Graeme Stevenson, Paddy Nixon and Simon Dobson

Systems Research Group
School of Computer Science and Informatics
UCD Dublin Belfield, Dublin 4, Ireland

http://www.ucd.ie/csi

graeme.stevenson@ucd.ie

Towards a Reliable, Wide-Area Infrastructure for Context-Based Self-Management of Communications

Context Awareness

- Context aware applications
  - Adapt behaviour based on available information
  - Display relevant information to user
- Target: pervasive computing environments
  - Large numbers of heterogeneous data sources
  - Failure of devices must be treated as commonplace
  - Developers spend time on the details of obtaining information rather than specifying how applications should that information

Example: An Active Map Application

- Obtaining context information
  - Identify the required sensors
    - *Number of floors? Number of rooms?*
  - Discover how to communicate with the sensors
    - *Different sensor technologies? Are there an existing APIs?*
- Processing context information
  - Different sensors, different data formats
  - Convert data to required format
- Maintaining the application
  - Dynamic operating environment
    - *New sensors added? Existing sensors removed?*
    - *Sensor technology upgraded?*
  - Change not limited to sensors
    - *Compliment of personnel may change*
- There is a need for middleware support...

Aim of Project

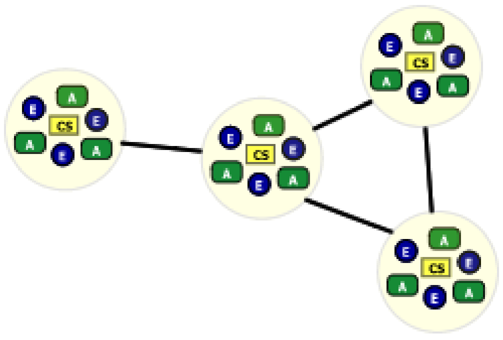
- Provide a service infrastructure which will:
  - Obtain sensor data from a variety of disparate sensor technologies
  - Convert the data to the level of abstraction desired by the applications which use it
  - Deliver it to the applications, and continue notifications when new or updated information is available
  - All with minimum user attention
- Allow developers to focus their time on using information rather than on obtaining it

Challenges

- Flexibility
  - Developers cannot anticipate at development time the physical sources of data that will be available to their applications
  - Potential gap between available information and required information
- Maintainability/Reliability
  - Sensors and other sources of information may be prone to failure
  - Self-repair if possible
- Scalability
  - Number of devices/data sources is unpredictable
  - Assumptions about scale cannot be made
- Interoperability
  - Use of standard data formats and communication protocols where possible

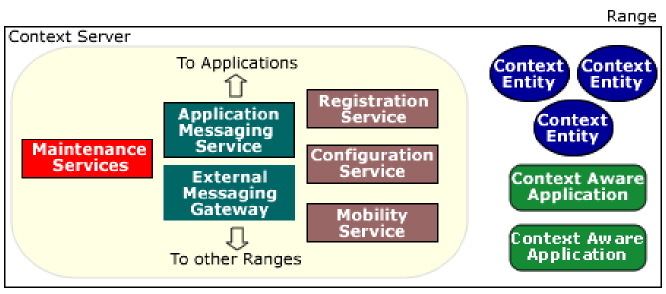
Overview of Infrastructure

- High level view: ConStruct-NET
  - Partially connected overlay of ***Ranges***
  - Ranges are functionally equivalent
  - Each Range provides services to Context Entities (CE) and Context Aware Applications (CAA)
  - Any CE or CAA which utilises the services of a Range is considered part of that Range
  - ConStruct-NET uses a P2P protocol for population management



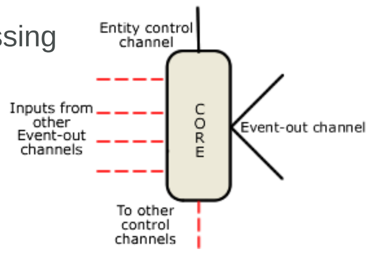
Components of a Range

- Low level view: Range
  - Context Server provides services to Context Entities and Context Aware Applications
    - *Consists of 6 components: Maintenance Services, Application Messaging Service, External Messaging Gateway, Registration Service, Configuration Service, Mobility Service.*
  - Built on top of Sun's Java System Message Queue software
    - *Implementation of the Java Message Service specifications*
    - *Provides 1-1 message queue and 1-many public/subscribe semantics*



