

An Agent-Centered Perspective on Context-Aware Computing

Jamie J. Payton
8 June 2004

Christine Julien
Gruia-Catalin Roman

Mobile Computing Laboratory



Washington University in St. Louis

[Motivation]

- Interoperability has been studied through middleware development
- The emerging class of context-aware systems also require middleware support
- We must answer fundamental questions
 - What is context-awareness?
 - What is the minimal set of constructs?
- A formal treatment of context is necessary

[What Is Context?]

- Context is what defines a situation
 - Entities of possible influence
 - Identity — PARCTab, Active Badge, ...
 - Location — GUIDE, Cyberguide, ...
 - Status — Gaia, ...
 - Time — FieldNote, ...
- Dominant perspectives
 - Local
 - what I can touch
 - Immediate
 - what I can reach

[A New Design Paradigm: Context-Aware Computing]

- Expansive
 - Operational environment — potentiality
- Specific
 - Context — relevance and scope control
- Explicit
 - Specification — formal treatment
- Separable
 - Decoupling — analyzability
- Transparent
 - Context maintenance — productivity

[Building Quintessential Models]

- Shared memory
 - UNITY (Chandy and Misra)
 - programs + shared variables + union
- Mobile computing
 - Mobile UNITY
 - programs + local variables + location + coordination
- Context-aware computing
 - Context UNITY
 - programs + ?

[Basic Computational Model]

- A world of independent mobile agents
- A reference agent
 - State transitions
 - local
 - reflection of operating environment
 - Context definition
 - under agent control
 - under uniform governance

[Program Perspective]

- Mobile UNITY heritage
 - variables representing state
 - assignments representing state transitions
 - reactions extending basic assignments
- Program dynamics
 - local state transitions
 - **internal variables**
 - visible environment changes
 - **context variables**
 - local changes visible in the environment
 - **exposed variables**

[Exposed Variables Explained]

- One must expose
 - what one is willing to reveal
 - what others need
- A rich context requires much information
 - name
 - type
 - value
 - **unique variable identifier**
 - **owner agent**
 - **access control function**

[Built-in Exposed Variables]

- Support is needed for
 - context-sensitive program selection
 - access control
- Minimalist exposure
 - agent type
 - agent location
 - agent credentials

Defining Contexts:

Egocentric Context Specification

- Context-sensitive selection
- Existential quantification
- Non-deterministic assignment
- Consistency control
- Impact on the operational environment

Acquaintance List Example

- A simple policy: **relative location**

Q **uses** *location l of some agent A*
given *l is within range*
where *A is added to Q*
reactive

- Questions
 - What happens when A moves out of range?
 - How do we test for the out-of-range condition?
 - How do we handle more complex criteria?

[Token Passing Example]

