

Hidehiko Masuhara
Tokyo Tech

Context-Oriented Programming and Units of Adaptation

Hidehiko Masuhara

Tokyo Tech

What do self-adaptive systems adapt to?

from a PL's viewpoint

Background: Context-Oriented Programming (COP) [Hirschfeld'o8]

- for modularizing *context-dependent behaviors*

behaviors specific to *contexts*

- by providing language mechanisms
 - to affect multiple (existing) modules
 - Otherwise, context-dependent behaviors make programs less modular (eg. conditional branches, state design pattern)

Examples of context-dependent behaviors

- Pedestrian navigation [Kamina'11]
- Adventure game [Appeltauer'13]
- Energy-aware mobile app [Cohen'12]
- Library behaviors [Takeshita'14][Aotani'15]

Example: pedestrian navigation

[Kamina'11]

Context:

device location (indoor/outdoor)

Behavior:

type of map (floor plan/town map)

Example: adventure game [Appeltauer'13]

Contexts:

- hero's status (drunk/not)
- hero's location (town/field)
- weather (rain/not)

Behaviors:

- hero's movement
(normal/slowly/randomly)
- hero's utterance (confused/not)

Example:

energy-aware mobile app [Cohen'12]

Context:

- device status (plugged/unplugged)
- networking status (wifi/3G/disconnected)
- app phase (compute/IO-bound)

Behavior:

- rendering quality (hi-/lo-res)
- data saving freq. (always/checkpt/none)

Example: library behaviors

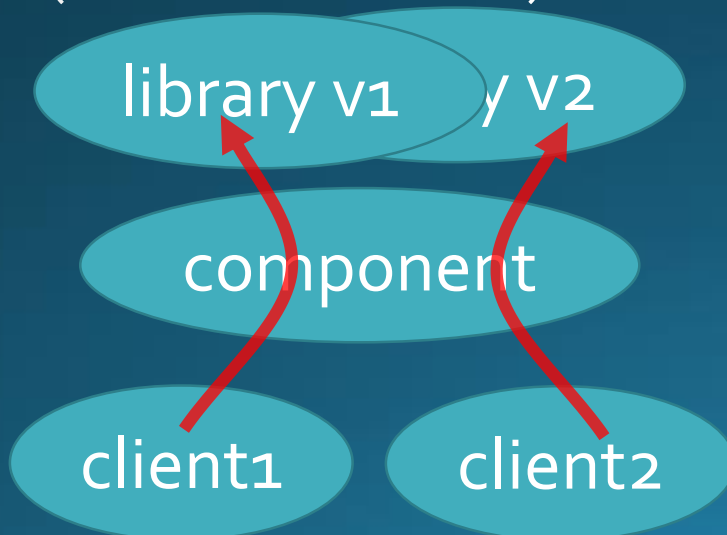
Method Shells [Takeshita'14]

Context:

- client component

Behavior:

- library implementation (different versions)



Collection library [Aotani'15]

Context:

- client phase (init/main computation)

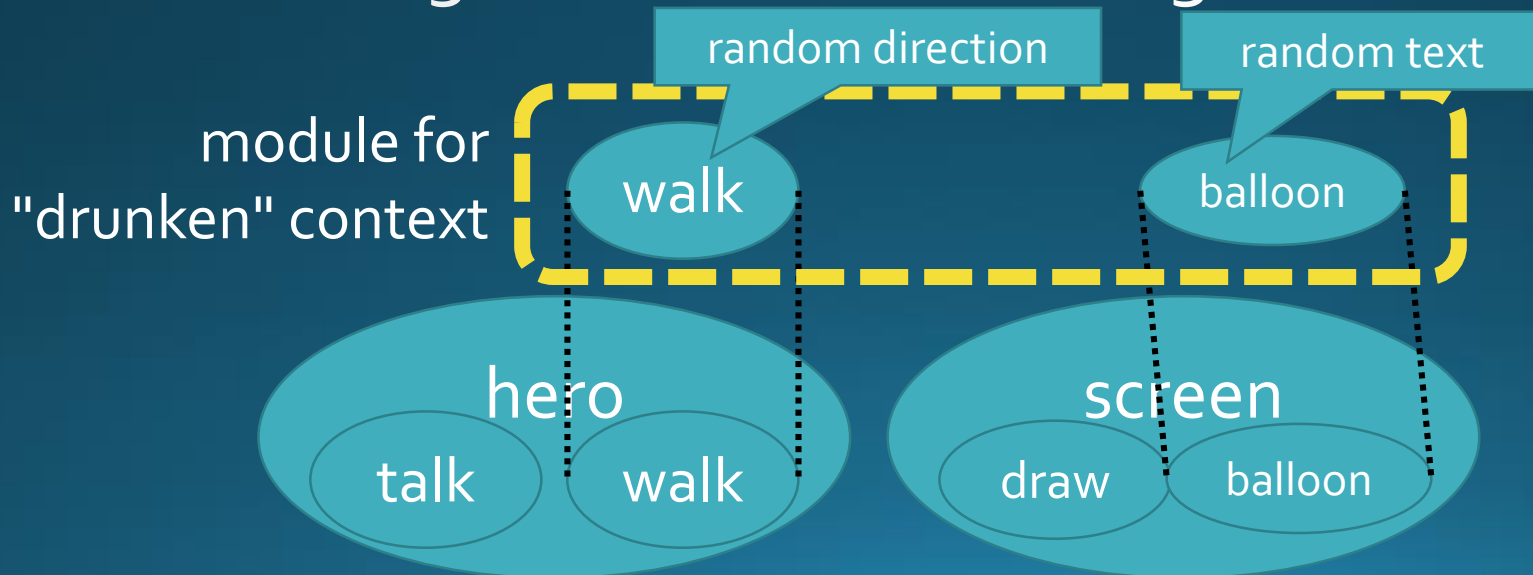
Behavior:

