

It Takes an Ecosystem
Gary Chastek
John D. McGregor

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Gary Chastek
April 25, 2012



Introduction

During the second half of 2011, the Software Engineering Institute collaborated with Project Manager Force XXI Battle Command Brigade and Below (PM-FBCB2) to create an ecosystem model for the Mounted Computing Environment (MCE).

This presentation describes that preliminary effort:

- What were the benefits?
- What was the context for the work?
- What is an Ecosystem?
- What is STREAM?
- How did it all work in the MCE?



Ecosystem Modeling and Analysis Benefits

Improved understanding of complex supply (acquisition) and value chains

Prediction of the effects of strategic decisions on the large and complex software development efforts

Improved software development practices such as

- building a business case
- developing a CONOPS
- scoping
- software architecture definition



Context for the Work

The MCE ecosystem modeling work was in the context of the U.S. Army's Common Operating Environment (COE).

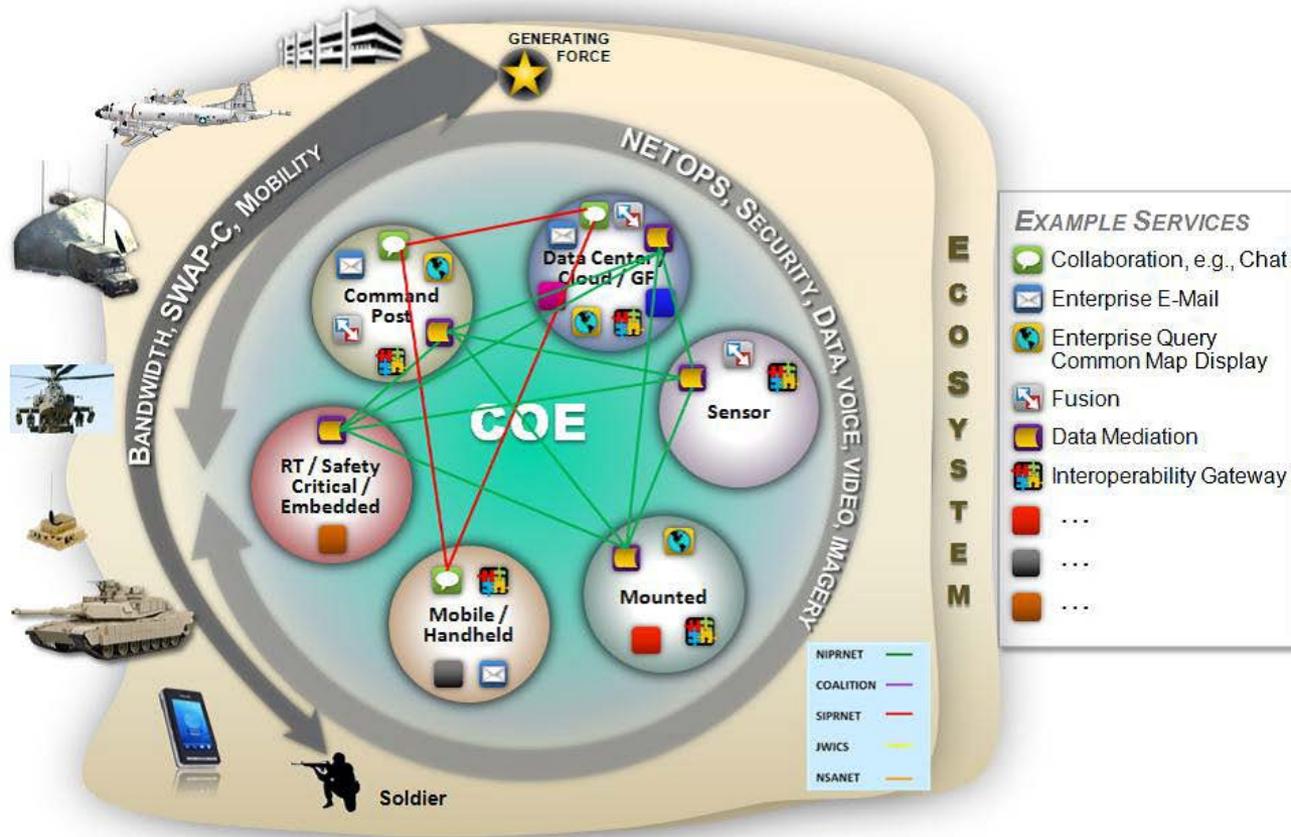
“The Common Operating Environment is an approved set of computing technologies and standards that enable *secure* and *interoperable applications* to be *rapidly developed* and *executed* across a variety of Computing Environments.”

