ULS Systems Research Roadmap

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

June 2007

Background

Increasing Scale In Military Systems

Increasingly Complex Systems

- ultra-large, network-centric, real-time, cyber-physical-social systems
  - thousands of platforms, sensors, decision nodes, weapons, and warfighters
  - connected through heterogeneous wired and wireless networks

Goal: *Information Dominance*

- *Transient and enduring resource constraints and failures*
- *Continuous adaptation*
  - changes in mission requirements
  - changes in operating environments
  - changes in force structure
  - perpetual systems' evolution
  - addition of new systems
- *Sustainable - legally, technically, politically*

A Reason for Concern

Such systems are going to be larger and more complex than any previously seen

- very serious technical challenges, obvious and undoubtedly to-be-discovered
- many vendors, many technologies, many systems
- evolving doctrine + evolving technology + (or $\Rightarrow$?) ill-defined requirements

The US Army is concerned that the scale of future systems is beyond our reach.

Ultra-Large-Scale (ULS) Systems Study

Gather leading experts to study:

- characteristics of ULS systems
- challenges and breakthroughs required
- promising research and approaches

Intended outcomes:

- ULS System Research Agenda
- program proposal
- collaborative research network

About the Effort

Funded by the Army (ASA ALT)

Staffing: 9 member SEI team
13 member expert panel

Duration: one year (04/05 -- 05/06)

ULS Systems Research Study Report

Acknowledgements

Executive Summary

Part I

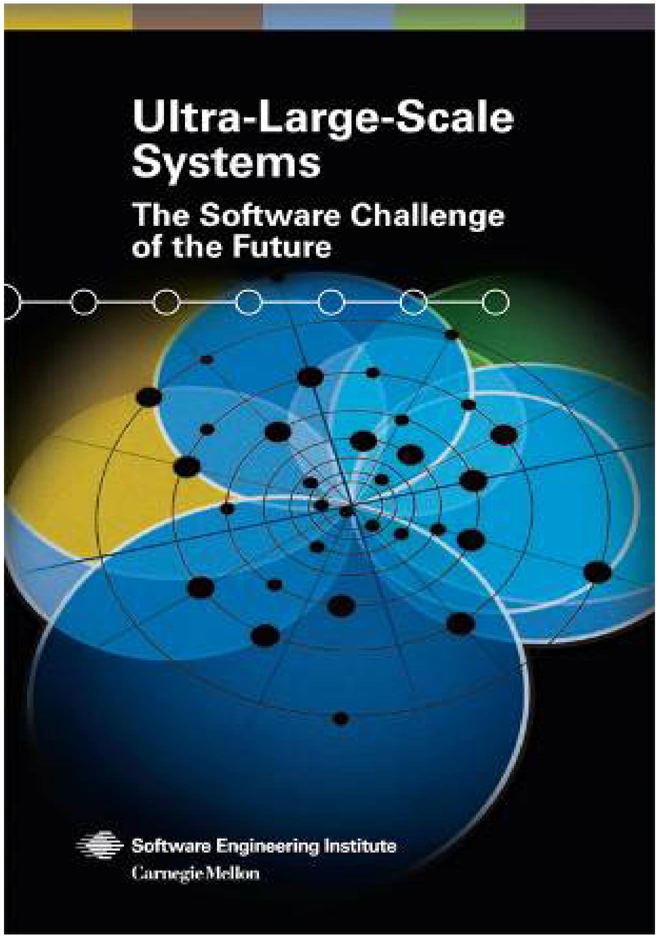
1. Introduction
2. Characteristics of ULS Systems
3. Challenges
4. Overview of Research Areas
5. Summary and Recommendations

Part 2

6 Detailed Description of Research Areas

- Glossary

http://www.sei.cmu.edu/uls/



What Is an Ultra-Large-Scale (ULS) System?

A ULS System has unprecedented scale in some of these dimensions:

- Lines of code
- Amount of data stored, accessed, manipulated, and refined
- Number of connections and interdependencies
- Number of hardware elements
- Number of computational elements
- Number of system purposes and user perception of these purposes
- Number of routine processes, interactions, and “emergent behaviors”
- Number of (overlapping) policy domains and enforceable mechanisms
- Number of people involved in some way

ULS systems will be interdependent webs of software-intensive systems, people, policies, cultures, and economics.

ULS systems are systems of systems at internet scale.

Scale Changes Everything

Characteristics of ULS systems arise because of their scale.

- Decentralization
- Inherently conflicting, unknowable, and diverse requirements
- Continuous evolution and deployment
- Heterogeneous, inconsistent, and changing elements
- Erosion of the people/system boundary
- Normal failures
- New paradigms for acquisition and policy

These characteristics may appear in today's systems and systems of systems, but in ULS systems they dominate.

These characteristics undermine the assumptions that underlie today's software engineering approaches.

Study Conclusions

There are fundamental gaps in our current understanding of software development at the scale of ULS systems.

These gaps

- present profound impediments to the technically and economically effective achievement of the DoD goals*
- require a broad, fresh perspective and interdisciplinary, breakthrough research

We recommended

- a ULS Systems Research Agenda that included research areas based on a fresh perspective aimed at challenges arising from increasing scale

** As stated in the Quadrennial Defense Review (QDR) Report, Feb 2006*

Moving Forward

The ULS System Research Agenda did not define a roadmap or discrete fundable research packages

The Army needed more specificity to move forward

CERDEC funded the creation of a roadmap for a portion of the ULS System Research Agenda

The remainder of this presentation introduces the ULS Systems Research Roadmap

The Research Roadmap

Roadmap Intent

Motivate Research

- The roadmap shows how an individual research initiative (a 3-4 year effort of $1M/year) supports one or more ULS-system technical challenges

Help evaluate the ULS systems relevance of existing or planned research

- The roadmap structure explicitly shows a ULS system perspective

Prioritize research funding

- The roadmap provides a basis for determining which research is most critical/relevant/impactful for achieving a future ULS systems capability

Framework for incorporating additional ULS systems research

Roadmap Overview

Six Research Objectives

- Combining 9 (of 30) Research Topics from the ULS Systems Report
- 16 potential basic research initiatives (6.1)
- 8 potential applied research initiatives (6.2)
- Each initiative is a suitable 3-4 year effort at ($1M/yr)
- Total program: $24M/year (if all were funded)

Six ULS Systems Technical Objectives addressed by the Research Objectives

Five ULS Systems Perspectives for evaluating current and proposed research

Information showing how the research initiatives are related to the technical objectives and system perspectives

Significant contributions of the roadmap

- Shows how research initiatives map to DoD technical challenges and vice versa
- Presents solutions to DoD challenges from multiple perspectives
  - e.g., combines software engineering *and* research from other fields

Roadmap Structure and Development Process

Start with: a needed ULS system warfighter **capability**

Make: **Observations** about this capability

- Example: user needs change dynamically

Use: ULS systems **perspective** (contrasted with conventional approach)

Identify: **Technical challenge** (related to ULS systems perspective)

- Contrast with the "usual" technical challenge

Restate challenge as: **Research objective**

Cite: ULS Systems report **Research Topic**

Define **Research Initiatives**: Several supporting each research objective

- Some research initiatives contribute to more than one research objective
  - Such initiatives provide opportunities for cross-cutting leverage and funding impact

The Roadmap Root: A Warfighter Capability

The ULS report mentioned six capabilities needed by the DoD

The roadmap combines two of them (C1/C6) into a single capability to show research relevance to desired military capabilities

- **Common Relevant Operational Picture (CROP)**: Maintain coherent common operating picture
  - across echelons, services, and coalitions in a mix of ultra-large-scale environments (C1)
  - applying local context to global information sources to ensure use of the right data any time, any place, for any mission (C6)

A Needed Warfighter Capability

Common Relevant Operational Picture: Maintain coherent common operating picture by rapidly collecting, processing, disseminating, and protecting information spanning echelons, services, and coalitions across a mix of ultra-large-scale environments. Apply local context to global information sources to ensure use of the right data any time, any place, for any mission.

Technical Observations

Common Relevant Operational Picture: Maintain coherent common operating picture by rapidly collecting, processing, disseminating, and protecting information spanning echelons, services, and coalitions across a mix of ultra-large-scale environments. Apply local context to global information sources to ensure use of the right data any time, any place, for any mission.

