Extrinsically adaptable systems

R. Poss
Universiteit van Amsterdam
Motivation

- Observation 1: as “consumer” systems get form, they lose customizability
- Observation 2: at least in computer architecture, we lack innovators
- Is there a cause-effect relationship?
- Can we improve the situation?
Extrinsic adaptations

Considering a system $S$ designed at time $t_0$ by party $p_0$

A new party $p_1 \neq p_0$ comes with a new requirement at time $t_1 > t_0$

$p_1$ changes $S$ “from the outside” without asking $p_0$
first = an extrinsic adaptation

this scenario is one of the main instruments of innovation

But when is this possible? What is needed?
Examples

- Angry birds vs. advertisements
- Text encoding in video player
- Code generation for a new architecture
- In general: 2ndary party comes up with new requirement, then implements change outside of the “preferred” process of primary party
General process

Initial specification

Initial implementation by primary parties

Deployed system visible to secondary parties

Expression of a new requirement by secondary parties

Canonical adaptation by primary parties

Modified system, change sanctioned a priori

Extrinsic adaptation by secondary parties

Modified system, changes not sanctioned, possibly idiosyncratic traits

Adoption of extrinsic adaptation by primary party

Modified system, change sanctioned a posteriori

woensdag 4 december 2013
Extrinsic vs invasive adaptations

- So far: extrinsic = modify a “composite from the outside”, don’t change the individual parts

- Assumption: the concepts “parts” and “composite” exist

- Do they?

- Yes: “extrinsic” has a meaning

- No: changes become invasive
Extrinsic vs invasive adaptations

Example: a smartphone is modified by installing a custom ROM

User 1 considers the system phone+ROM as a whole; customization violates the apparent system integrity and perceived as disruptive, unwelcome

User 2 knows about the difference between hardware and software; knows that the original ROM can be restored easily; may be curious about the customization
Extrinsic vs invasive adaptations

Why this matters: SUBJECTIVENESS

INVASIVE = BAD

EXTRINSIC = GOOD, or at least acceptable

Given an objective adaptation (eg it already happened):

If an audience is told there were no boundaries, the change is perceived as invasive, “bad”

If the audience knows about internal boundaries and the change seems outside of them, it is perceived as extrinsic and acceptable
Objectiveness of perception

- Surely we can evaluate whether changes are acceptable using objective criteria?
- Cost functions
  - Against budgets
  - Comparisons
- Two problems: consensus on cost function, consensus on pareto factors
Objectiveness of perception

Start: two candidate changes A and B

Cost function available? yes
  cannot compute cost

Pareto efficiency function agreed upon? yes
  no agreement on cost priorities

Budget is known? yes
  no budget

A ≠ B according to efficiency function? yes
  equivalent costs

A more extrinsic / less invasive than B? yes
  Pick A or B depending on cost

Pick the most extrinsic of A or B

Decision influenced by the (subjective) choice of component boundaries
What we have so far

• Adaptations happen
  • people want them, innovation process

• Extrinsic or invasive? Matters for acceptability
  • Cost function available? All good, but unusual
  • Otherwise, depends on the definition of what is a “part” and what is a “composite”

• Ultimately evaluation requires a choice of component boundaries
Component boundaries

PID 1234

httpd  mod_py

PID 1337

httpd  mod_py
Component boundaries

Boundary for the programmer: code bases
Component boundaries

PID 1234

Boundary for the sysadm: processes

PID 1337
Component boundaries

PID 1234

httpd  mod_py

PID 1337

httpd  wsgi

woensdag 4 december 2013
Component boundaries

Programmer view

httpd  mod_py

httpd  wsgi

change of interface at boundary: inconsistency, disruptive
Component boundaries

PID 1234

PID 1337

No impact on component identity with this chosen boundary: change acceptable
Acceptability & Component boundaries

- In general the choice of boundaries determines whether a change is invasive or extrinsic, thus “bad” or “acceptable”

- Multiple choices possible, usually not congruent

- Example used process vs code base, but also:
  - License vs processes
  - Physical packaging vs software sources
  - Vendor vs software sources

- In general problematic: law vs technical view
Acceptability & Component boundaries

- Example: consumer multimedia box
  - DRM, DMCA = legal boundary
  - But these boundaries do not exist from the logical, pure software perspective
  - So the pure software programmer is a-OK with software customization, Vendor usually not OK
  - All this only matters in the eyes of 3rd party non-technical observers, either bystanders or (worst case) a court of justice
  - This is a social / economical struggle, not technical

woensdag 4 december 2013
What we have so far (2)

- Adaptations happen
  - people want them, innovation process
- Extrinsic or invasive? Matters for acceptability
  - Cost function available? All good, but unusual
  - Otherwise, depends on the definition of what is a “part” and what is a “composite”
- Choice of component boundaries differs between interested parties: social, economical, sometimes political struggles
- Can we recognize this when it happens?
  - Can we quantify the struggle somehow?
Proposal: quantify extrinsic adaptability

- Difficult to do directly: “possible to modify” and “acceptable to modify” are too abstract
- BUT: we can quantify the opposing forces, which prevent adaptability, namely:
  - technical friction and
  - friction against transparency
- Extrinsic adaptability = 1 – “hindrance factor”
Technical friction

“Makes changes difficult to implement”

I found the following 3 fundamental forms:

- against alternate integration
- against extension
- against change resilience over time

NB: Substitution = extension + alternate integration

Can you think of more?
Technical friction

- Example mechanisms that oppose friction:
  - Tight integration
    (acts against alternate integration)
  - Warranty seals
    (all 3 forms)
  - Validation of component signatures
    (acts against extension)
  - Can you think of others?
Friction against transparency

- “Hiding the internal boundaries”
- Pushes changes towards invasiveness, or increases the costs
- I found the following 5 fundamental mechanisms: secrecy, physical barriers, obfuscation, encryption, alienation
- Can you think of others?
Weight for scoring

- Hindrance factor = Ftrans + N * Ftech
  - Ftrans = friction against transparency
  - Ftech = technical friction
  - N = envisioned number of changes over the lifetime of the system

- Extr. Adaptability = 1 - Ftrans - N * Ftech

- NB: only in the context of a specific selection of component boundaries!
Relevance today

- Computing systems:
  - Software systems = OK, largely adaptable
  - Software+hardware systems = hm....
  - Modularity and adaptability used to be welcome, desired; currently not due to insane competition in stagnant market and technology walls

- But it needn’t be so hard!
  - Adaptable systems: no vendor lock in, higher reusability, less waste, lower evolution costs.
  - This is about ETHICS!
    - (and maybe government regulation...)
How we can move forward

- Recognize “invasive” vs “extrinsic” subjective evaluations of changes
- Be conscious of discourse that shifts the perception of component boundaries
- Talk about factors that ease adaptability, or reduce friction to changes
- Quantitatively grade/review products and technology based on their hindrance factor
Thank you