Deputy Chief of Staff, G-3/5/7, 24 May 2010 [11]

The COE Implementation Plan “highlights critical enablers to COE success including the establishment of a software Ecosystem and enterprise business strategies that are necessary for the Army to *leverage industry best practices* and rapidly develop secure and interoperable applications that satisfy operational requirements.” [11]



Common Operating Environment



What is the Mounted Computing Environment?

The Mounted Computing Environment (MCE) provides

- operating and runtime systems,
- native and common applications and services (e.g., awareness, execution functions, integration of local sensors),
- software development kits (SDK), and
- standards and technologies to implement *mission command*.

MCE must interoperate with the other COE Computing Environments

- Handheld
- Command Post
- Sensor
- RT/Safety Critical/Embedded
- Data Center Cloud/GF (Generating Force)



What is an Ecosystem? - 1

Every software development organization is involved in relationships with *suppliers, buyers, collaborators, and competitors* whose actions impact the operation of the organization.

These intertwining relationships form a *network* that includes *symbiotic and predatory relationships*.

These interacting entities form an ecosystem similar in many ways to a natural ecosystem.

Organizational management practices such as developing a *business case, risk management planning, market analysis, and technology forecasting* use information about the ecosystem to guide the strategic decisions of the organization.



What is an Ecosystem? - 2

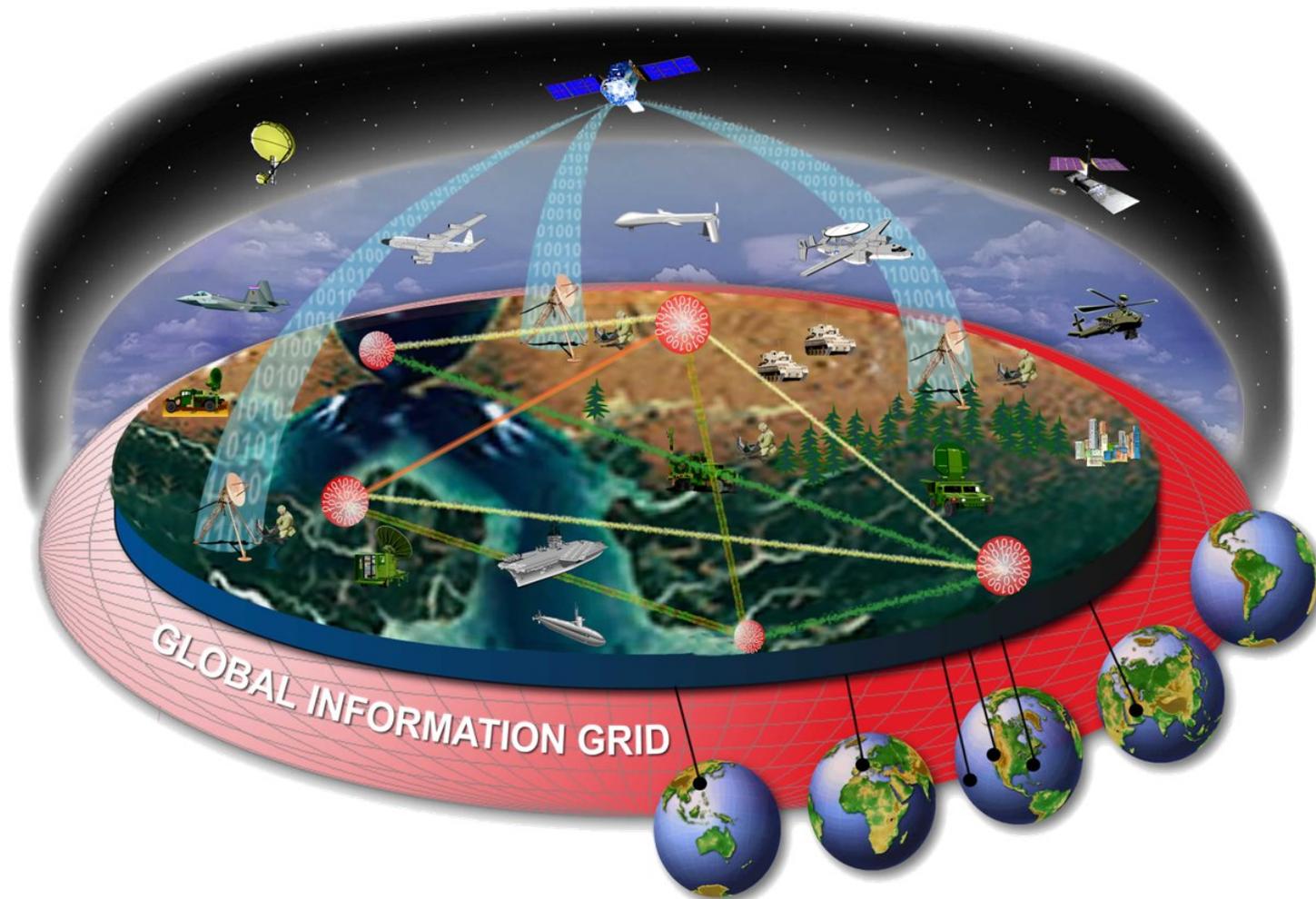
While there is no one globally accepted definition, with few exceptions, the common theme is the *blurring of boundaries of systems and organizations* that serve to *enable interconnected communication and execution of critical capabilities*.