- thread safety (fast sequential/thread safe)

Key Idea in Context-Oriented Programming

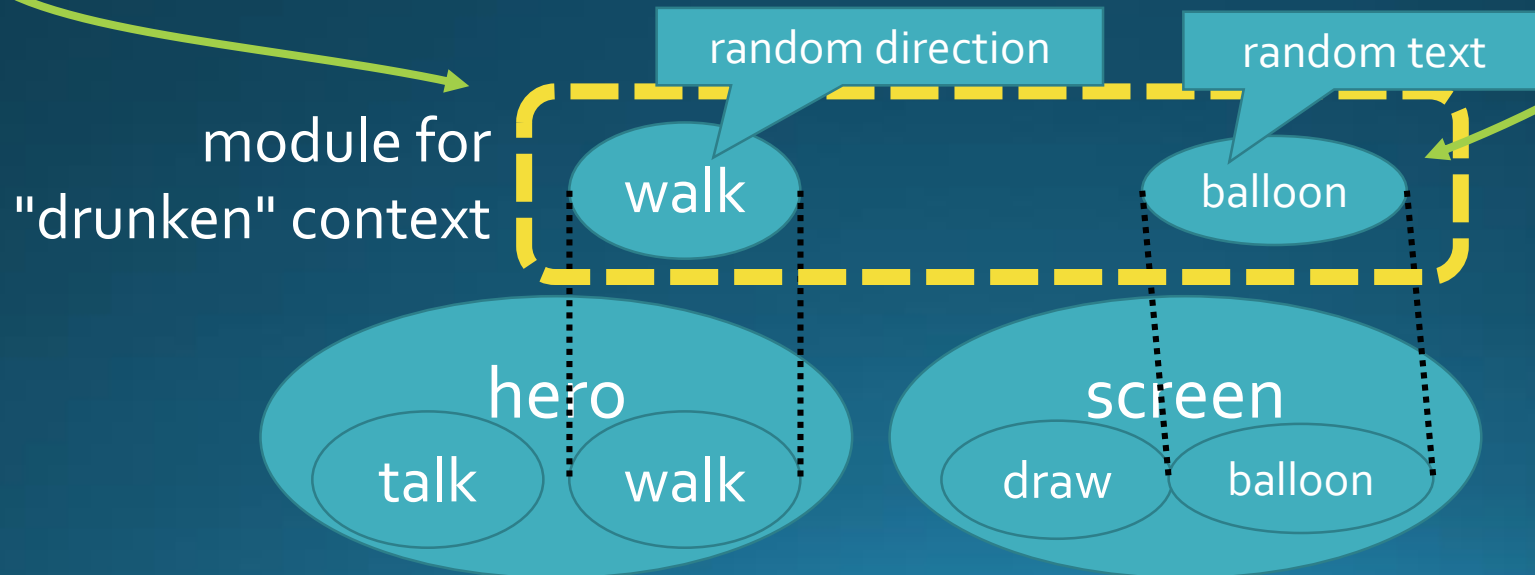
Provide a module that encapsulates context-dependent behaviors