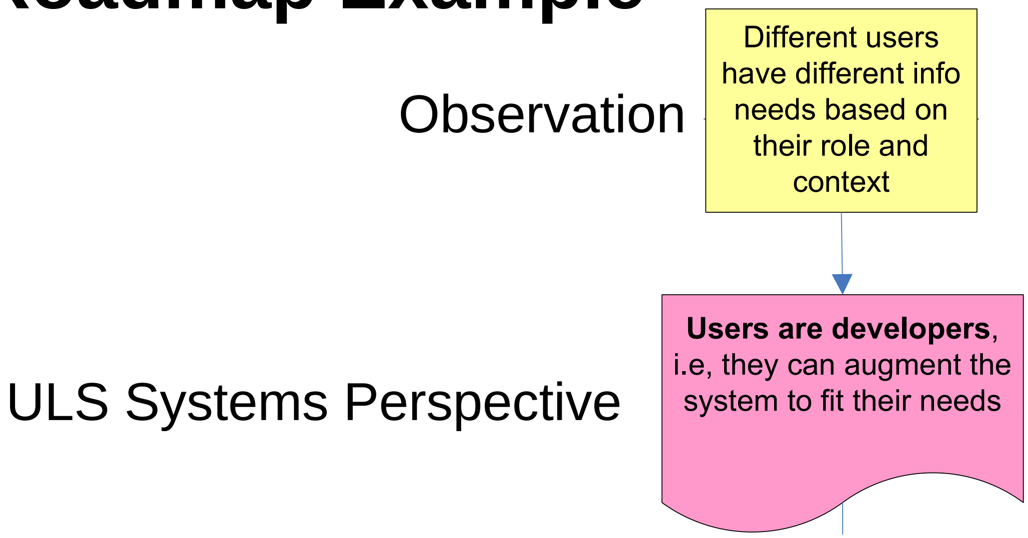
- **Different users have different info needs based on their role and context**
- User needs for info change dynamically
- System connectivity and info flow changes dynamically
- People will (mis)use the system in unexpected ways, stressing HW and SW
- CROP capability evolves non-uniformly in its structure, components, and uses

