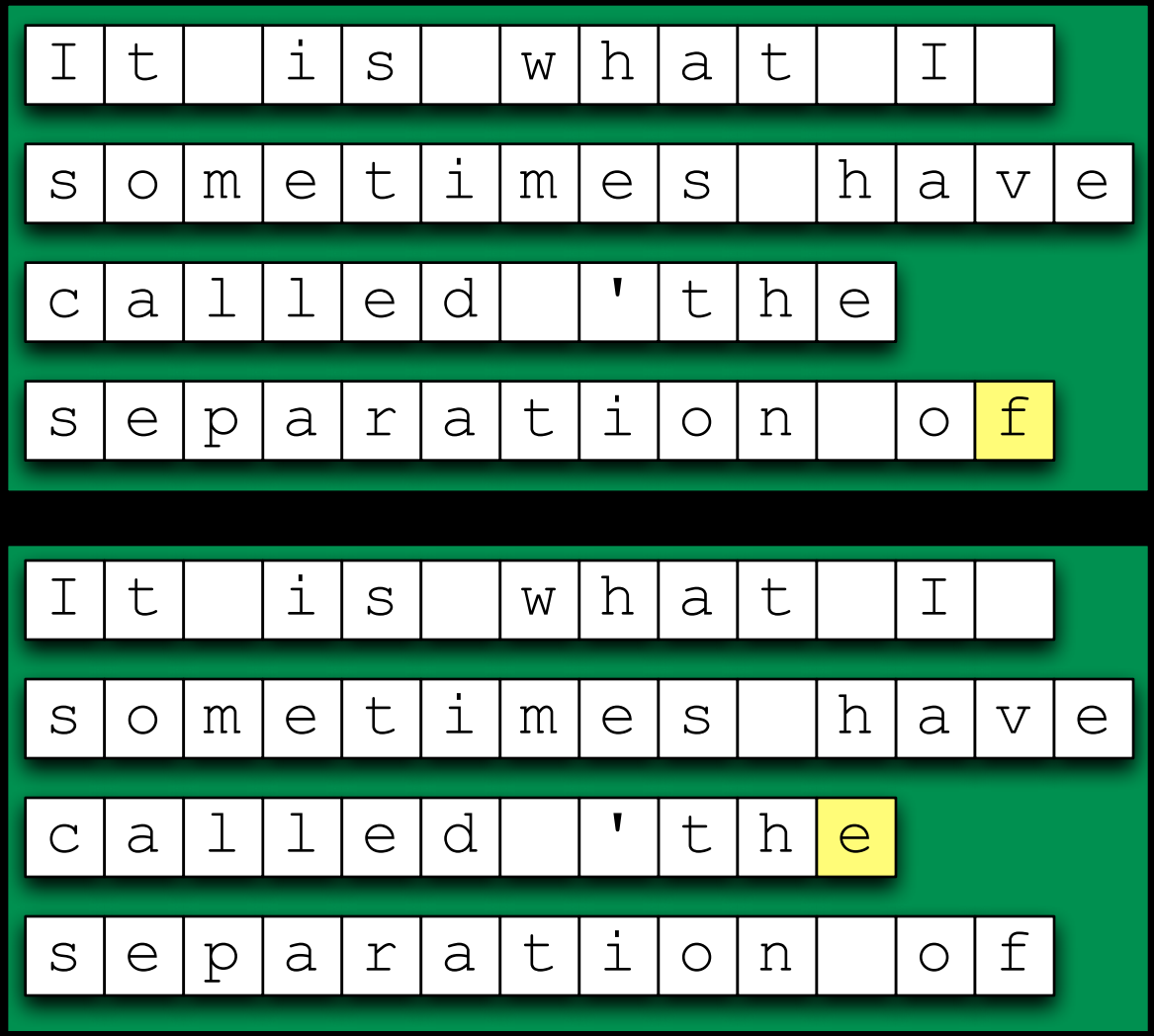


after insertion:



cursor up

before and after:



view 1: sequence of characters

File

$left, right : \text{seq } Char$

File.csrRight

$\Delta\, File$

$right \neq \langle\,\rangle \;\land\; right' = tail(right)$

$left' = left \frown \langle head(right) \rangle$

File.insertChar

$\Delta\, File$

$c? : Char$

$left' = left \frown \langle c? \rangle \;\land\; right' = right$

view 2: sequence of lines

Grid

$$lines : \text{seq seq } Char$$
$$x, y : \mathbb{N}$$

$$lines \in wrapped \land y \in \text{dom } lines \land x \in \text{dom } lines[y]$$