An ecosystem is “A set of businesses functioning as a unit and interacting with a shared market for software and services, together with relationships among them. These relationships are frequently underpinned by a common technological platform and operate through the exchange of information, resources, and artifacts.”

From David G. Messerschmitt and Clemens Szyperski (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry*. Cambridge, MA, USA: MIT Press.



Example: The Global Information Grid (GIG)



GIG Complexity

Think about the number of entities in the GIG: satellites, ships, planes, ground vehicles, etc.

Think about the number of primary organizations that contribute to the creation of each of these already complex entities.

Think about the relationships that each of those primary organizations have with other secondary organizations.

Think about predicting the ripple effect of a decision such as choosing to adopt a specific open source product.



STREAM Overview

The **STR**ategic **E**cosystem **A**nalysis **M**ethod (STREAM) provides the framework within which a model of an ecosystem is created and measures such as total cost to supply and robustness are computed.

The techniques employed in STREAM allow the focus of the model to be specialized to the specific goals and issues of the software organization that deploys the method.

- For example, if having a *shorter time to market* is a goal of the organization, then the model would need to include data from the ecosystem elements about their *order-to-delivery times*.
- Similarly one can include cost data from the ecosystem elements if reduced total delivery cost is a goal.

STREAM reduces the complexity of an ecosystem to manageable levels by separation of concerns.



The STREAM Four Phases

Exploration – The modeling effort begins with an exploration of how the information in the ecosystem should be classified, represented, and analyzed. The team may construct prototype descriptions and evaluate how well the descriptions communicate to stakeholders.

Construction – The ecosystem model is being created. Data is collected, filtered, and represented.

Exploitation – The ecosystem model is being used as a decision-making aid. Decision makers use the relationships defined in the model to trace impacts and to predict the reactions to specific decisions.

Evolution – The ecosystem model is maturing with new information being added and existing information being modified or deleted to correct areas of inaccuracies.



STREAM Viewpoints

STREAM uses the architecture construct of a viewpoint to separate concerns and reduce the complexity of the ecosystem model.

The Business Viewpoint captures the ecosystem elements that are organizations.

The Software Viewpoint captures all of the software elements that belong to the ecosystem.