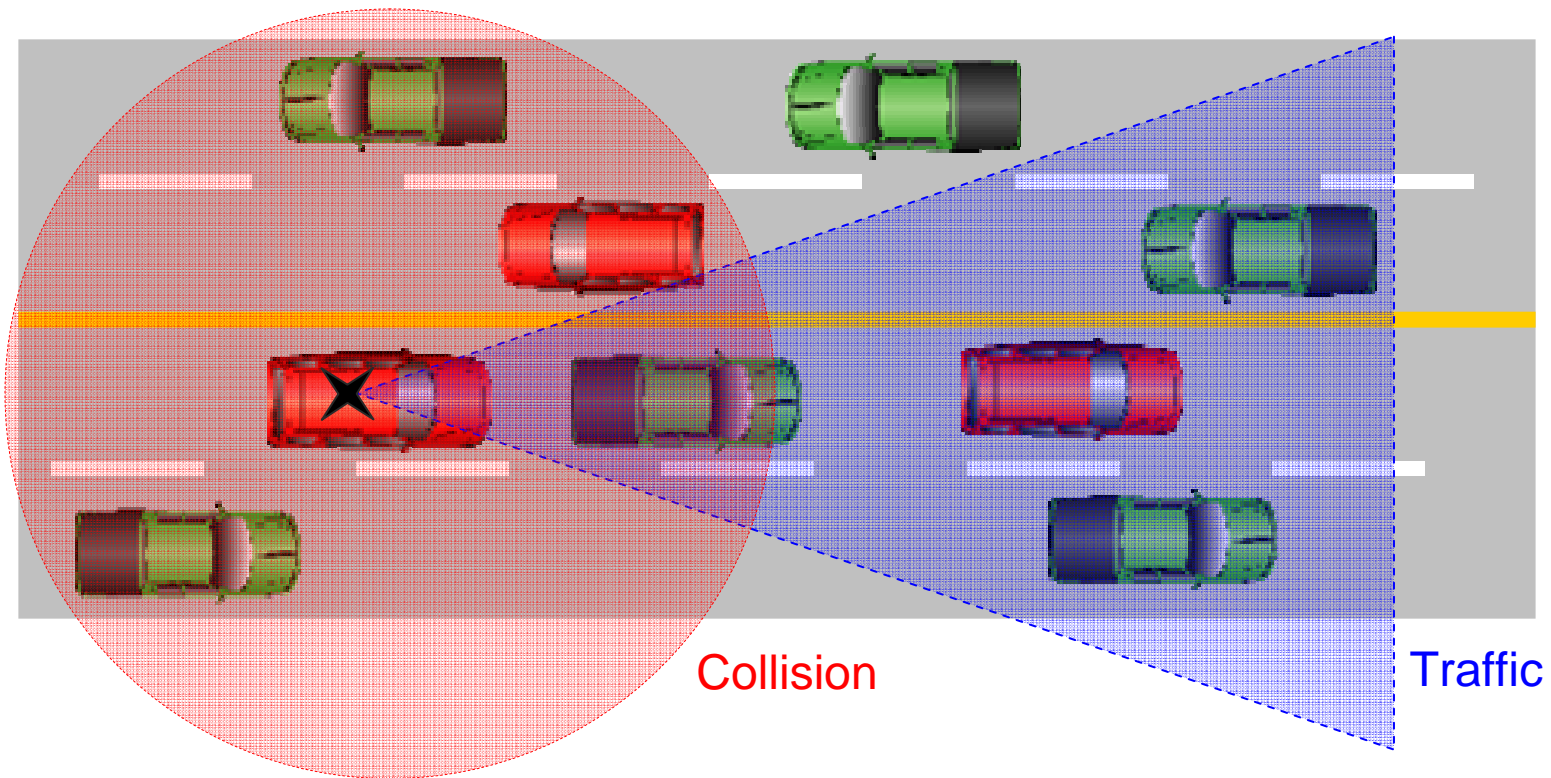
- The policy: pass tokens to any agent having a lower id

T **uses** *token set T' and location l of some agent A*
given *l is within range and A has a lower id*
where *tokens in T are added to T'*
 T is cleared of all its tokens
reactive

- Questions
 - Could this be done in Mobile UNITY?
 - the coordination mechanics of Mobile UNITY are at work here
 - context and impact specifications are highly localized
 - interactions are with unknown entities in unknown settings

Tailored Context Definitions

- EgoSpaces



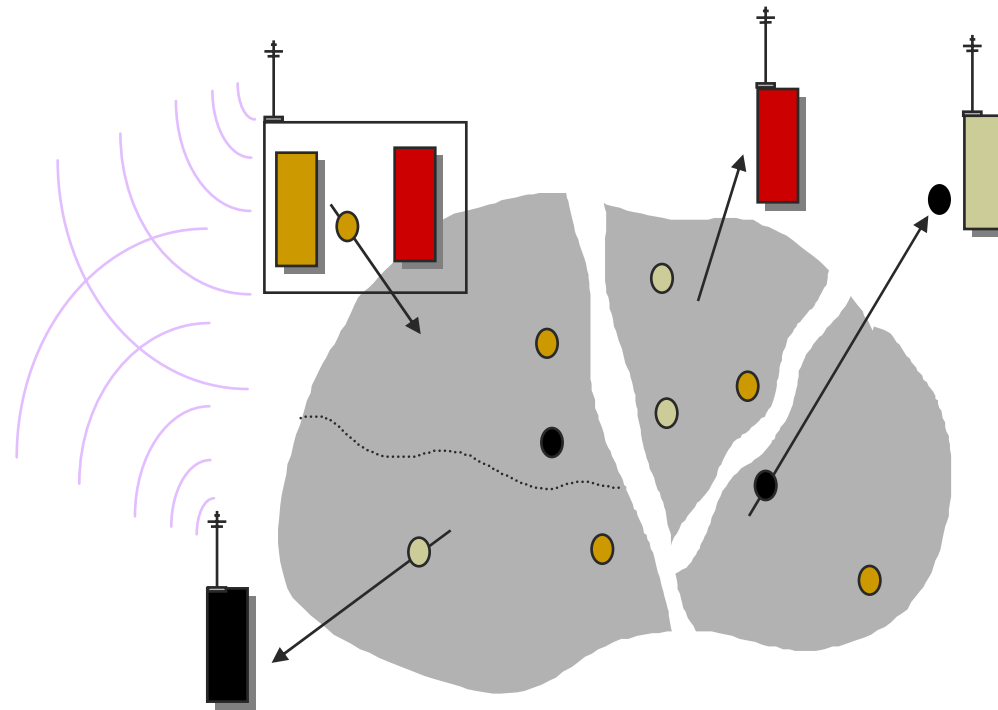
Defining Contexts:

