

self-updating software
progress report
sept 98 - dec 99

Barbara Liskov & Daniel Jackson
February 19, 1999
MIT Lab for Computer Science

overview

today's problem: administering a PC

- installing time-consuming & error-prone
- interdependences make configuration fragile
- hard to update or remove applications

tomorrow's problem: pervasive updating

- on entry to airport, PDA downloads flight reminder code
- car downloads driving regimen from road sign
- handheld controller connects to devices in room

3-pronged approach

- archaeology
 - examine dependences in existing complex systems
- infrastructure
 - design & build infrastructure for self-updating
- applications
 - demonstrate & evaluate results in real applications

progress: candidate architecture

key elements

- device platforms

handle updates, shielding application updating mechanism

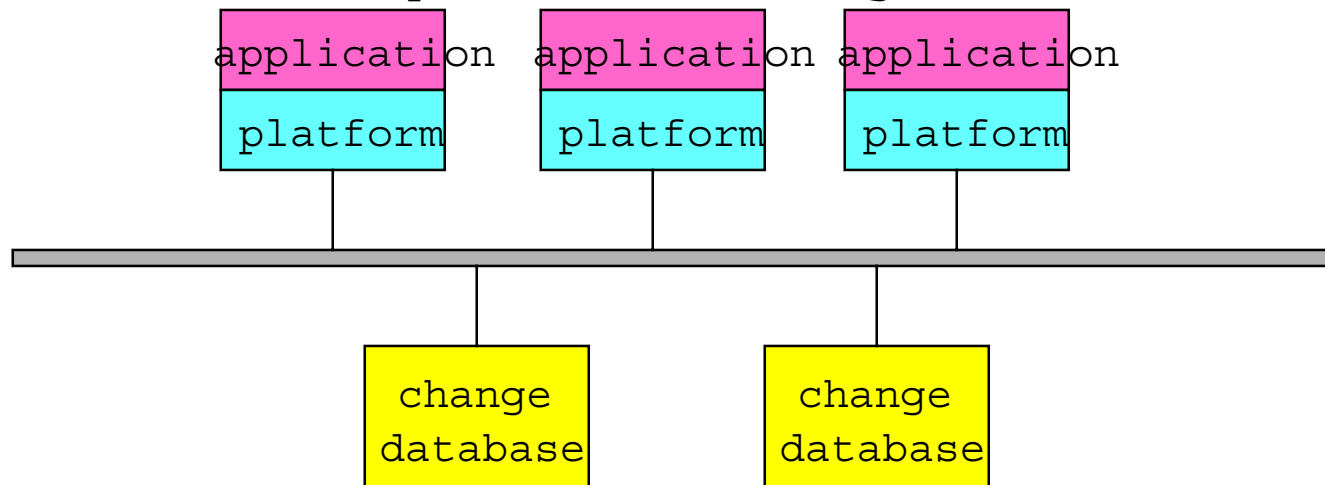
maintain database about local condition

- change databases

hold new objects (code & data), specs, dependencies, et

- transport

network, may include caching



plans

archaeology

- examine dependencies in Linux, Windows
use monitoring, logs, scripts, etc
- develop dependency model
captures common kinds of dependence

infrastructure

- elaboration of architecture
- implementation in Thor

demonstrations

- self-updating in Thor
- updatable PC telephone, using TAPI
- stock tracker: updating using agentJava