Context Entities

- Context Entity
  - Software abstraction of a data source or processing component
- Consists of three parts:
  - Control channel
  - Event channel
  - Functional core
- Seven flavours of entity
  - Source, fusioner, aggregator, transformer, generaliser, filter, and merger
- Entity Profile
  - XML formatted meta-data which describes the properties of the information supplied by the entity
  - Used to locate resources in the network


Entity control channel
Inputs from other Event-out channels
CORE
Event-out channel
To other control channels

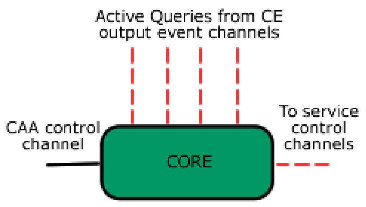
Entity Profile

- Entity profiles contain four main parts
  - A classification, a location, a description of the output generated by the entity, and descriptions of any inputs the entity requires

```
<?xml version="1.0" encoding="UTF-8"?>
<entityProfile>
   <entityName> Celsius Temp Publisher </entityName>
   <entityClassification> source </entityClassification>
   <location> ComputerScienceBuilding/Level0/RoomA007 </location>
   <outputEvent>
      <ontology> construct.ontology.temperature </ontology>
      <staticFields>
         <field>
            <fieldName> unit </fieldName>
            <fieldValue> celsius </fieldValue>
         </field>
      </staticFields>
   </outputEvent>
   <inputEventSet></inputEventSet>
</entityProfile>
```

Context Aware Applications

- Context Aware Application
  - Similar to a Context Entity but does not publish events
  - Has an associated query file which describes its data requirements

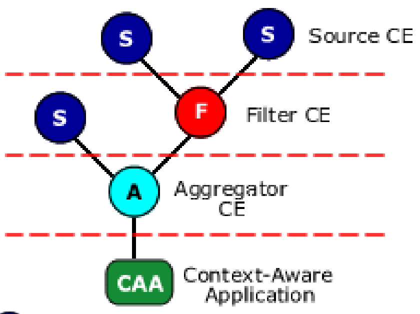

Active Queries from CE output event channels
CAA control channel
CORE
To service control channels

- Application query file
  - A queryID, a location, a description of the input events required by the application

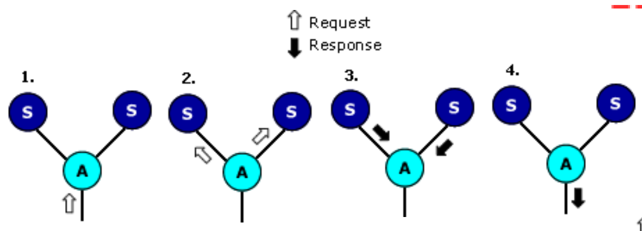
```
<?xml version="1.0" encoding="UTF-8"?>
<queries>
    <query>
        <queryID> LevelOnePeople </queryID>
        <location> REGEX[ComputerScienceBuilding/L1/] </location>
        <eventType> construct.ontology.location </eventType>
        <staticFields>
            <objectType> person </objectType>
        </staticFields>
    </query>
</queries>
```

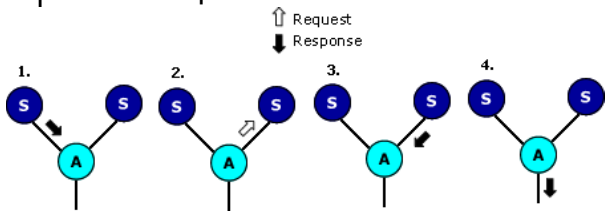
Configurations and Interactions

- Configurations
  - Messages passed up the hierarchy
  - Events passed down the hierarchy



- Interaction Model



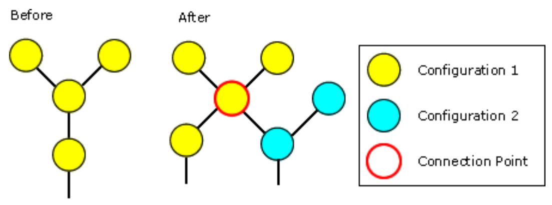


Query Resolution

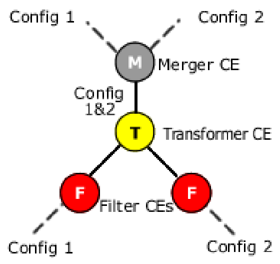
- Query Resolution Process using Automatic Path Creation
  - Identify Context Entities that match the desired location and output event type requested by the application
  - Compare the attribute-value pairs describing the output supplied by each candidate entity against the application's requirements and group entities into one of four categories: no match, partial match, exact match, and over match
  - If exact matches exist, examine their input requirements in turn, and determine if they can be satisfied (using this procedure)
  - If not, examine the input requirements for any partial matches in a similar manner. If a complete configuration can be formed, a filter is automatically generated and configured to bridge the gap between the output of the configuration and the requirement of the application
  - The final option is to evaluate the group of entities in the over match category. The results of all successfully evaluated configurations can then be merged together to provide the application with the best possible match available