Uniform Coordination

- Support for context-aware computing
 - comes in the form of tool kits and frameworks
 - belongs to all and to none
- Governance rules model such situations
 - are uniformly applied
 - are able to work with unknown entities
 - coordinate among multiple exposed variables

Uniform Context Definitions

- MARS, LIME



**federated tuple space
(LIME)**

Modeling Requirements Summary

- Should allow programs to define individualized contexts
 - Use non-determinism and quantification
- Should allow programs to affect their context
 - Define rules for resolving context changes
- Should cater to the varied needs of context-aware applications
 - Provide consistency options for context definition and resolution

Proof Logic

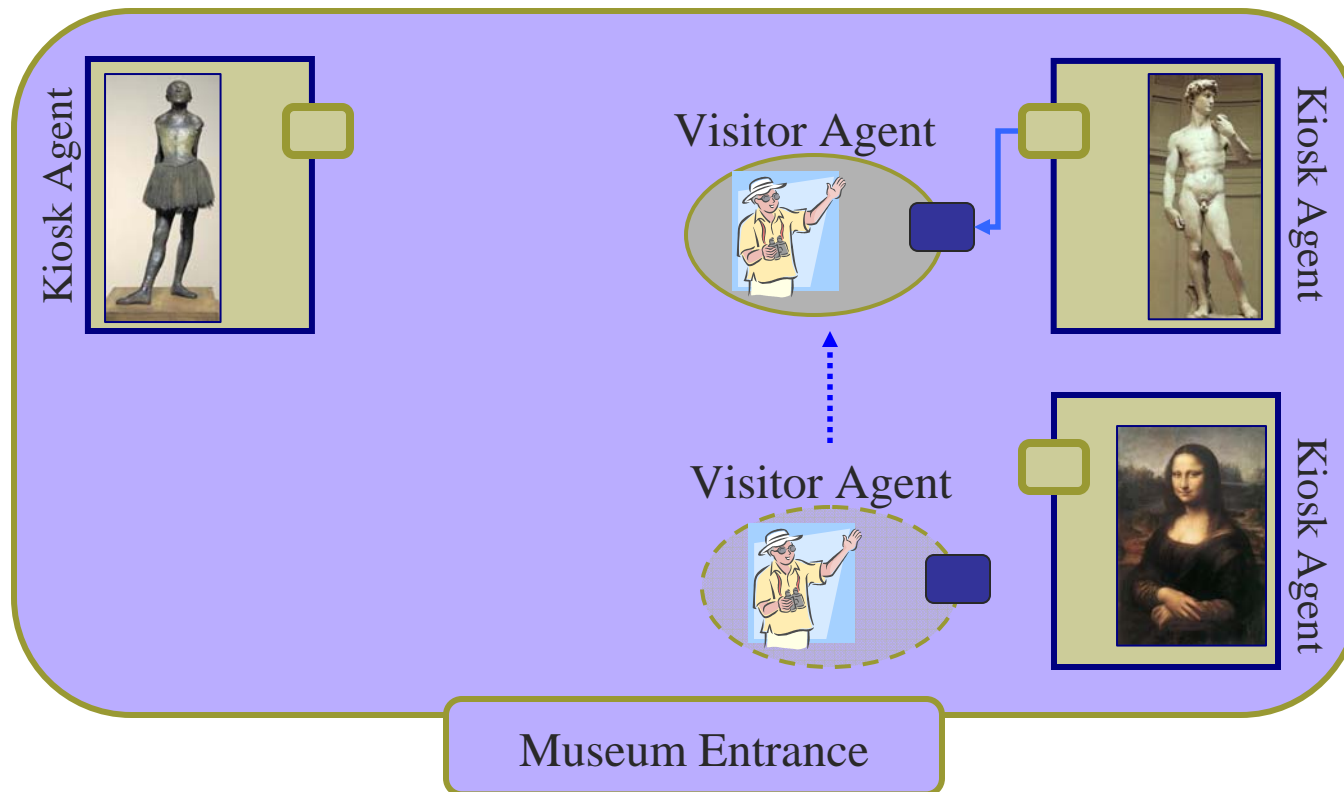
- Context UNITY constructs reduce to assignments and reactions
- Mobile UNITY proof rules apply directly
 - Hoare triples have been redefined to accommodate non-determinism and reactions