The Innovation Viewpoint captures the novelty in the ecosystem.



STREAM Business Viewpoint

The business viewpoint supports strategic decisions about feature selection, release schedules, and organizational activities such as partner recruitment and alliance formation.

- The ecosystem analysis may identify opportunities for joint marketing strategies, coordinated releases, and other collaborations.

The business viewpoint captures the ecosystem elements that are *organizations*.

- There are several relationships among organizations that are of potential interest, for example, one organization *supplies* another, one is a *competitor* of another, and many others.
- STREAM uses a modified version of a recognized business strategy tool, Porter's Five Forces for Business Strategy Development, to help identify the organizations of interest.



STREAM Software Viewpoint

The software viewpoint helps make strategic decisions about such issues as what should be included in the SDK.

- Which software elements are of such common interest that they should be bundled together?
- What tools are needed to support the use of these software elements?

The software viewpoint captures all of the *software elements* that belong to the ecosystem.

- The software usually can be clustered into *domains* such as program development tools and applications in a specific domain.
- The domain applications are often related to one another via a “uses” relationship.



STREAM Innovation Viewpoint

The innovation viewpoint captures classes of innovation, including

Product – Product innovations usually result in *new features*. These new features are reflected in the software viewpoint and are often linked to new markets in the business viewpoint.

Process – Process innovations result in new or modified techniques that enhance the *quality of the products* or that *deliver* those same products *in less time or at less cost*. New techniques may produce work products that reflect changes in scope or emphasis of the processes that create them.

Customer Experience – Innovations in the customer experience include providing unique features such as adding a live chat feature to the e-commerce web site product.



STREAM Analysis

Purpose of the analysis is to identify risks and make recommendations for improvements.

Risks to the hub organization are identified during the data collection and data modeling phases.

- Sensitivity points in the ecosystem can be computed.
- A scan of the supply chains may identify organizations with a poor record of timely deliveries.

The ecosystem model contains the data necessary to support a range of recommendations about the organizational management of the hub organization and its role in the ecosystem.

The recommendations come from the analysis of scenarios in the context of the ecosystem model.



STREAM Analysis: Techniques Used - 1

The information in the ecosystem model is used to understand how including a specific product in a complex system might ripple down the supply chain, requiring many additional acquisitions.

The CONOPS describes the operation of the software organization.

- The ecosystem model provides threads based on the roles of the members of the ecosystem.
- Each thread is an action that should be governed by the rules defined in the CONOPS.



STREAM Analysis: Techniques Used - 2

The business case is an economic analysis showing that the software development will be profitable.

- The ecosystem model provides an inventory of the users of the products as part of market forecasting.

The software architecture shows the structures and relationships among software elements.

- The ecosystem model can identify suppliers of industry standard, de facto standard, and otherwise useful software modules.
- These modules affect the structure of the software architecture.



MCE Case Study - 1

The primary question explored is whether the proposed form and content of the ecosystem model, and in particular the use of the three viewpoints, provides useful and sufficient information to strategic decision makers.

MCE is currently in the early phases of its work. The current products that MCE will be based on are

- long lived – about 3 to 5 years from inception to deployment on average and many years of service in the field
- installed on multiple different OS/hardware platforms with a large number of fielded installations
- coexist on the hardware target with software from other sources.



MCE Case Study - 2

The authors were embedded in the MCE as members of teams for several different activities including

- ecosystem modeling
- software architecture definition
- CONOPS analysis
- scope definition

The development of the format for the ecosystem model was carried out with input from personnel on several projects across several organizations.

An outline document describing the model including the three viewpoints, risks, and recommendation sections was sent out for comments on multiple occasions.



MCE Case Study - 3

Feedback from managers and engineers helped shape the model and clarify the documentation needs.

The initial version of the document was a combination of model and tutorial.

Subsequent versions presented more data and less tutorial material.

The detailed data migrated to appendices, leaving summaries in the main model.



MCE Recommendations - 1

The initial ecosystem modeling effort with MCE yielded a number of initial recommendations.