Roadmap Example

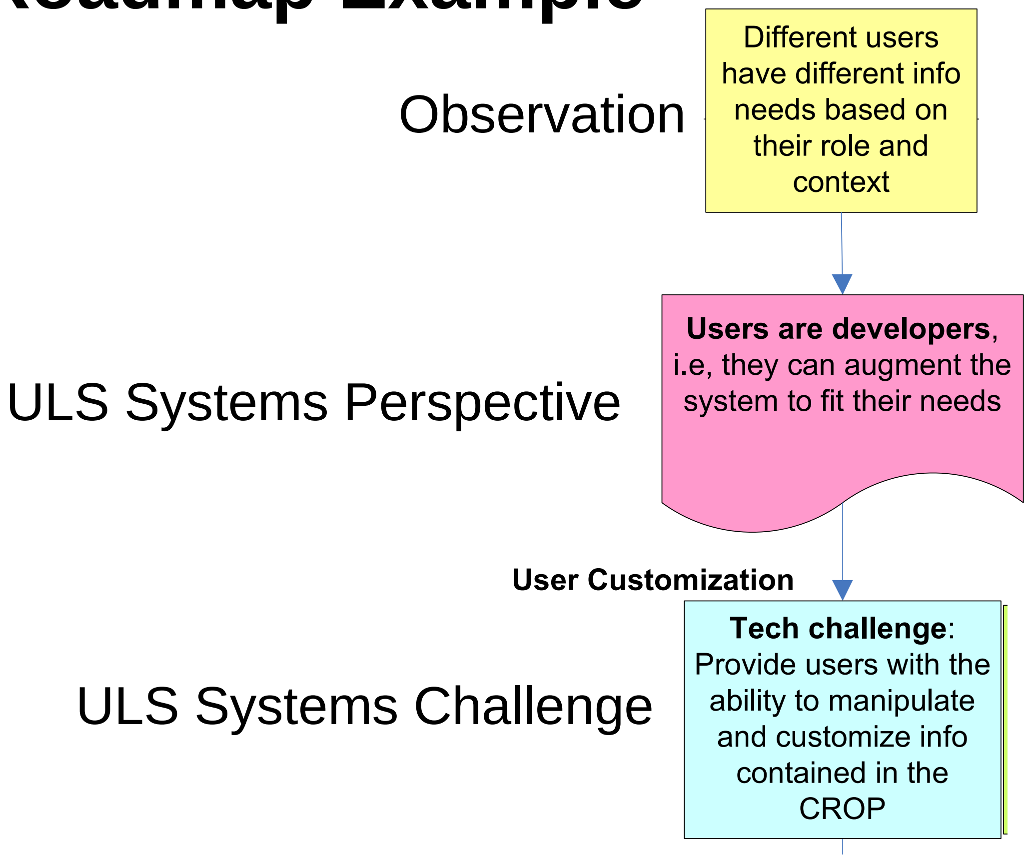
Observation

Different users have different info needs based on their role and context

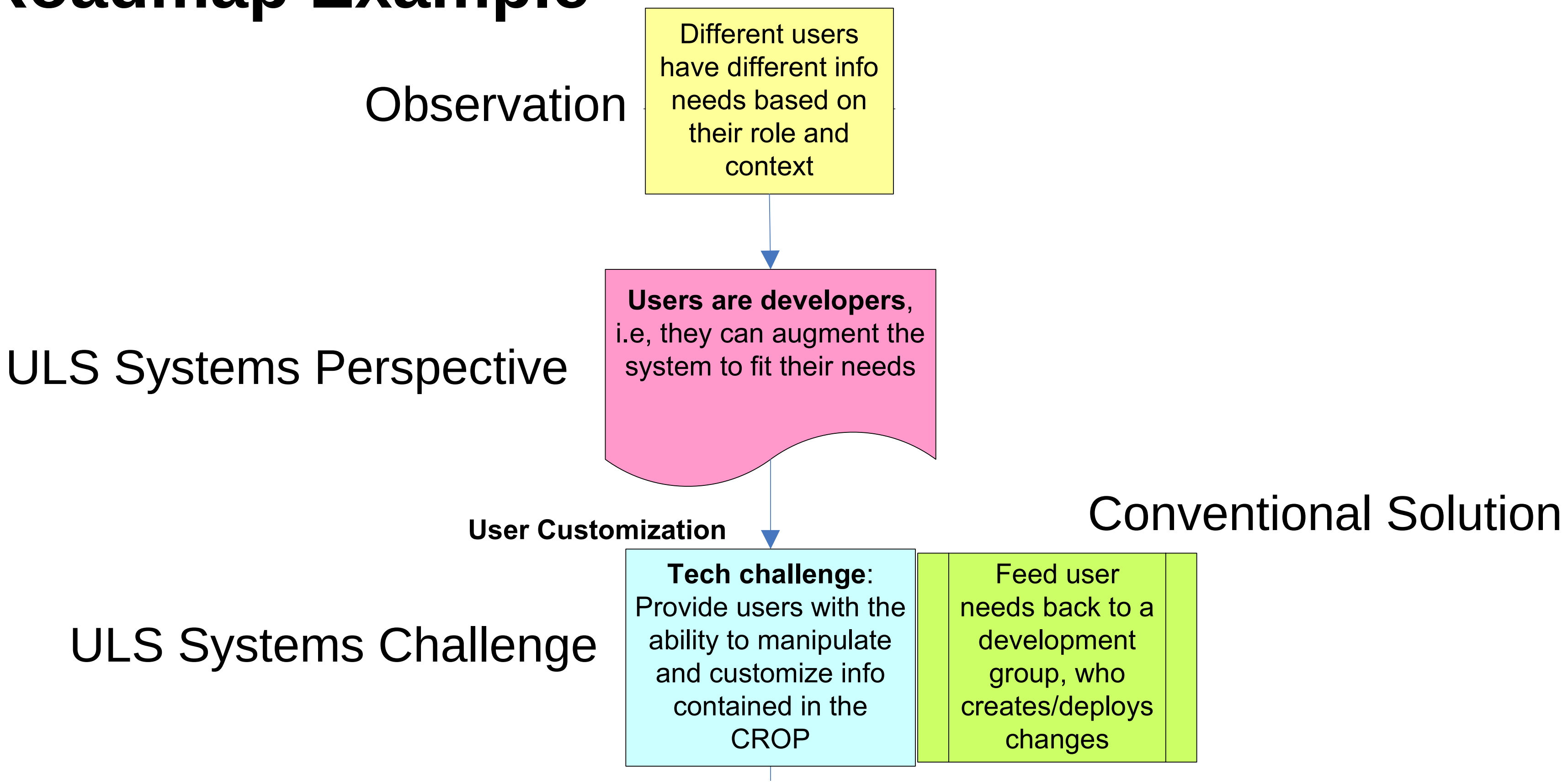
Roadmap Example



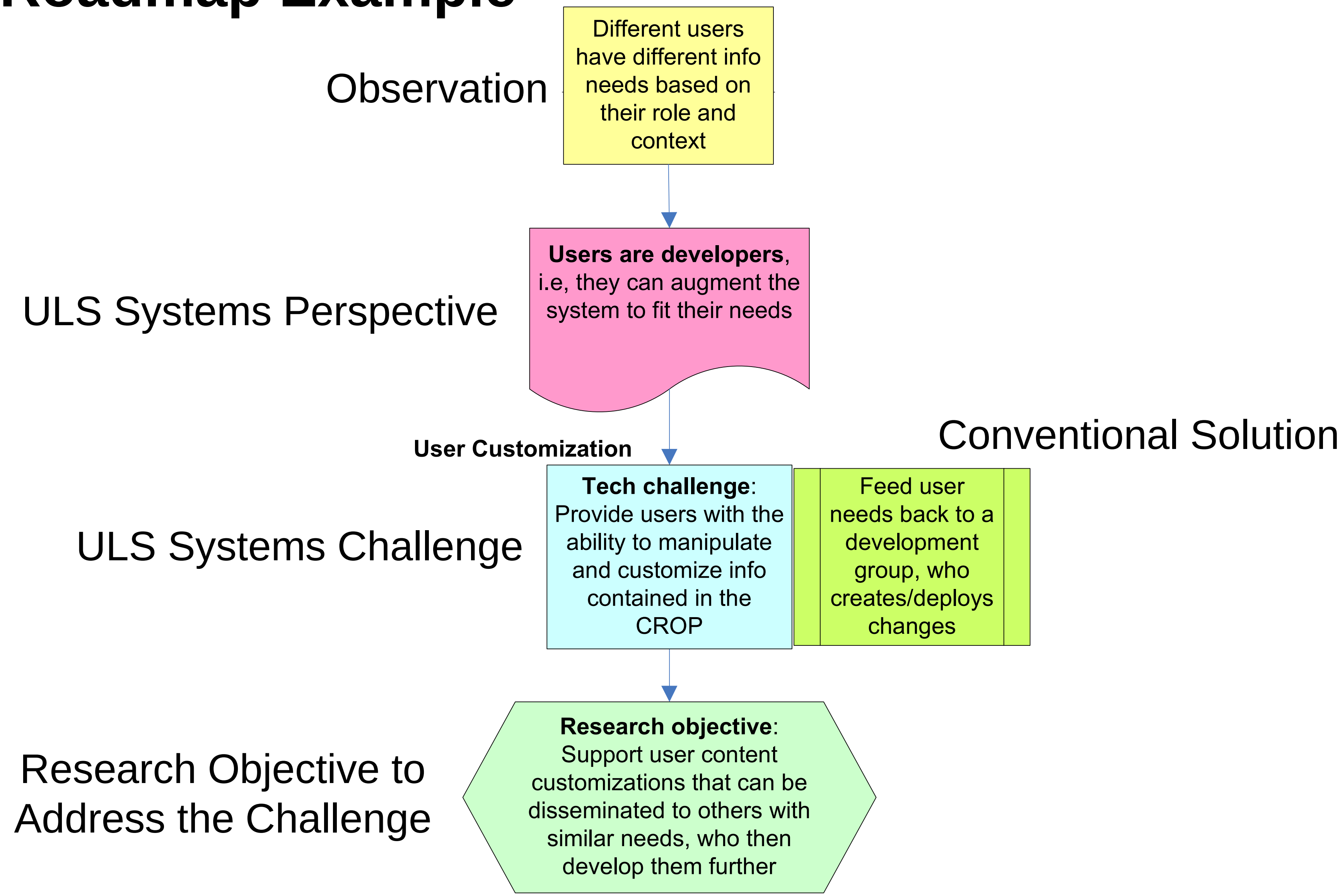
Roadmap Example



Roadmap Example



Roadmap Example



ULS System Perspectives

Manage differences rather than eliminate them

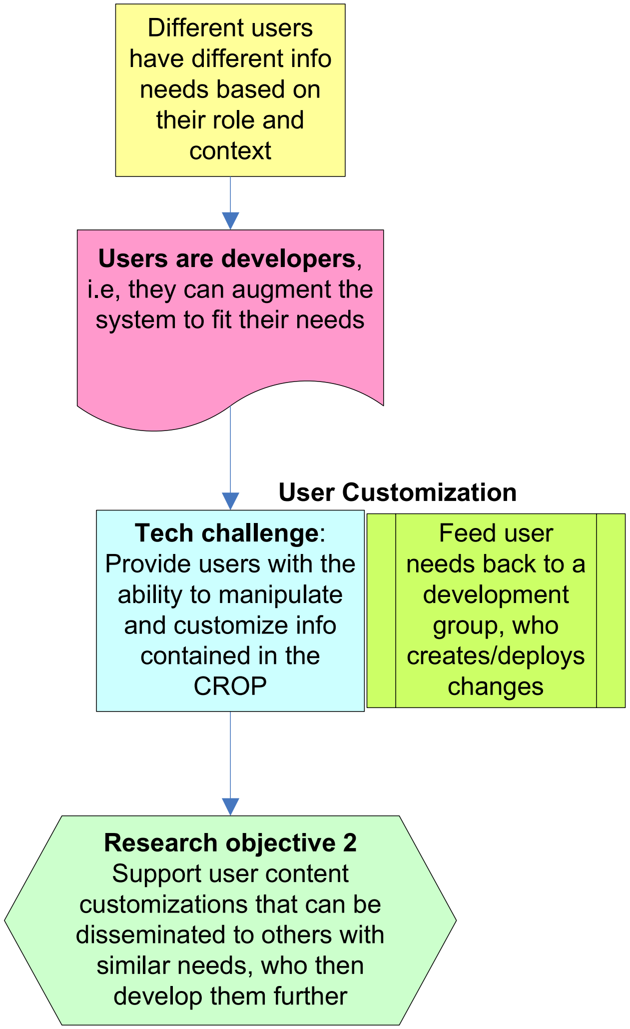
Users are developers, i.e, they can augment the system to fit their needs