Grid.csrUp

$$\Delta Grid$$

$$y > 1 \land y' = y - 1 \land x' = min(x, \#lines[y'])$$
$$lines' = lines$$

combining the views

Flatten

File, *Grid*

$$left \frown right = \frown lines \quad \land \quad \#left = x + \Sigma i : 1..y-1 \bullet \#lines[i]$$

Editor

Flatten

$$\forall ls : \text{seq seq } Char \bullet Flatten[ls/lines] \Rightarrow \neg\ (ls > lines)$$

$$insertChar = [\Delta Editor \mid File.insertChar]$$
$$csrRight = [\Delta Editor \mid File.csrRight]$$
$$csrUp = [\Delta Editor \mid Grid.csrUp]$$
$$delEol = [\Delta Editor \mid Grid.delEol]$$

where did this go?

technical problems
- easy to get it wrong
- accidental overconstraint
- Alloy with unsat core might help

wrong level of granularity?
- neat, but not many problems like this?
- most view compositions much simpler

big gap to implementation
- Robert Nord [CMU, 1992]
- not much since?

example: package router

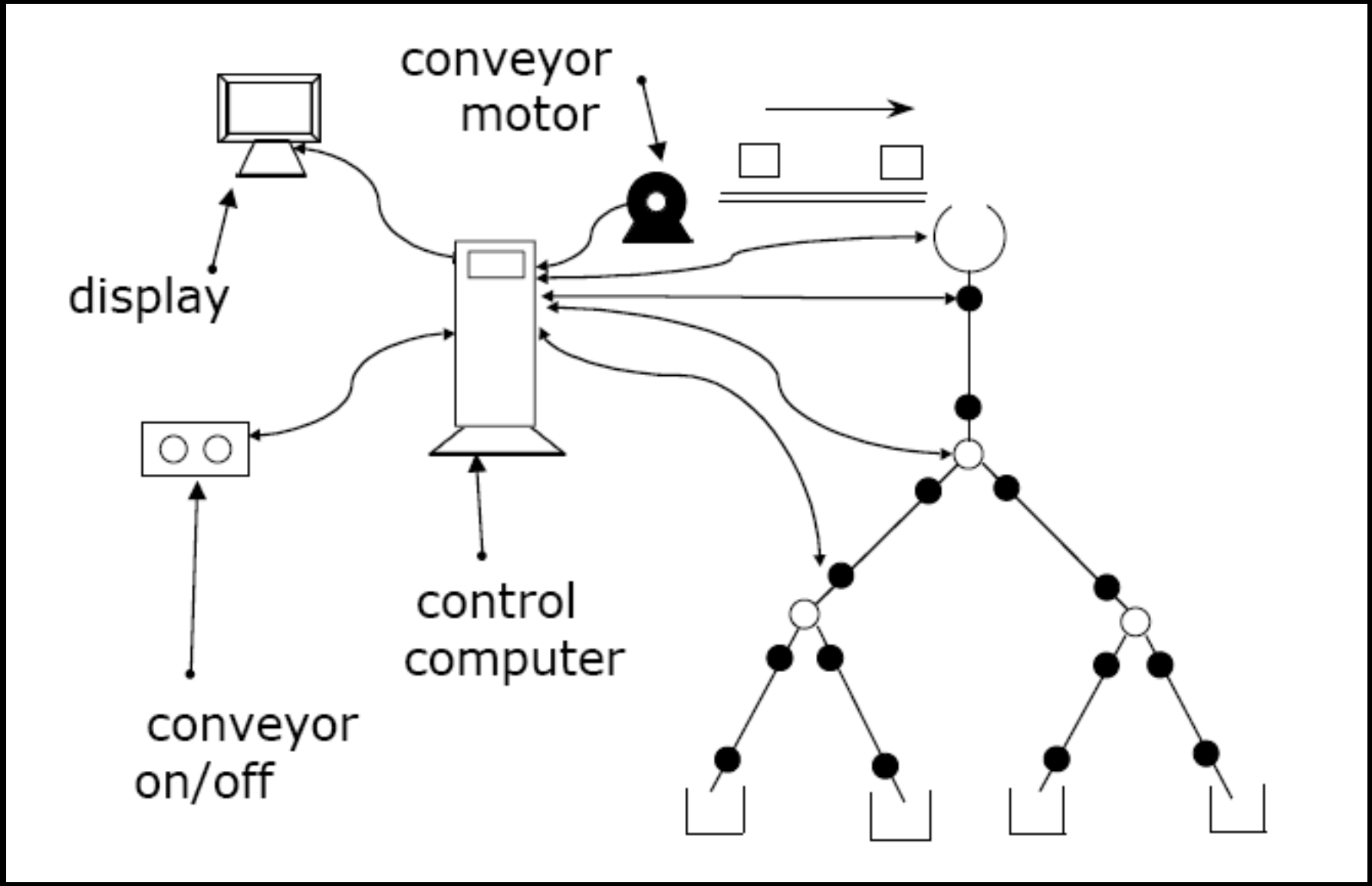
problem frame analysis

Michael Jackson. Problem Frames. Addison-Wesley, 2001.

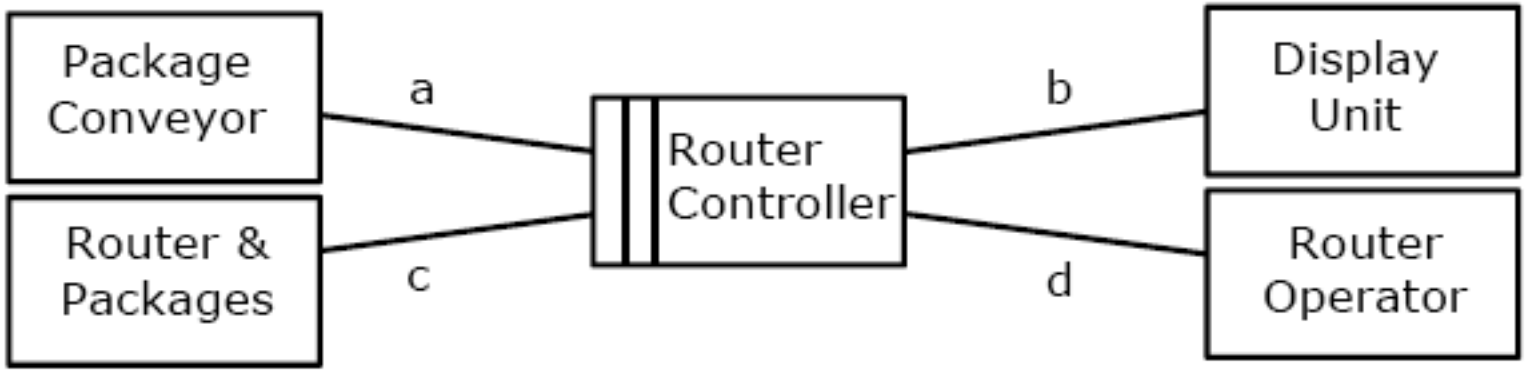
specification with views

Daniel Jackson and Michael Jackson.
Problem Decomposition for Reuse. SE Journal, 1996

package router equipment



context diagram



a: RC! {OnC, OffC}
b: RC! {ShowPkgId, ShowBin, ShowDestn}
c: RC! {LSw(i), RSw(i)}
RP! {SendLabel(p,l), LId(l,i), LDest(l,d), SwPos(i), SensOn(i)}
d: RO! {OnBut, OffBut}

