Extensibility as a Collaboration Enabler: A Case Study for Group-Context-Aware Mobile Applications

Marc Novakouski Grace A. Lewis Enrique Sánchez

Software Engineering Institute Research, Technology and Systems Solutions (RTSS) Program Advanced Mobile Systems (AMS) Initiative May 9, 2012

Software Engineering Institute Carnegie Mellon

© 2012 Carnegie Mellon University

Report Documentation Page

Form Approved OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 09 MAY 2012	2. REPORT TYPE	3. DATES COVERED 00-00-2012 to 00-00-2012	
4. TITLE AND SUBTITLE Extensibility as a Collaboration Enabler: A Case Study for Group-Context-Aware Mobile Applications		5a. CONTRACT NUMBER	
		5b. GRANT NUMBER	
		5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)		5d. PROJECT NUMBER	
		5e. TASK NUMBER	
		5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)	
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)	

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

SEI Architecture Technology User Network Conference (SATURN 2012), May 7-11, 2012, St Petersburg, FL.

14. ABSTRACT

Context-aware mobile applications can sense and respond to changes in environment or context. An example is a location-based application that uses GPS coordinates to define the information displayed to mobile users based on their current location. Our work focuses on the integration of an individual's context with that of nearby individuals operating as part of a group or unit, such as in the military or first-responder situations. This integrated context can then be used to enhance the precision of information provided to users and give a more complete picture of the status of a mission. Given the early stages of the research process when there are many unknowns, we defined extensibility as the main architectural driver. This work has provided the ability to leverage the architecture to support collaboration. By identifying extensibility scenarios early in the design process, we were able to construct an architecture that supports multi-organizational collaboration to construct and evaluate different pieces of the architecture: context data models, context sources such as sensors, context reasoning engines and rules, and context visualization activities. This has allowed us to reach out to researchers from multiple universities and industry, resulting in synergistic research and development furthering the goals of all participants.

ABSTRACT OF PAGES	RESPONSIBLE PERSON
16. SECURITY CLASSIFICATION OF: 17. LIMITATION OF 18. NUMBER	19a. NAME OF

Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std Z39-18 Copyright 2012 Carnegie Mellon University.

This material is based upon work supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

*These restrictions do not apply to U.S. government entities.



Background Architecture Drivers and Scenarios Architecture Decisions Extensibility as a Collaboration Enabler — Results Conclusions



Background Architecture Drivers and Scenarios Architecture Decisions Extensibility as a Collaboration Enabler — Results Conclusions



Carnegie Mellon SATURN 2012 May 9, 2012

Group-Context-Aware Mobile Applications ₁

Context-aware mobile applications are capable of sensing and responding to changes in their environment or context

Group-context-aware mobile applications integrate the individual's context with that of nearby individuals operating as part of a group or unit, such as in the military or first responder situations

Integrated context is used to enhance the precision of information provided to users as well as a more complete picture of the status of a mission

 Goal is to produce a capability that can sense as much of the emerging context as possible and apply that context to filter data such that only the most relevant information is displayed





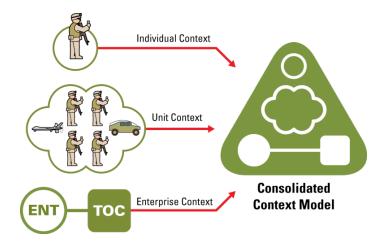
ftware Engineering Institute Carnegie Mellon

SATURN 2012 May 9, 2012 © 2012 Carnegie Mellon University

Group-Context-Aware Mobile Applications ₂

Desired characteristics of the solution include

- Capturing context information on a handheld device in a non-intrusive manner
- Extending the sources of contextual information beyond location and time
- Storing context information and disseminating this information to peers
- Capturing and using context information efficiently without imposing an unreasonable burden on handheld device resources
- Integrating local and group context information and only displaying information that is of relevance to the individual and mission according to pre-defined rules





Software Engineering Institute Carnegie Mellon

SATURN 2012 May 9, 2012 © 2012 Carnegie Mellon University

Motivation

One of the more interesting results of this work has been the ability to leverage the architecture to support collaboration

By identifying extensibility scenarios early on in the design process, we were able to construct an architecture that supports multi-organizational collaboration to construct and evaluate different pieces of the architecture

- context data models
- context data storage
- context sensors
- context reasoning engines and rules

Software Engineering Institute

context views

This has allowed us to reach out to researchers from multiple universities and industry, resulting in synergistic research and development, furthering the goals of all participants

Loading

Background Architecture Drivers and Scenarios Architecture Decisions Extensibility as a Collaboration Enabler — Results Conclusions



Business and Architectural Drivers

Business Drivers

- Opportunistic integration of new technology
- Ease of integration with components produced by collaborators
- Applicability of architecture to different edge-enabled applications

To meet business drivers we defined **extensibility** as the main architectural driver.