Exploit self-interest rather than overriding it

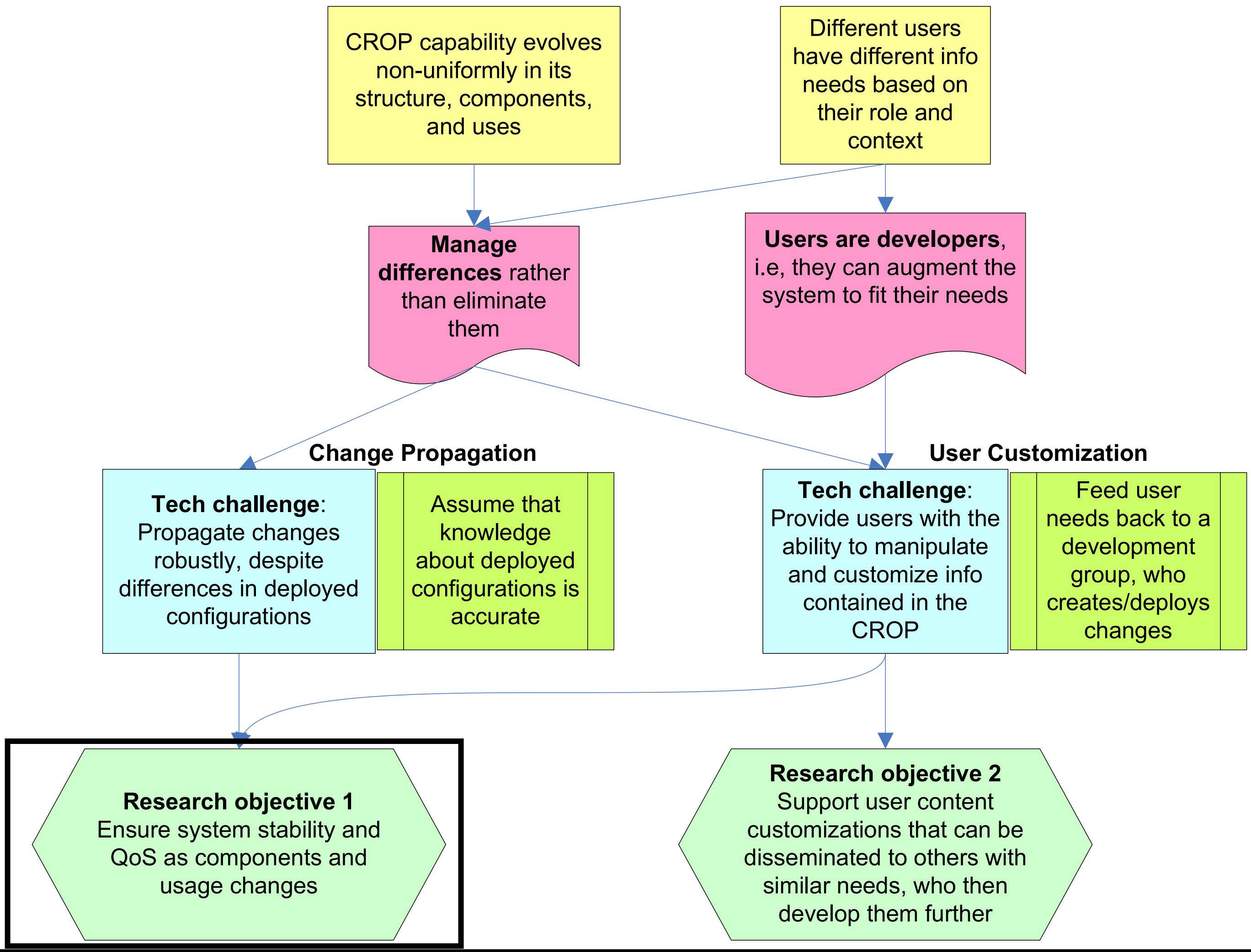
Automated adaptation rather than manual intervention to deal with failures and differing operational conditions

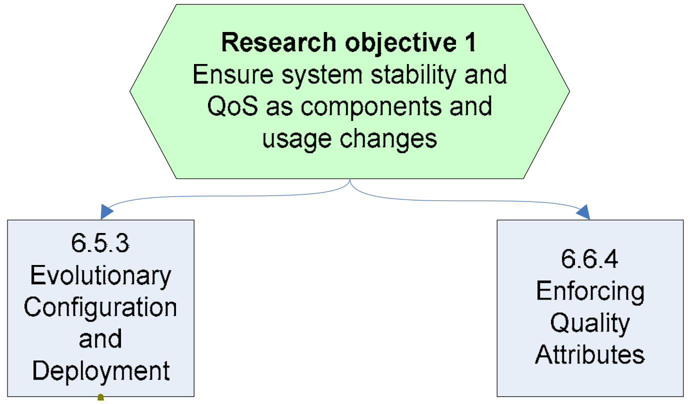
Failures are normal, i.e., because of scale, even unlikely events will occur