package router subproblems

3 basic subproblems

- conveyor control
- routing packages
- reporting misrouted packages

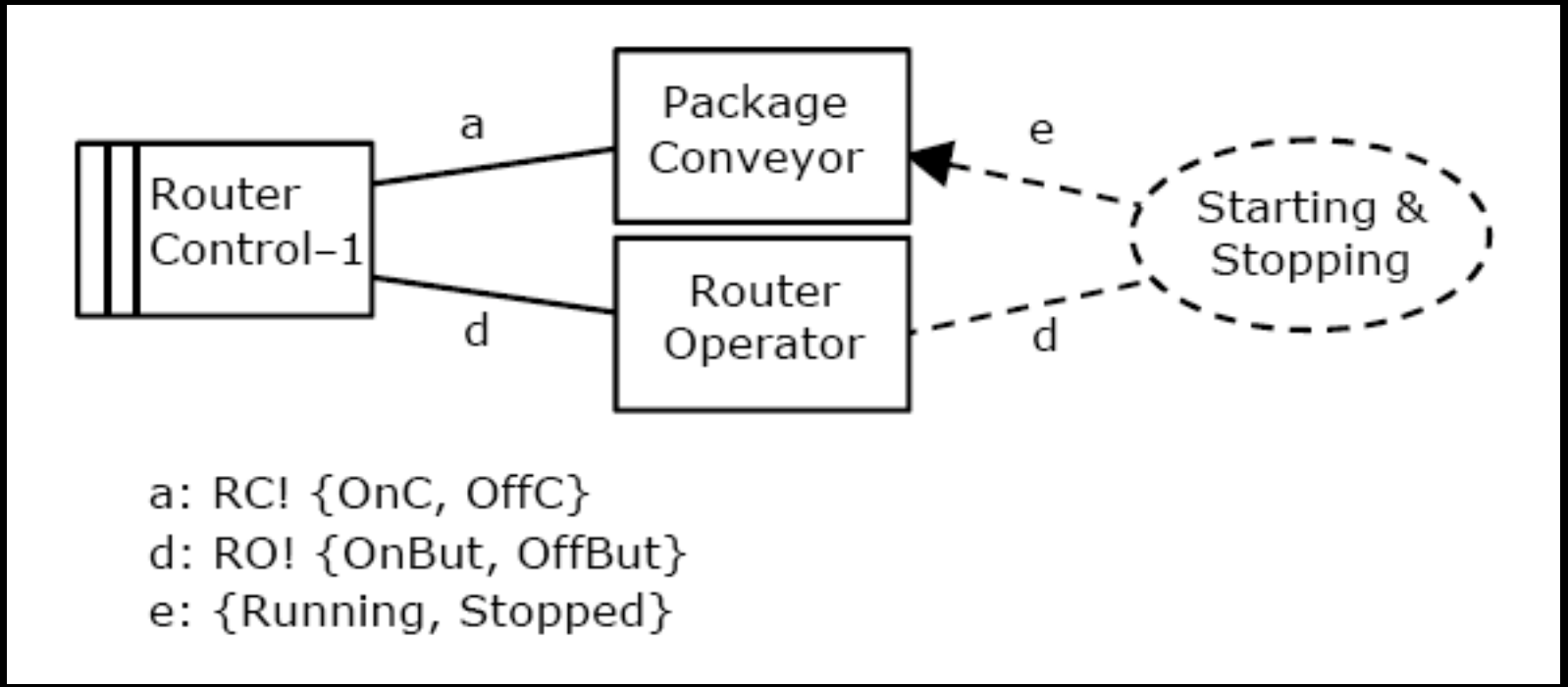
conveyor control

requirement

› stop and start conveyor as commanded

kind of problem

› ‘commanded behaviour’