Extensibility Scenarios

#	Name	Attribute Concern	
1	Add a New Sensor	Separation of Concerns	
2	Add a New Sensor	Modifiability	
3	Add a New Communication Mechanism	Separation of Concerns	
4	Add a New Communication Mechanism	Modifiability	
5	Add a New Context Event / Action	Separation of Concerns	
6	Add a New Context Event / Action	Modifiability	
7	Add a New Context View	Separation of Concerns	
8	Add a New Context View	Modifiability	



Scenario 3: Add a New Communication Mechanism

Scenario	Add a New Communication Mechanism		
Attribute	Extensibility		
Attribute Concern	Separation of Concerns		
Scenario	Stimulus	Developer	
Refinement	Stimulus Source	Developer identifies a communication mechanism that can be used to share context data with other mobile devices	
	Environment	Developer is sufficiently comfortable with application to make changes in a reasonable amount of time	
	Artifact	Communications Manager of the context-aware system	
	Response	Communications Manager is changed to implement message passing using the new communication mechanism	
	Response Measure	Aside from communication-mechanism-specific code, only the Communications Manager is changed to accommodate the new communications mechanism.	



Scenario 5: Add a New Context Event / Action

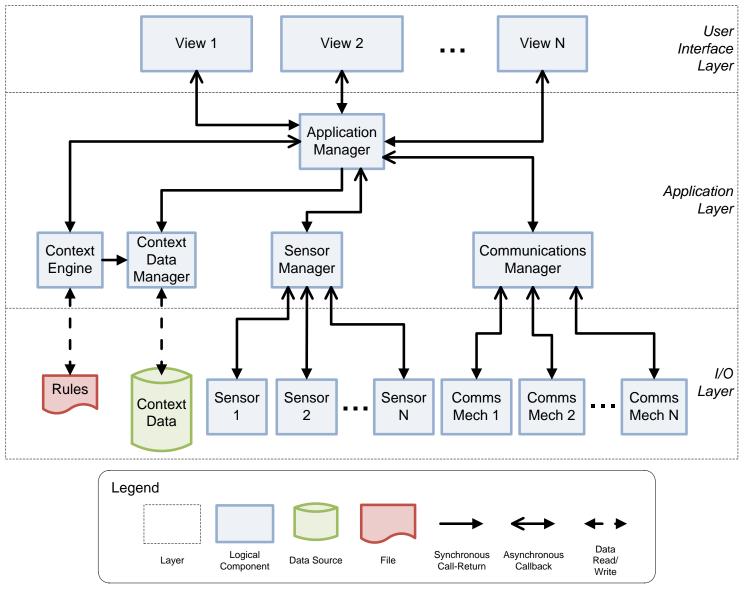
Scenario	Add a New Context Event/Action		
Attribute	Extensibility		
Attribute Concern	Separation of Concerns		
Scenario	Stimulus	Developer	
Refinement	Stimulus Source	Developer identifies a new event that can be detected by examination of context data	
	Environment	Developer is sufficiently comfortable with application to make changes in a reasonable amount of time	
	Artifact	Context Engine of the context-aware system	
	Response	Context Engine is changed to detect the conditions for the event and generate a new action when it is detected	
	Response Measure	Only the Context Engine is changed to allow for detection of events and generation of actions	



Background Architecture Drivers and Scenarios Architecture Decisions (Extensibility as a Collaboration Enabler — Results Conclusions