Previous Roadmap Example

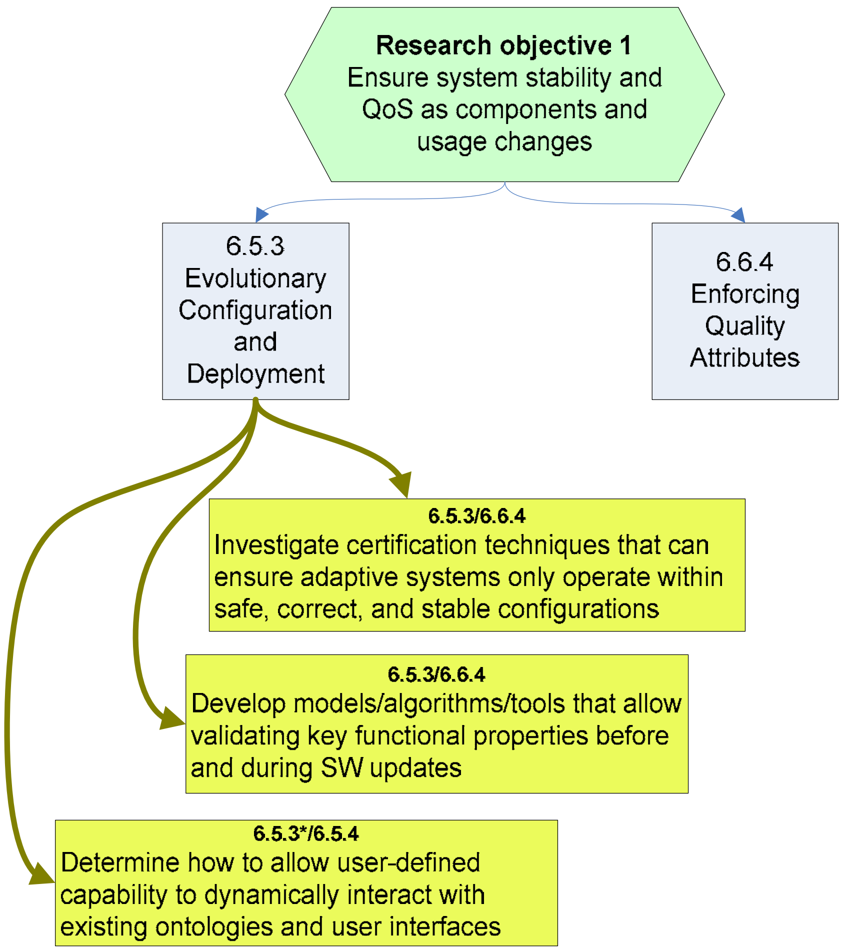


Augmented Roadmap Example



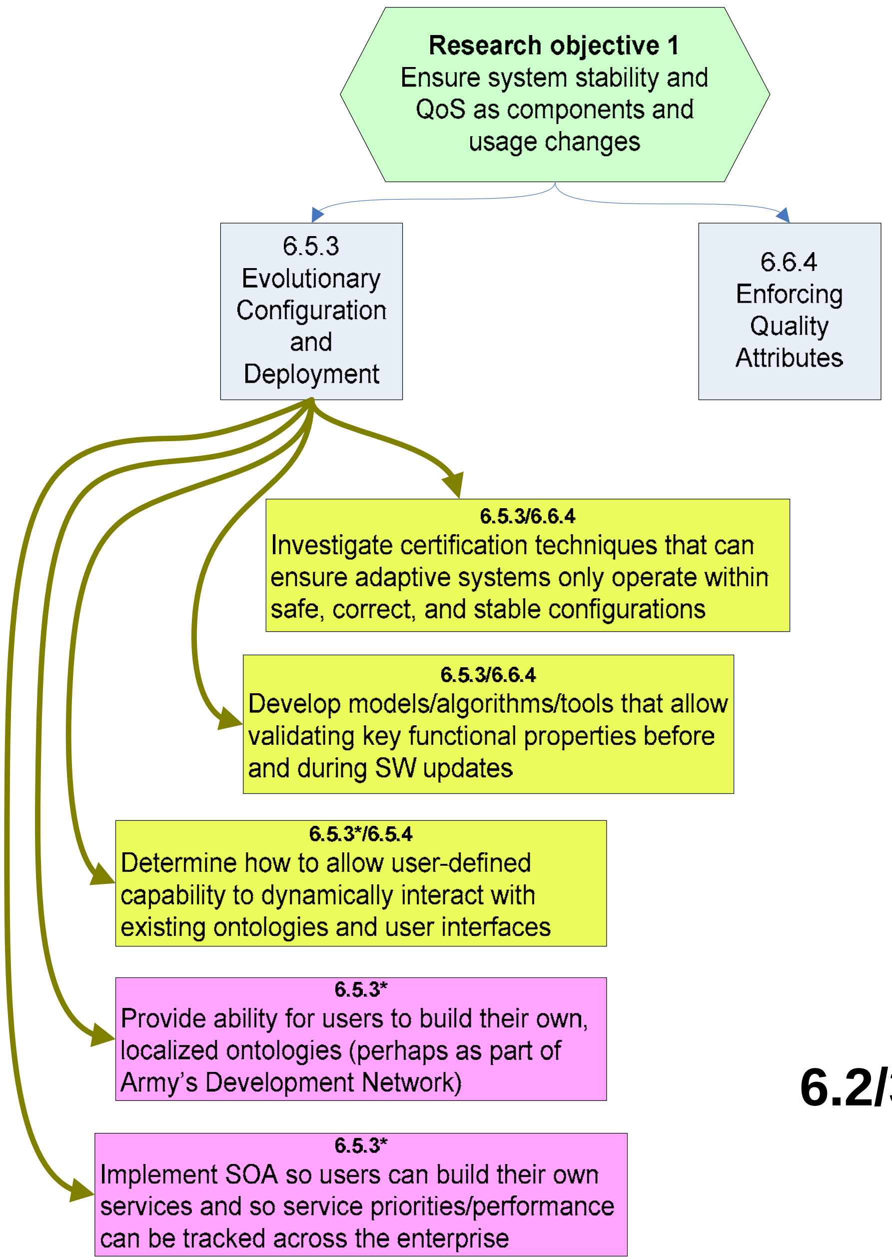


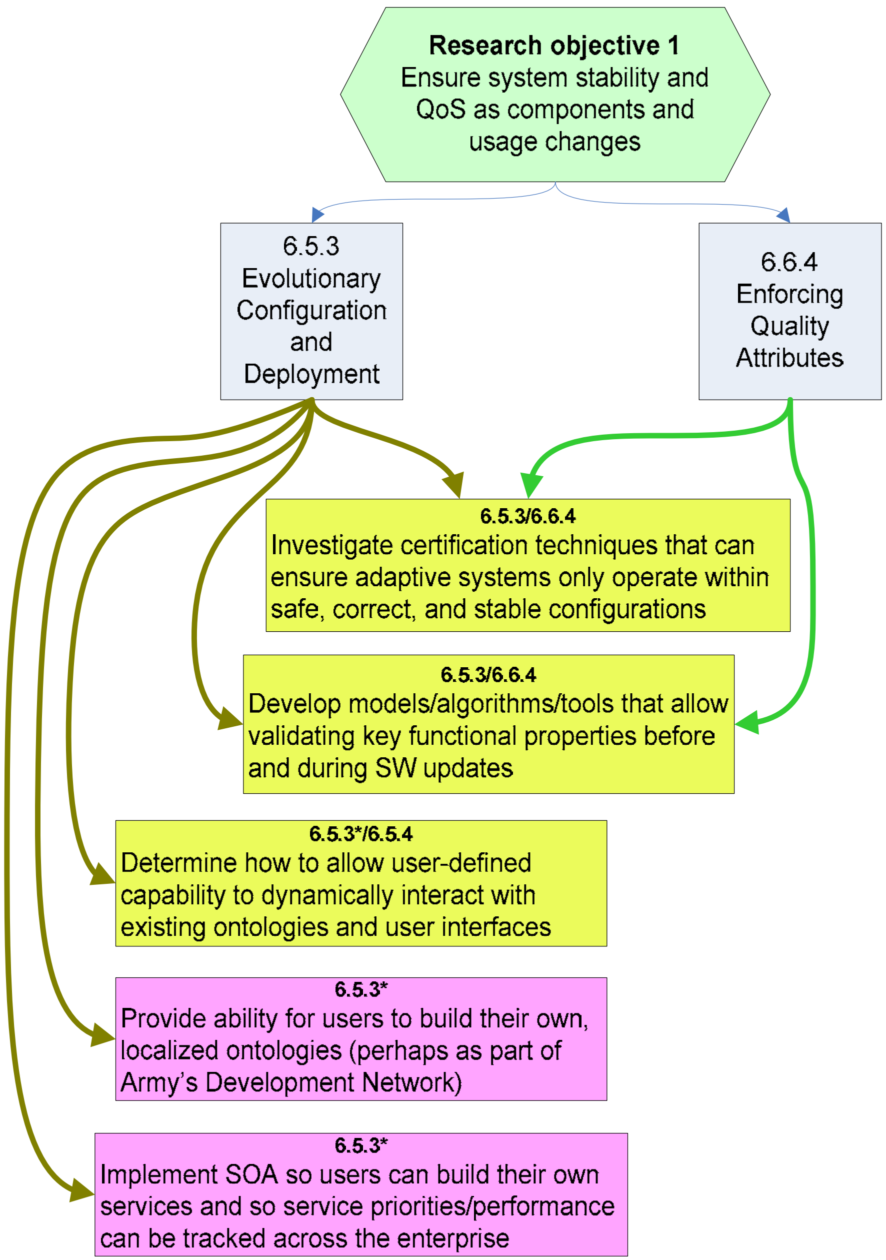
Research Topic from ULS Systems Report

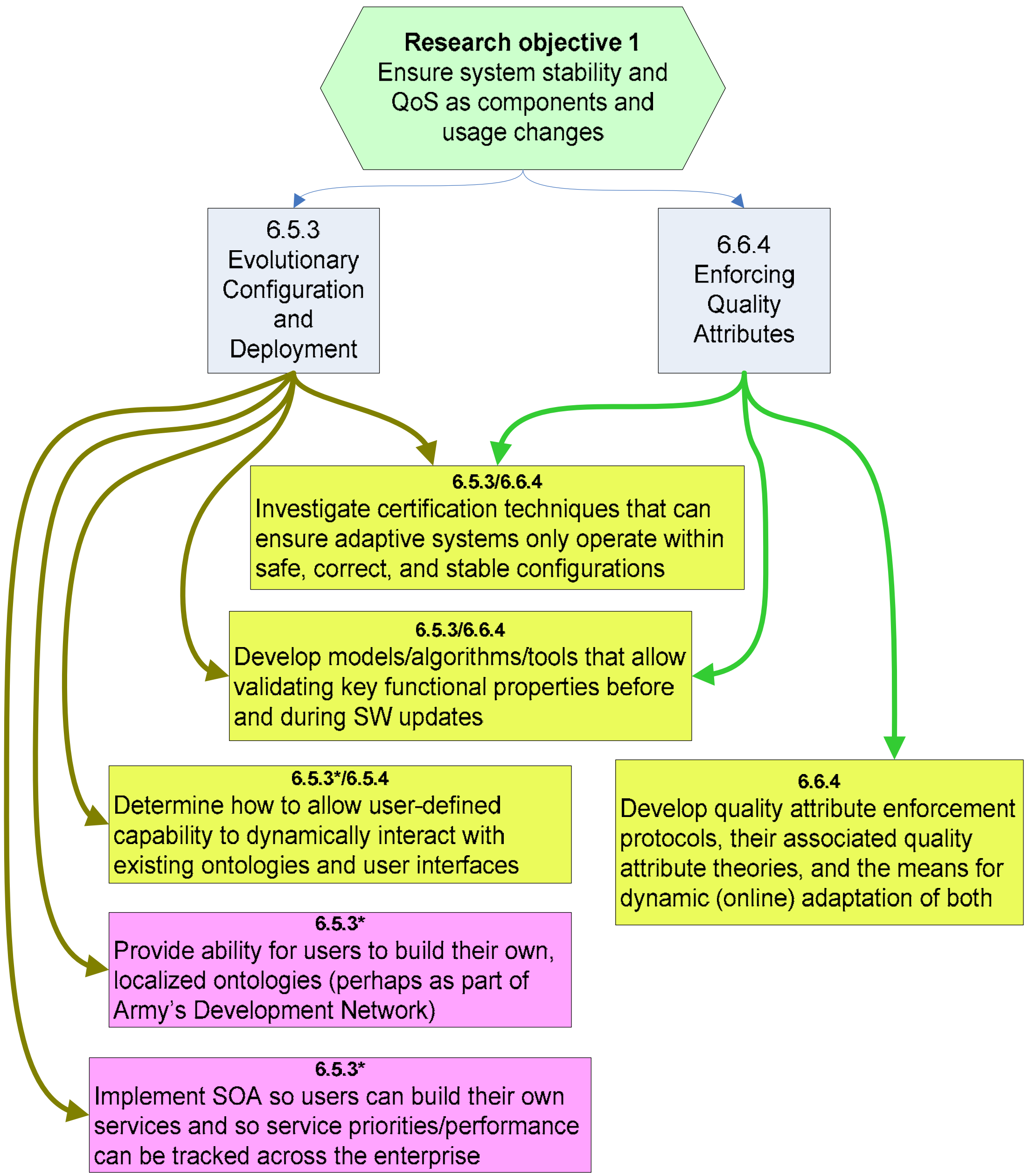

Research objective 1
Ensure system stability and QoS as components and usage changes
6.5.3 Evolutionary Configuration and Deployment
6.6.4 Enforcing Quality Attributes
6.5.3/6.6.4
Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations
6.5.3/6.6.4
Develop models/algorithms/tools that allow validating key functional properties before and during SW updates
6.5.3*/6.5.4
Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces

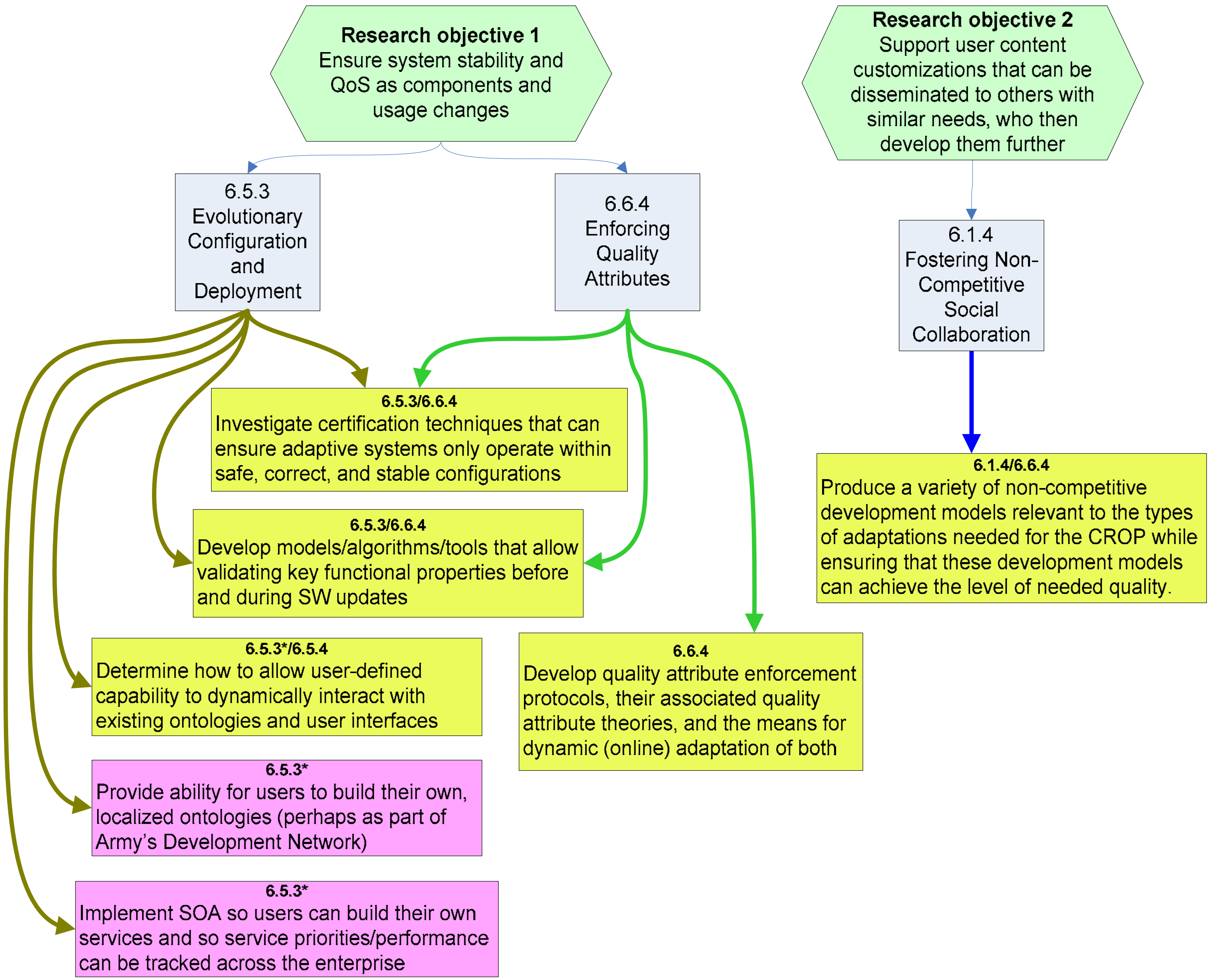
6.1 Research

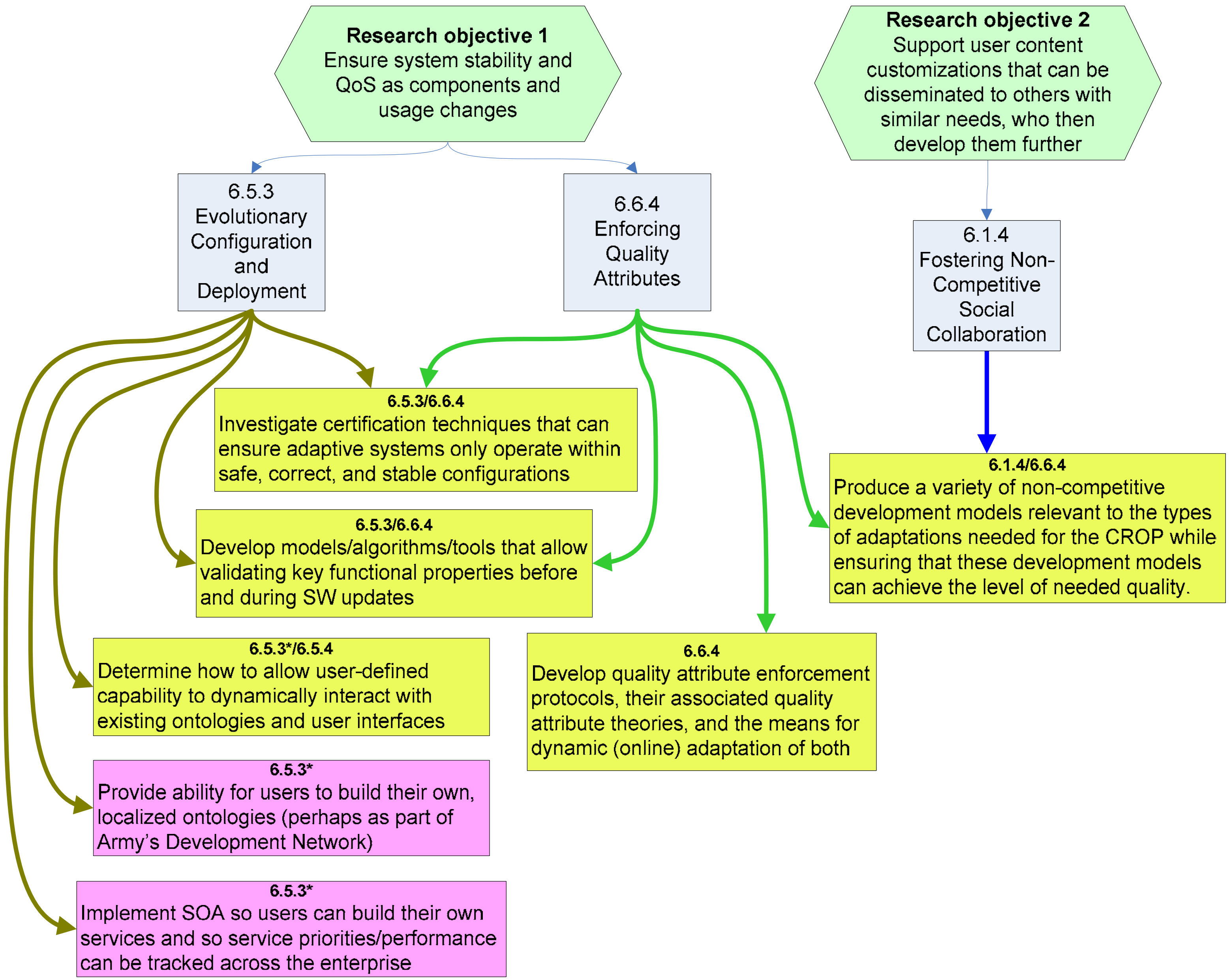
6.2/3 Research




Research objective 1
Ensure system stability and QoS as components and usage changes
6.5.3 Evolutionary Configuration and Deployment
6.6.4 Enforcing Quality Attributes
6.5.3/6.6.4
Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations
6.5.3/6.6.4
Develop models/algorithms/tools that allow validating key functional properties before and during SW updates
6.5.3*/6.5.4
Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces
6.5.3*
Provide ability for users to build their own, localized ontologies (perhaps as part of Army's Development Network)
6.5.3*
Implement SOA so users can build their own services and so service priorities/performance can be tracked across the enterprise