Resource Reuse

- Reuse of Event Streams



- Reuse of Entities
  - Mergers and filters automatically generated

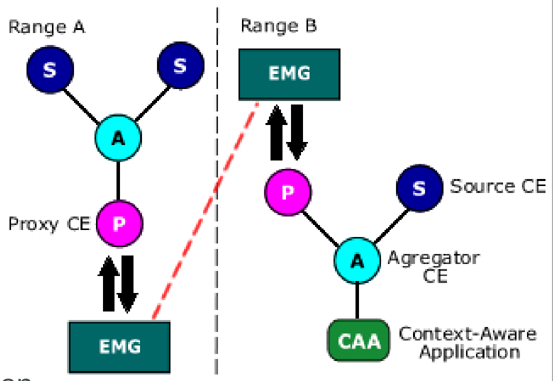


Automated Runtime Maintenance

- Runtime maintenance of configurations
  - Periodic passing of Ping/Pong style messages to detect Entity/Application activity
  - Services try to repair broken ‘branches’ in a configuration
    - *Applications remain unaware unless the end point of a configuration is broken*
  - Periodic revaluations make use of additions to the resource pool
  - Planned extension to utilise QoS information in order to improve entity selection when multiple choices exist

Inter-Range communication

- Communicating across Ranges
  - External Messaging Gateway sets up a Pastry node to join ConStruct-NET
  - Configuration Service can route messages to the Configuration Services in remote Ranges
  - Proxy entities serve as a local representation of a remote resource
- Application Mobility
  - Mobility Service caches application events if requested
  - Responds to maintenance messages on behalf of an application

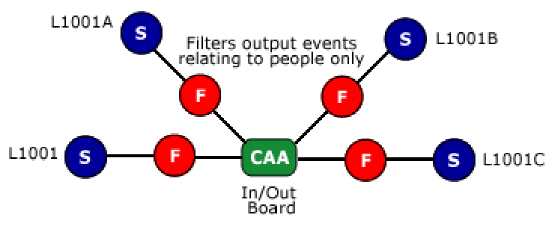


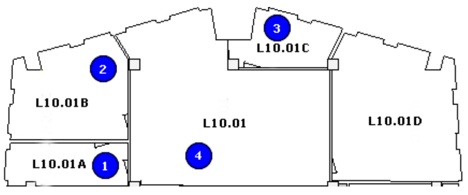
Programming Model

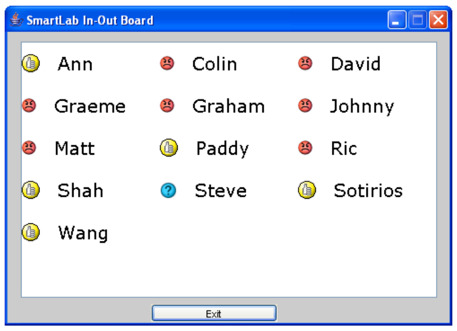
- Abstract classes perform all interactions with the Context Server services
- Simple API provided for developing new Context Entities and Context Aware Applications
  - subscribe(), unsubscribe(), addQuery(), removeQuery(), getLastEvent() etc…
- Developers need only provide an implementation for methods of interest
  - evaluate(), eventHook(), queryResponseHook(), etc..

Applications (1)