1. Establish a Community of Interest (COI) within the MCE Ecosystem that will promote open, continuous communication among members.
2. Grow the ecosystem by having attractive supplier incentives and by supplying the capabilities that users want faster and cheaper than previously possible.
3. Lower barriers to productivity between ecosystem members, particularly within clusters, by establishing collaborative relationships with users and suppliers.



MCE Recommendations - 2

4. Accommodate the diversity of the ecosystem members by looking for clusters of ecosystem members with similar needs and providing specialized asset bases for each.
5. Achieve agility by having governance structures that react quickly and by orchestrating processes, such as change management and unit testing, through automation.
6. Include an ecosystem impact analysis as a consideration in strategic decision making by identifying the ecosystem members likely to be affected and trading off the positive and negative impacts.



Questions?



References - 1

- [1] Bosch, J. “Software Product Lines to Software Ecosystems,” 111-119. *Proceedings of the 13th International Software Product Line Conference*. San Francisco, CA, Aug. 2009. ACM Press, 2009.
- [2] John D. McGregor. “A Method for Analyzing Software Product Line Ecosystems,” 73-80. *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*. Copenhagen, Denmark, Aug. 2010. AMC Press, 2010.
- [3] Philippe Kruchten. *The Rational Unified Process: An Introduction*, 2nd ed. Addison-Wesley, 2000.
- [4] Basili, Victor R., Caldiera, Gianluigi, & Rombach, H. Dieter. “The Goal Question Metric Approach.” *Encyclopedia of Software Engineering*, Wiley, 1994.
- [5] Gephi, Gephi.org (accessed 2/11/2012).
- [6] Adner, Ron. “Match Your Innovation Strategy to Your Innovation Ecosystem.” *Harvard Business Review* (April 2006).



References - 2

- [7] Lansiti, Marco & Levien, Roy. "Strategy as Ecology." *Harvard Business Review* (March 2004).
- [8] Viljainen, Martti & Kauppinen, Marjo. "Software Ecosystems: A Set of Management Practices for Platform Integrators in the Telecom Industry." *Lecture Notes in Business Information Processing 80* (2011): 32-43.
- [9] Baldwin, Carliss Y. "Where Do Transactions Come From? Modularity, Transactions, and the Boundaries of Firms." *Industrial and Corporate Change* 17, 1 (February 2008): 155-195.
- [10] Porter, Michael E. "How Competitive Forces Shape Strategy." *Harvard Business Review* (March 1979).
- [11] *ASA(ALT) Common Operating Environment Implementation Plan Core* (v3.0 Draft, Distribution Statement A). MITRE, 2011.
<http://usarmy.vo.llnwd.net/e2/c/downloads/232001.pdf>



Reference - 3

[12] “Global Information Grid Operational View.” Wikipedia, 2008.
http://en.wikipedia.org/wiki/File:Gig_ov1.jpg

[13] Messerschmitt, David G. & Szyperski, Clemens. *Software Ecosystem: Understanding an Indispensable Technology and Industry*. MIT Press, 2003.



Acknowledgments

We gratefully acknowledge Project Manager Force XXI Battle Command Brigade and Below (PM-FBCB2), which funded the work described in the Case Study under agreement PWS #5-357D.

This presentation is based on the paper “Ecosystem Modeling in a Software Product Line Organization Using STREAM.” We wish to thank the PEO C3T Public Release Office for clearing that paper for open publication.



Contact Information

Gary Chastek, PhD

Senior Member of the Technical Staff

Research, Technology, and System Solutions

Telephone: +1 412-268-2826

Email: gjc@sei.cmu.edu

Web

www.sei.cmu.edu

www.sei.cmu.edu/contact.cfm

U.S. Mail

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Customer Relations

Email: info@sei.cmu.edu

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257



Acronyms

COE	U.S. Army's Common Operating Environment
COI	Community of Interest
GIG	Global Information Grid
MCE	Mounted Computing Environment
PM-FBCB2	Project Manager Force XXI Battle Command Brigade and Below
SDK	Software Development Kit
STREAM	STRategic Ecosystem Analysis Method



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.