Research objective 1
Ensure system stability and QoS as components and usage changes
6.5.3 Evolutionary Configuration and Deployment
6.6.4 Enforcing Quality Attributes
6.5.3/6.6.4
Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations
6.5.3/6.6.4
Develop models/algorithms/tools that allow validating key functional properties before and during SW updates
6.5.3*/6.5.4
Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces
6.6.4
Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
6.5.3*
Provide ability for users to build their own, localized ontologies (perhaps as part of Army's Development Network)
6.5.3*
Implement SOA so users can build their own services and so service priorities/performance can be tracked across the enterprise


Research objective 1
Ensure system stability and QoS as components and usage changes
Research objective 2
Support user content customizations that can be disseminated to others with similar needs, who then develop them further
6.5.3 Evolutionary Configuration and Deployment
6.6.4 Enforcing Quality Attributes
6.1.4 Fostering Non-Competitive Social Collaboration
6.5.3/6.6.4
Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations
6.5.3/6.6.4
Develop models/algorithms/tools that allow validating key functional properties before and during SW updates
6.1.4/6.6.4
Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality.
6.5.3*/6.5.4
Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces
6.6.4
Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
6.5.3*
Provide ability for users to build their own, localized ontologies (perhaps as part of Army's Development Network)
6.5.3*
Implement SOA so users can build their own services and so service priorities/performance can be tracked across the enterprise


Research objective 1
Ensure system stability and QoS as components and usage changes
Research objective 2
Support user content customizations that can be disseminated to others with similar needs, who then develop them further
6.5.3 Evolutionary Configuration and Deployment
6.6.4 Enforcing Quality Attributes
6.1.4 Fostering Non-Competitive Social Collaboration
6.5.3/6.6.4
Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations
6.5.3/6.6.4
Develop models/algorithms/tools that allow validating key functional properties before and during SW updates
6.1.4/6.6.4
Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality.
6.5.3*/6.5.4
Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces
6.6.4
Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
6.5.3*
Provide ability for users to build their own, localized ontologies (perhaps as part of Army's Development Network)
6.5.3*
Implement SOA so users can build their own services and so service priorities/performance can be tracked across the enterprise

Roadmap Structure and Development Process

Start with: a needed ULS system warfighter **capability**

Make: **Observations** about this capability

- Example: user needs change dynamically

Use: ULS systems **perspective** (contrasted with conventional approach)

Identify: **Technical challenge** (related to ULS systems perspective)

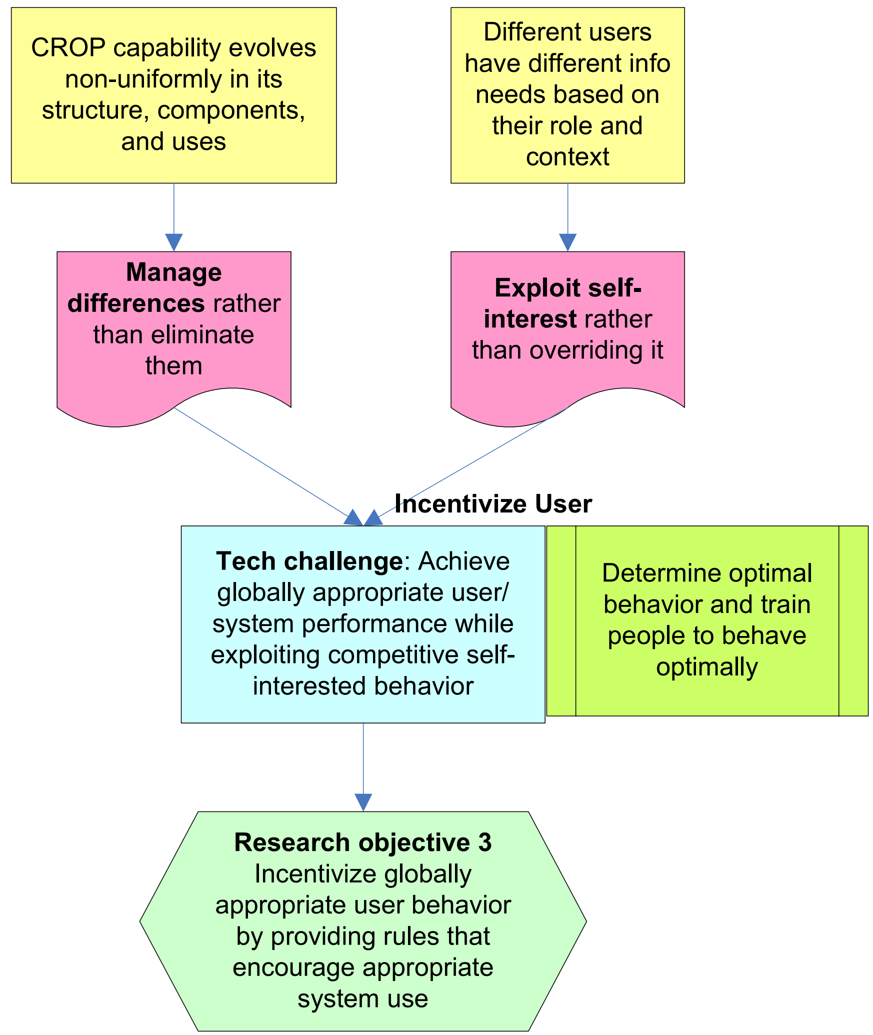
- Contrast with the "usual" technical challenge

Restate challenge as: **Research objective**

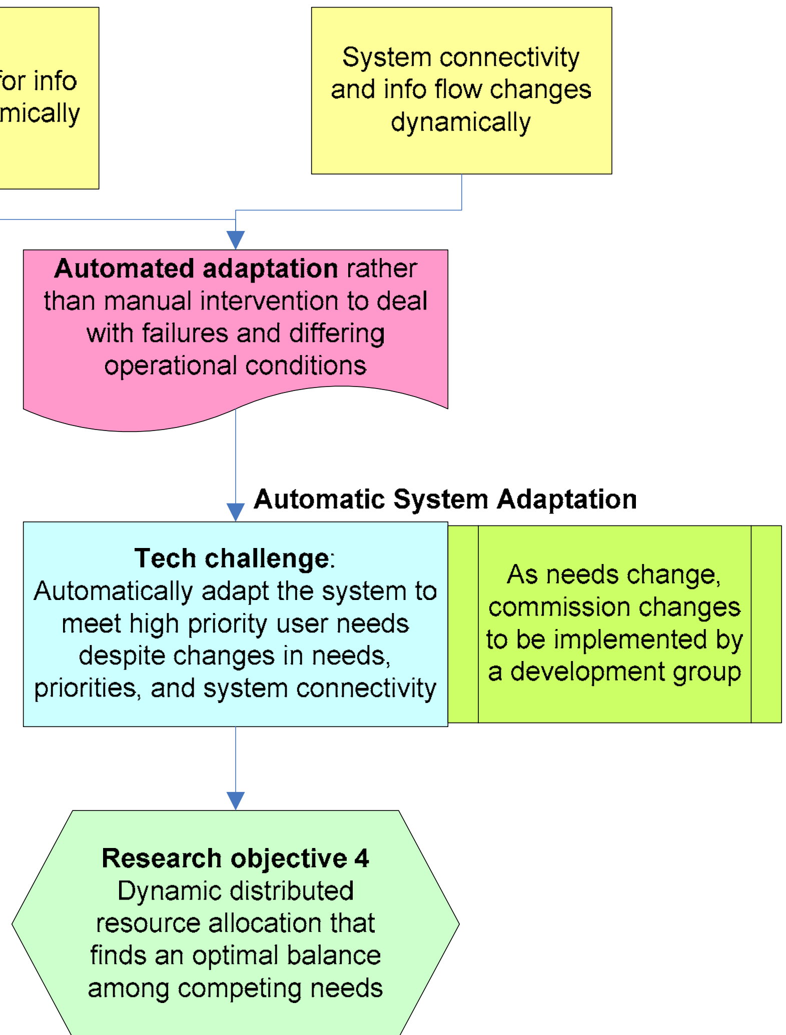
Cite: ULS Systems report **Research Topic**