- In/Out Board
  - Uses location information to display status of lab occupants



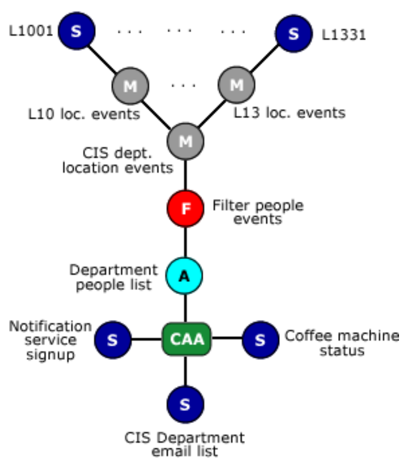


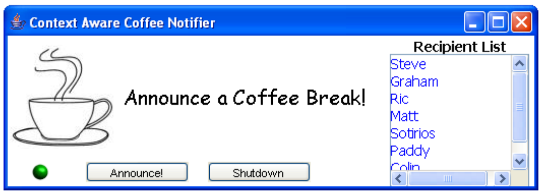


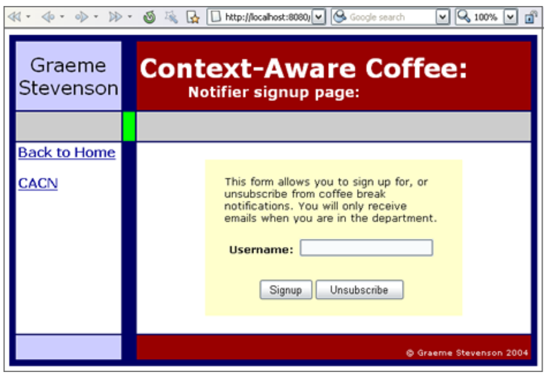
Ann	Colin	David
Graeme	Graham	Johnny
Matt	Paddy	Ric
Shah	Steve	Sotirios
Wang		

Applications (2)

- Coffee Break Notifier
  - Announce coffee break only to interested parties who may be able to attend.







Context-Aware Coffee:
Notifier signup page:

This form allows you to sign up for, or unsubscribe from coffee break notifications. You will only receive emails when you are in the department.

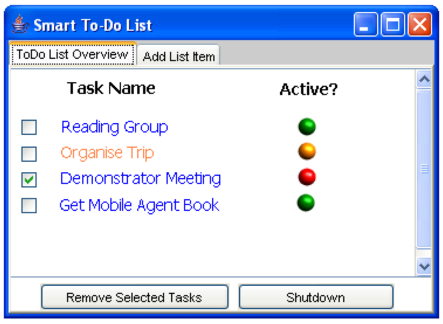
Username:

Signup Unsubscribe

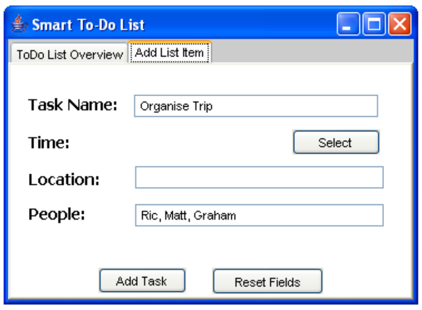
© Graeme Stevenson 2004

Applications (3)

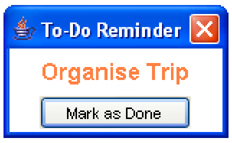
- Smart To-Do List
  - Reuses location sensors in previous applications
  - Application generates new queries at runtime using addQuery() and removeQuery() methods



	Task Name	Active?
☐	Reading Group	●
☐	Organise Trip	●
☑	Demonstrator Meeting	●
☐	Get Mobile Agent Book	●



Task Name:	Organise Trip
Time:	
Location:	
People:	Ric, Matt, Graham



Organise Trip

Towards Context Provision in an AC Environment

- Focus of work up to this point has been on the decomposition of applications and minimising developer effort
  - Entities provide reusable building-blocks
  - Loose coupling between applications and the data they require
  - Specification of data by its properties rather than physical location
- Self-organising and self-maintaining at the ConStruct-Net and configuration levels, but does not address all AC concerns
  - Context Servers are single points of failure
  - Design decision made to lessen the burden on individual participants
  - However, it is a fair criticism of the work from an AC perspective

Towards Decentralisation

- Future target: Decentralisation of the infrastructure
  - Many issues to address, such as resource discovery, configuration maintenance etc.
- Currently looking at load balancing
  - The demand placed on entities is not equal
  - The capabilities of devices are not equal
  - Synthesis of data sources requires runtime generation of new Context Entities
  - Applying work from *Globspace* project
    - *Achieves decentralised load balancing by creating a number of nodes in a virtual network space in proportion to host 'fitness'*
    - *Entities can measure current message frequency and entity count on host*
    - *Migration to a randomly selected node results in arrival at a more suitable note with high probability*

Summary

- Goal was to build an infrastructure that would allow developers to focus on the use of context information rather than the details of obtaining it
  - Applications describe the properties of the context information they require
  - Services use this information to generate a configuration to connect applications to the data they require at the correct level of abstraction
  - Configurations are automatically maintained to account for new data sources and entity failures
  - Multiple deployments of the infrastructure services connect to form an overlay which provides location of remote sources of context, aiding scalability
- Successful in meeting goal, but does not yet address all AC concerns
  - The Context Server remains a single point of failure at the Range level
- The challenge of providing a fully decentralised infrastructure is the next step

The End