High-Level Reference Architecture



Software Engineering Institute

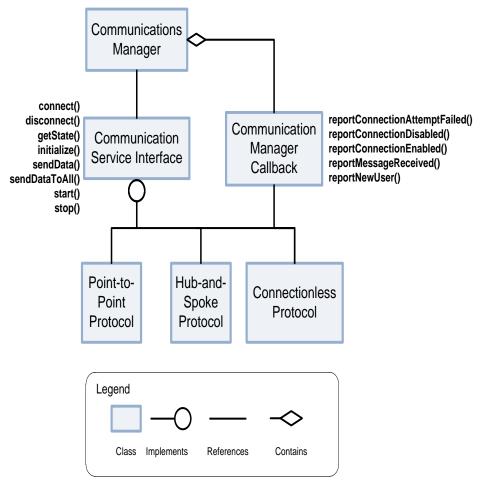
Architectural Decision 1: Communications Interface

Challenge: Integration of very different communication mechanisms

- Different protocols support different use cases
- Target hardware is unknown
- Need to adapt to target network capability

Solution

- Common service interface provides generic communication methods and callbacks that individual protocols can adapt as necessary
- Allows any sequence of communication events to account for differences in protocols





Software Engineering Institute | CarnegieMellon

SATURN 2012 May 9, 2012 © 2012 Carnegie Mellon University

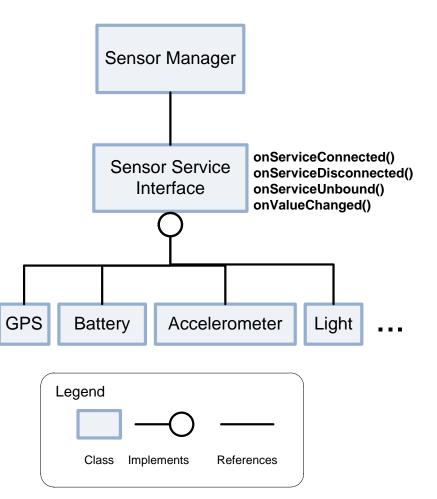
Architectural Decision 2: Sensor Interface

Challenges

- Integration of any current or future available sensor
- Control of sample rate and change threshold

Solution

 Common sensor interface provides generic communication methods that individual sensor implementations can adapt to as appropriate





Software Engineering Institute Carnegie Mellon

SATURN 2012 May 9, 2012 © 2012 Carnegie Mellon University

Problem

Peer review of the architecture raised the issue that a single thread would cause high-frequency sensors to overwhelm the application

Simple experimentation demonstrated that this was indeed a problem Solution

- Sensors implemented as Android Services (processes separate from the application)
- Communication via IPC to insulate application from high poll rate impact

Tradeoff

Higher complexity in sensor implementation although interface hides as much as possible

GPS Service	Gyroscope Service	Battery Service	Accelerometer Service	Sensor Manager
onValueChanged() [rate = 20ms]				
		onValueChar	nged() [rate = $\frac{1}{2}$ 0ms]	
		on	ValueChanged() [rate = 100	Dms]
			onValueChanged	() [rate = 20 ms]



Architectural Decision 3: Context Model "At the Center"

Challenges

- Easy creation of rules based on contextual data captured via sensors or user input
- Standardized rule processing

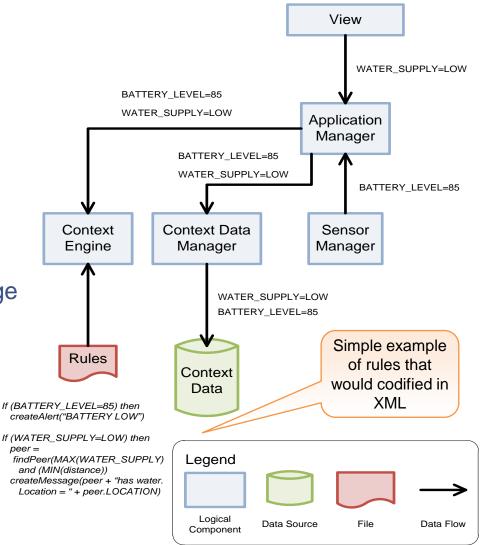
Solutions

- Generic and extensible context model that can handle a wide range of situations, environments, data
- Standardized rule set read by application from XML file

Tradeoff

 Both sensors and views have to know the context model element that they are affecting — strong coupling