routing packages

requirement
- each package arrives at the right bin

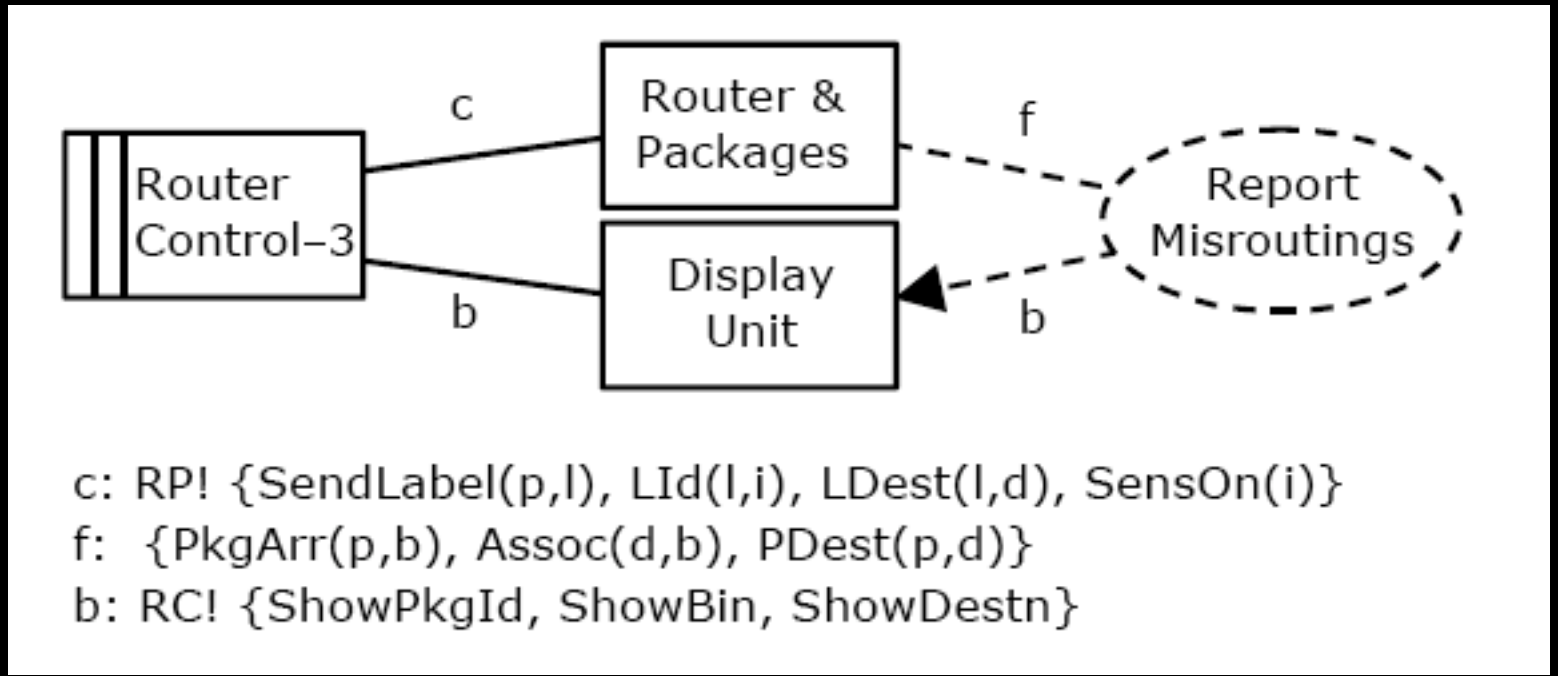
kind of problem
- ‘simple control’
- ‘simple information system’ as subproblem

c: RC! {LSw(i), RSw(i)}
RP! {SendLabel(p,l), LId(l,i), LDest(l,d), SwPos(i), SensOn(i)}
f: {PkgArr(p,b), Assoc(d,b), PDest(p,d)}

reporting misroutings

requirement
› report misroutings

kind of problem
› ‘simple information system’



maintaining SOC in implementation

challenges

- modularity, modularity, modularity

because we want to

- build concerns separately
- check them separately
- modify them separately

modularity of concerns

independence
› a concern shouldn't depend on existence of others

robustness
› refactoring one concern shouldn't affect another

symmetry
› no arbitrary precedence of one concern over another

encapsulation
› within concern, make properties safe from interference

use of names

best to avoid names of methods & classes?

- makes concerns fragile
- dependent on naming and packaging structure

use abstract linkage points instead?

- declared within a concern
- access control mechanisms

or even better, no names?

- Jonathan Edwards. Subtext: programming without text.
- http://www.subtextual.org

CSP: another model for AOP?

```
M = coin -> coin ->
        (selectChoc -> choc -> M)
    |   (selectNuts -> nuts -> M)

U = coin -> coin -> selectChoc -> U
```

features

- each process has its own, independent meaning
- has an alphabet that limits interference
- interference can be bounded: reduces non-det

reasoning

$$\frac{P \text{ satisfies } S,\ P' \text{ satisfies } S'}{P \,||\, P' \text{ satisfies } S \wedge S'}$$

concluding thoughts

research challenges
- identifying concerns
- modular implementation

what's AOP like?
- data abstraction? functional programming?
- object orientation?
- inheritance?
- C++ templates?
- model driven architecture tools?