- one module for each "context"
- overriding behaviors of existing modules



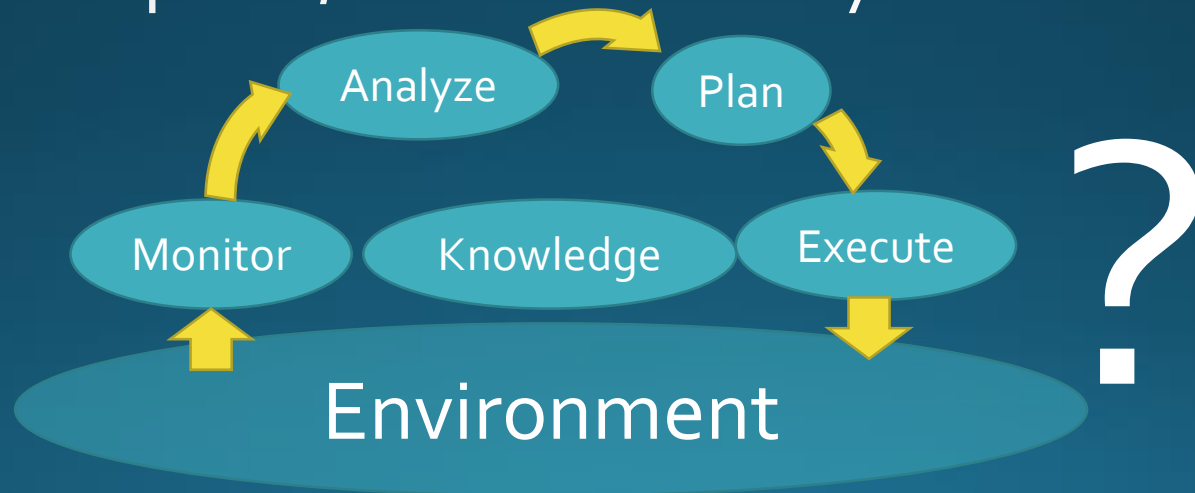
Language Constructs in COP

- Layers and partial methods
- Layer activation mechanisms
 - to turn on and off layers



What are contexts? (what to adapt?)

- "Any information which is computationally accessible" [Hirschfeld-Costanza-Nierstrasz'o8]
- Calling context (eg. with-block)
- In self-adaptive/autonomous systems:



*Kephart & Chess, "The Vision of Autonomic Computing",
IEEE Computer 2003.*

What are contexts?

Example: pedestrian
[Kamina'11]

*context exists outside
of the program*

Context:

device location (indoor/outdoor)

Behavior:

type of map (floor plan)

*outside world may have
structures*

What are contexts?

Example: adventure *context exists inside*

of the program

Contexts:

- hero's status (drunk/not)
- hero's location (town/field)
- weather (rain/not)

Behaviors:

- hero's movement
(normal/slowly/randomly)
- hero's utterance (confused/not)

*location may
be structured*

What are contexts for?

Example: *affects one character* [Appeltauer'13]

(cf. multi-player)

Contexts:

- hero's status (drunk/not)
- hero's location (town/field)
- weather (rain/not)

*affects all characters
in the region*

(only)

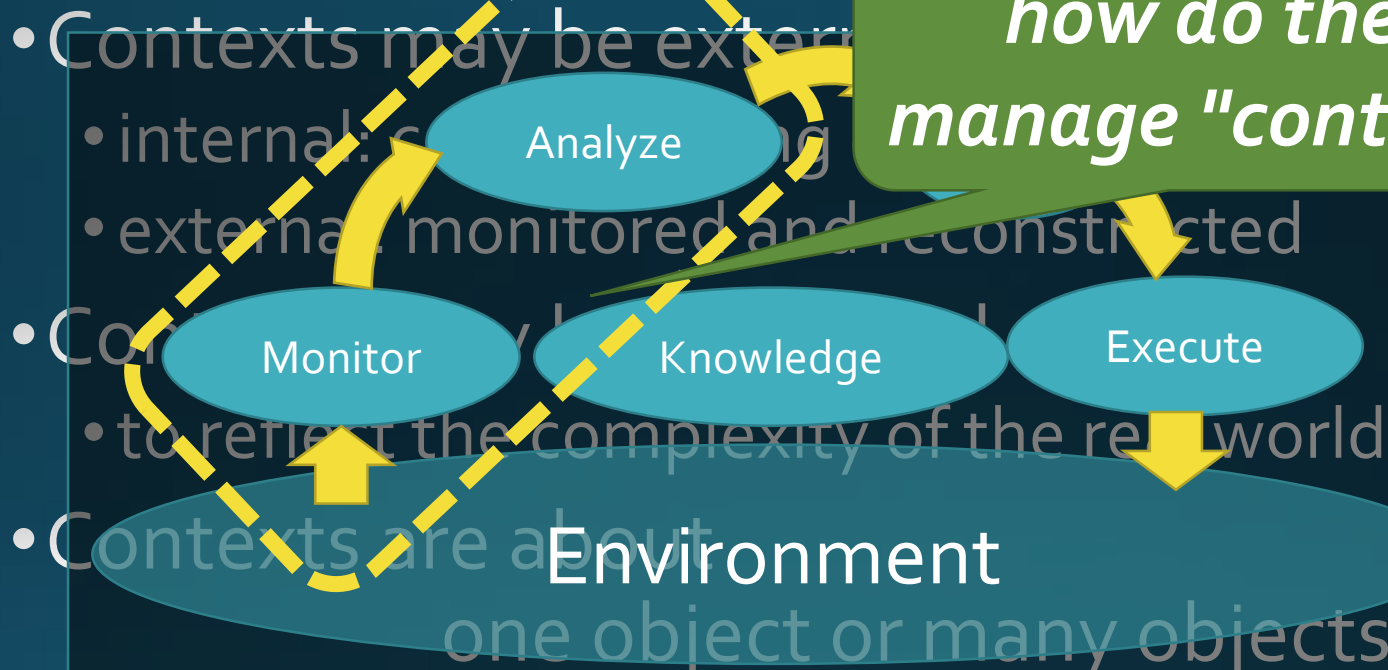
- hero's utterance (confused/not)