Software Engineering Institute



Carnegie Mellon SATURN 2012 May 9, 2012

May 9, 2012 © 2012 Carnegie Mellon University

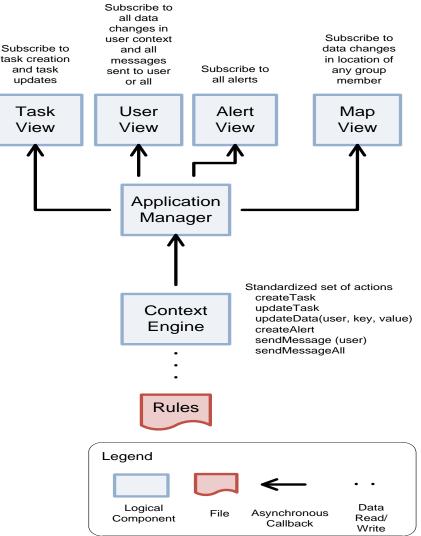
Architectural Decision 4: Standardized Messaging

Challenge

 Easy creation of views that can capture and/or display context data

Solution

- Publish/subscribe interface
 - Standardized set of actions that can be created by the context engine as the result of fired rules
 - Application manager publishes actions created by context engine as standardized events
 - Views subscribe to events

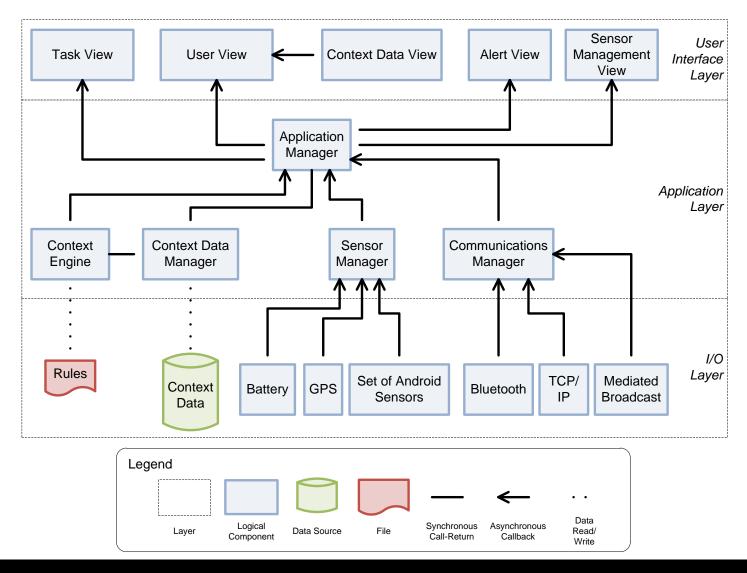




Software Engineering Institute | CarnegieMellon

SATURN 2012 May 9, 2012 © 2012 Carnegie Mellon University

First Responder Application Architecture



Software Engineering Institute

Background Architecture Drivers and Scenarios Architecture Decisions Extensibility as a Collaboration Enabler — Results Conclusions



Results ₁

The extensible architecture enables productive collaboration

- Sensor and communication service interface enable 3rd parties to contribute new/novel sensor and protocol implementations
- Standardized rule set approach allows enables adaptation to different context data models
- Standardized messaging enables easy integration of new context data views



Results ₂

Collaborators at GMU were able to modify their unique communication protocol to interface with application architecture in just a few weeks

Collaborators are working on developing a group context data model, unconstrained by implementation details and without affecting our progress in the meantime

Collaborators within SEI planning to integrate related projects for QoS management, code offloading, and end-user programming with no foreseen complications



Software Engineering Institute Carnegie Mellon

Background Architecture Drivers and Scenarios Architecture Decisions Extensibility as a Collaboration Enabler — Results Conclusions



Conclusions

Extensibility as an architecture driver enables productive collaborative research and development

Scenario-driven architecture design along with peer architecture evaluation is useful even for small projects

- Concrete definition of quality attribute requirements
- Early identification of risks and tradeoffs



Contact Information

Marc Novakouski

Research, Technology and Systems Solutions (RTSS) Program Advanced Mobile Systems (AMS) Initiative

Software Engineering Institute 4500 Fifth Avenue Pittsburgh, PA 15213-2612 USA

Phone: +1 412-268-4274 Email: <u>novakom@sei.cmu.edu</u> WWW: <u>http://www.sei.cmu.edu/staff/novakom</u>





Software Engineering Institute Carnegie Mellon