$$\frac{\{P\} s \{H\}, H \rightarrow (FP(R) \wedge Q) \text{ in } R}{\{P\} s^* \{Q\}}$$

- The UNITY proof logic continues to apply

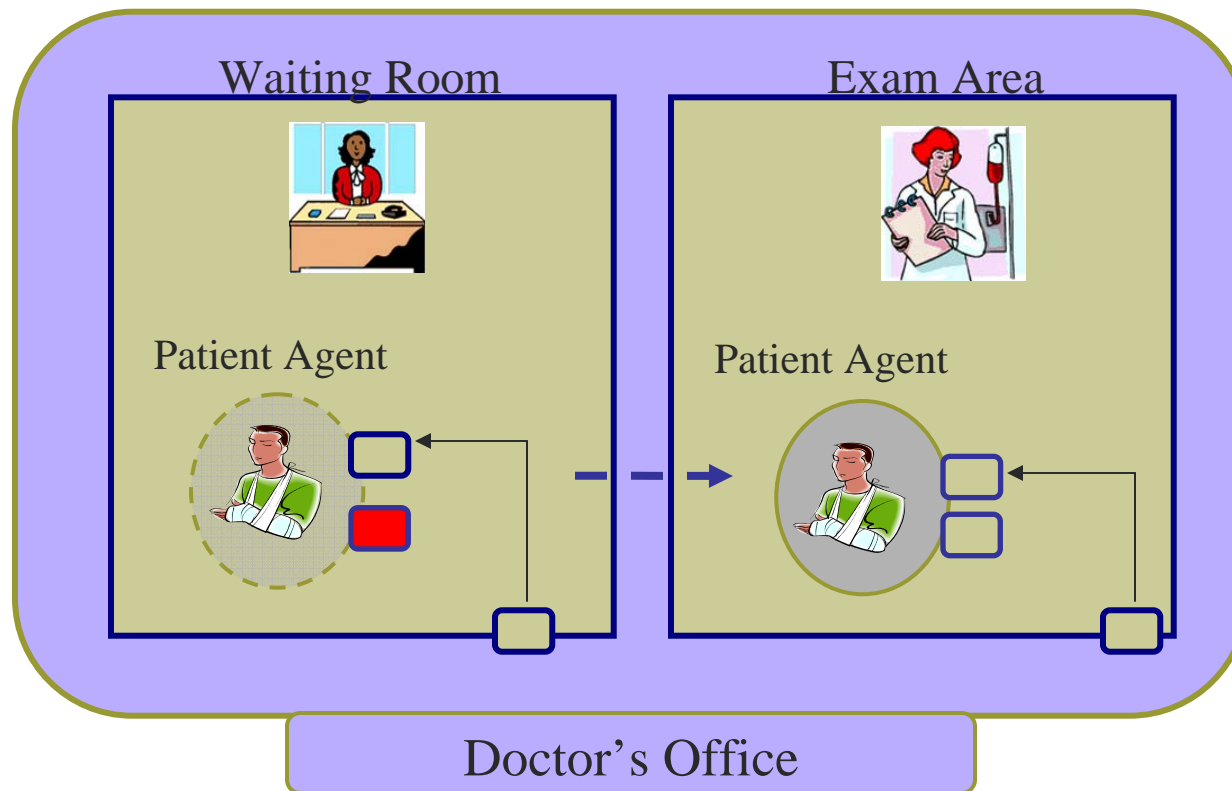
Patterns of Coordination

- Active spaces (e.g., Gaia)



Patterns of Coordination

- Context-sensitive data sharing



[Conclusions]

- Formal treatment of context-awareness is necessary to gain understanding about how to provide support for developers
- Coordination patterns are important lessons for collaborative efforts
- The Context UNITY model is specific to context-awareness, yet can be captured by more general models