Varieties of contexts

- Contexts may be external or internal
 - internal: corresponding data structure
 - external: monitored and reconstructed
- Contexts may be structural
 - to reflect the complexity of the real world
- Contexts are about
 - one object or many objects

Are there uniform language constructs?

Varieties of contexts

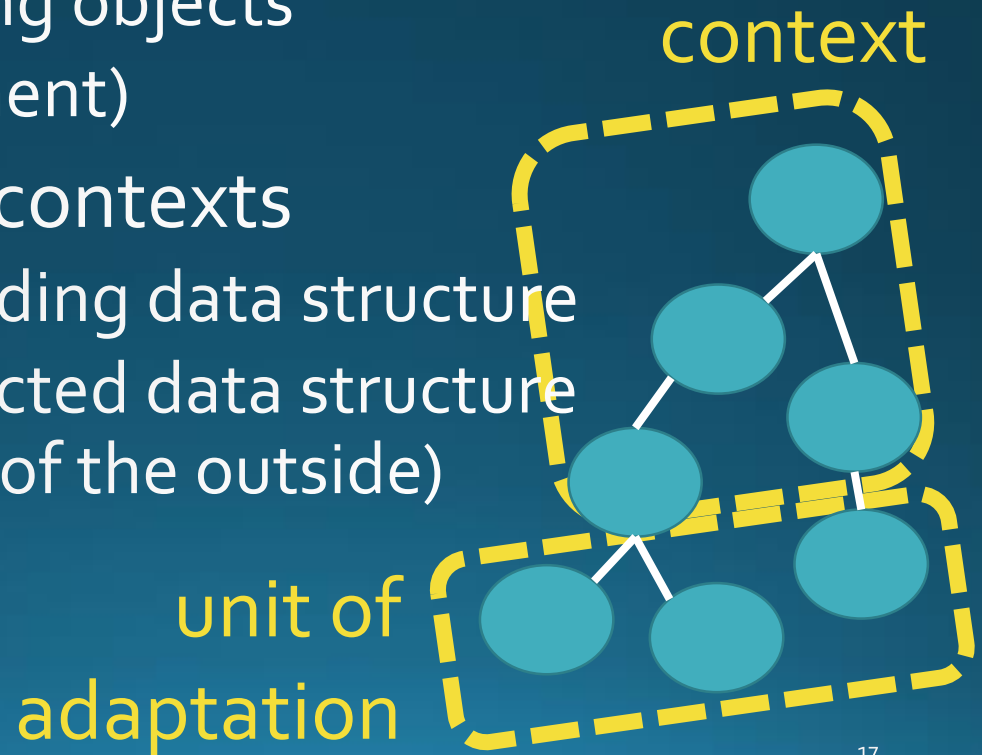


Kephart & Chess, "The Vision of Autonomic Computing", IEEE Computer 2003.

Are there uniform language constructs?

Proposal: structure-based contexts

- Context: surroundings of an adaptation unit
 - adaptation unit: a set of objects
 - context: surrounding objects (c.f. object confinement)
- Representation of contexts
 - internal: corresponding data structure
 - external: reconstructed data structure (mirrors of the outside)



Structure-based context: Language mechanisms

- Metaprogramming for context-dependent dispatching [Taeumel'14]
- More straightforward mechanism?
 - declarative predicates for expressing object relationships (eg. XPath, Association Aspects [Sakurai'04])
 - optimization techniques
 - caching results of data traversal
 - confinement/ownership/etc. types

Summary

- There are varieties of contexts in COP examples
- We look for a uniform language mechanism
- Structure-based context
 - Surrounding objects as contexts
 - Existing/reconstructed data structure
 - (still on the way)
- Contexts in self-adaptive systems?