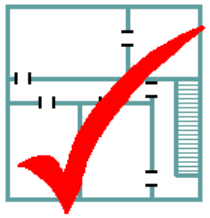
The Cambridge-MIT Institute



Autohan and Oxygen using Pebbles

Avoiding Feature Interaction

Funded partly by the CMI Cambridge MIT Institute
3 Year Project, Started October 2003.

API Reflection

o Many devices can now provide reflection of their API. This allows them to describe their passive properties and command interface to allow remote RPC or rendering of a graphic UI.

o We are investigating HLL and UML extensions to support system design and software enginering aspects when reflective APIs are present. It is quite a change from traditional compile/execute flows and raises a fundamental tension between compile time checking and runtime flexilibitly.

o API reflection techniques are, however, fairly stable: there exists XML RPC, WSDL and UPnP to name a few standards.

Code Reflection ?

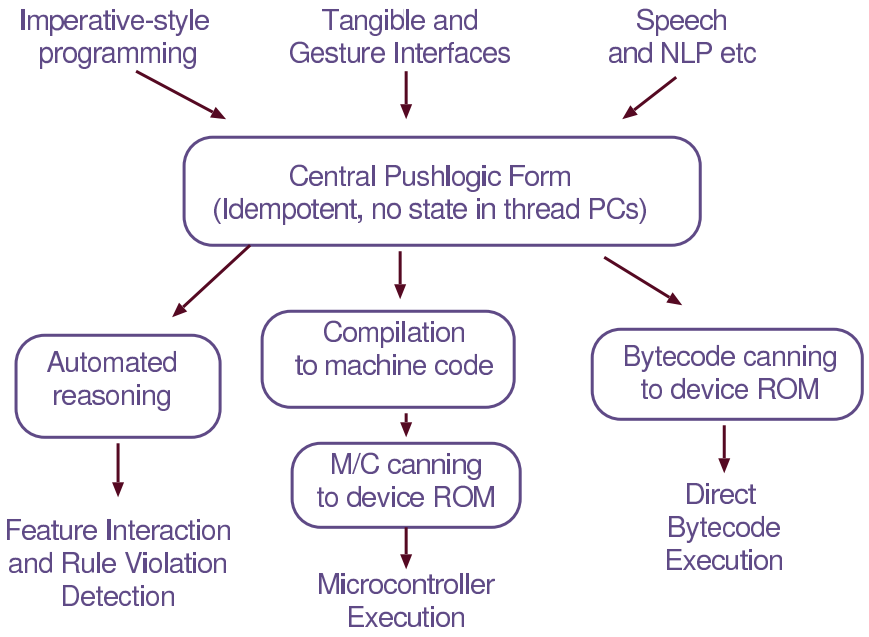
o Code Reflection of top-level embedded applications allows us to reason about how devices will interact.

o Devices must reflect (publish) their application scripts or a summary of them.

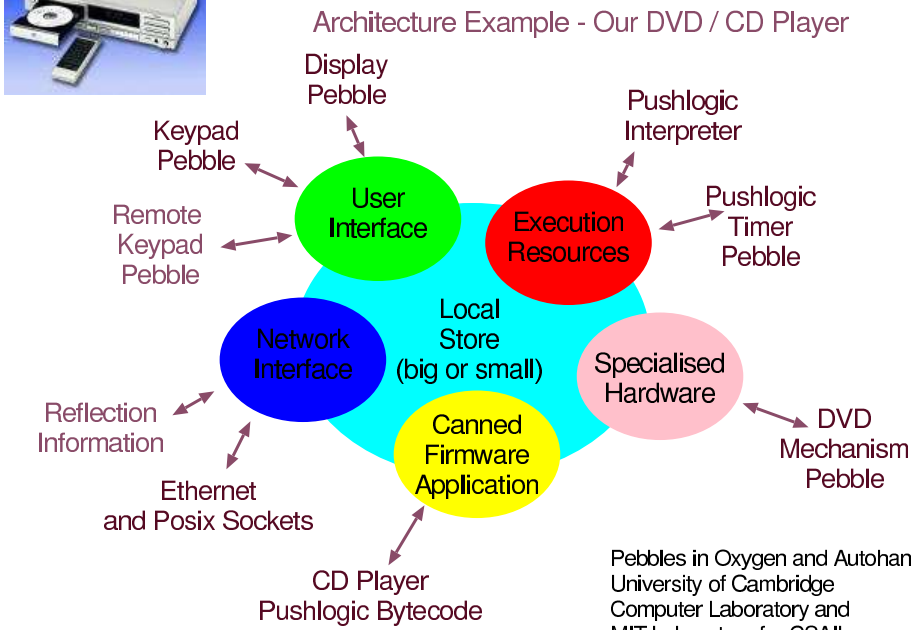
o Proof carrying code is one approach. We are instead exploring the use of code that is amenable to automated reasoning, including idempotent, stateless and declarative intermediate forms.

o Our current experimental scriping form is called 'Pushlogic'

Application Scripting Tool Flow



Device Componentisation into Pebbles



Pebbles in Oxygen and Autohan Project
University of Cambridge
Computer Laboratory and
MIT Laboratory for CSAIL