Define **Research Initiatives**: Several supporting each research objective

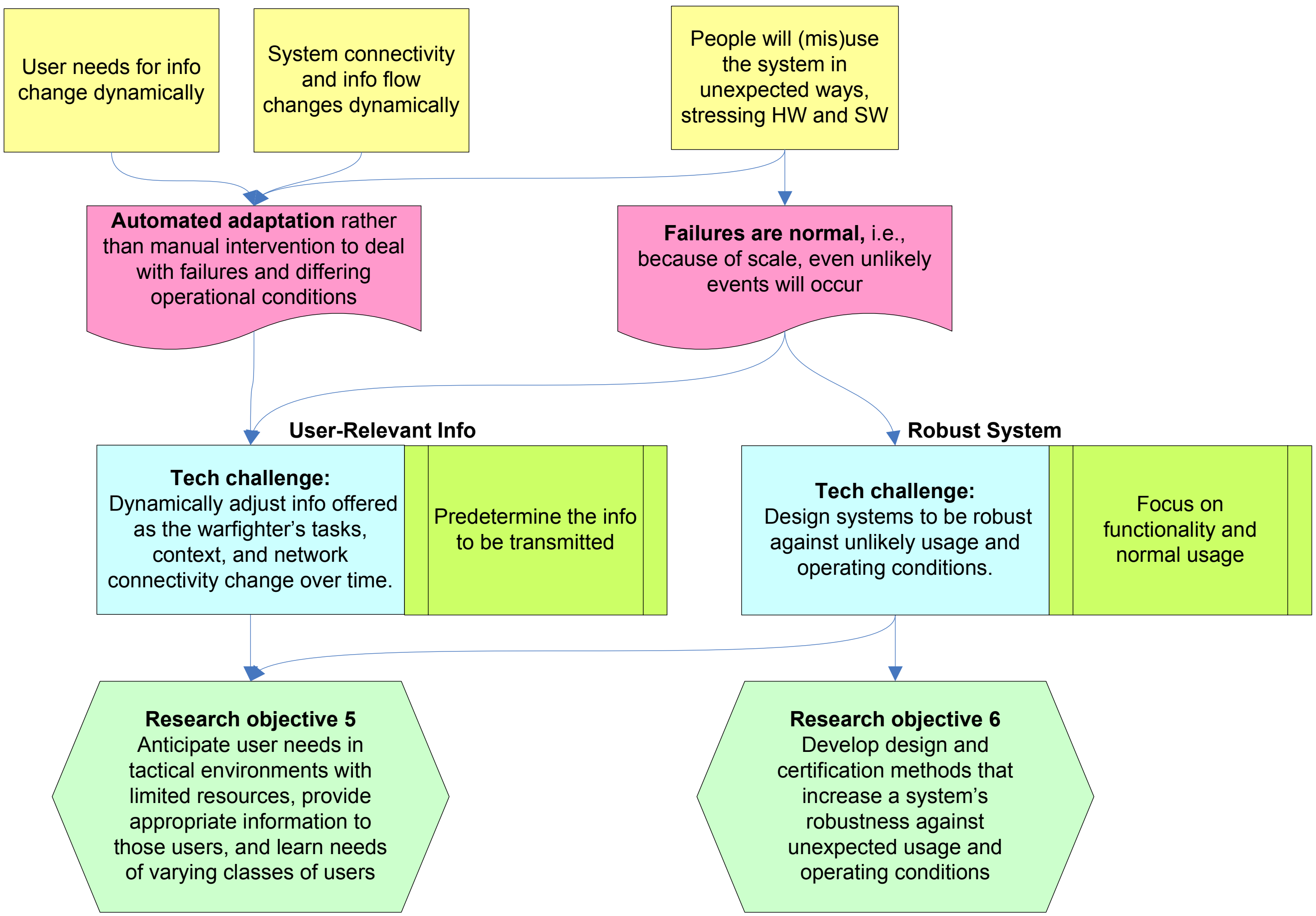
Incentivize User



Automatic System Adaptation



User Relevance and Robustness



ULS Systems Research Topics *In*/Not In Roadmap

6.1.1 Context-Aware Assistive Computing

6.1.2 Understanding Users and Their Contexts

6.1.3 Modeling Users and User Communities

6.1.4 Fostering Non-Competitive Social Collaboration

6.1.5 Longevity

6.2.1 Algorithmic Mechanism Design

6.2.2 Metaheuristics in Software Engineering

6.2.3 Digital Evolution

6.3.1 Design of All Levels

6.3.2 Design Spaces and Design Rules

6.3.3 Harnessing Economics to Promote Good Design

6.3.4 Design Representation/Analysis

6.3.5 Assimilation

6.3.6 Determining and Managing Requirements

6.4.1 Expressive Representation Languages

6.4.2 Scaled-Up Specification, Verification, and Certification

6.4.3 Computational Engineering for Analysis and Design

6.5.1 Decentralized Production Management

6.5.2 View-Based Evolution

6.5.3 Evolutionary Configuration and Deployment

6.5.4 In Situ Control and Adaptation

6.6.1 Robustness, Adaptation, and Quality Attributes

6.6.2 Scale and Composition of Quality Attributes

6.6.3 Understanding People-Centric Qual. Attr.

6.6.4 Enforcing Quality Requirements

6.6.5 Security, Trust, and Resiliency

6.6.6 Engineering Management at Ultra-Large Scales

6.7.1 Policy Definition for ULS Systems

6.7.2 Fast Acquisition for ULS Systems

6.7.3 Management of ULS Systems

Research Objectives

RO1: **Ensure system stability and QoS** as components and usage change 5 2

RO2: Support **user-created customizations** that can be **disseminated** to others with similar needs, who then develop them further 4 0

RO3: **Incentivize globally appropriate user behavior** by providing rules that encourage appropriate system use 3 1

RO4: **Dynamic distributed resource allocation** that finds an optimal balance among competing needs 13 4

RO5: **Anticipate user needs** in tactical environments with limited resources, provide appropriate information to those users, and learn needs of varying classes of users 8 4

RO6: Develop design and certification methods that **increase a system's robustness** against unexpected usage and operating conditions 5 0

Objective 1: Ensure System Stability and QoS

6.5.3 Evolutionary Configuration and Deployment

- Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces (see also 6.5.4, *In situ* Control and Adaptation)
- Provide ability for users to build their own, localized ontologies (perhaps as part of Army's Development Network)*
- Implement SOA so users can build their own services and so service priorities/performance can be tracked across the enterprise*

6.6.4 Enforcing Quality Attributes

- Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
- Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations (see also 6.5.3, Evolutionary Configuration and Deployment)
- Develop models/algorithms/tools that allow validating key functional properties before and during SW updates (see also 6.5.3, Evolutionary Configuration and Deployment)
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.1.4, Fostering Non-Competitive Social Collaboration)

Objective 2: User Customization/Dissemination

6.1.4 Fostering Non-Competitive Social Collaboration

- Define and test incentive structures for their ability to guide non-competitive development processes
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.6.4, Enforcing Quality Attributes)

6.6.3 Understanding People-Centric Quality Attributes

- Develop models and methods so warfighters have appropriate levels of trust (or mistrust) in the information being presented
- Integrate people-centric models that show how human performance and reliability contribute to overall system performance and reliability, and how human interactions, mediated by the system, affect overall mission success

Objective 3: Incentivize Global User Behavior

6.2.1 Algorithmic Mechanism Design

- Explore control-theoretic methods for handling rapidly changing demands and changing resource availability profiles; explore impact of service policies tuned for different system operating modes (see also 6.5.4, *In situ* Control and Adaptation, and 6.1.1, Context-Aware Assistive Computing)
- Given a lack of centralized control over individual behavior, design the CROP so info contributed to and extracted from it arises from (or is consistent with) the natural self-interests of individuals
- Apply auction mechanisms within the computational infrastructure to determine appropriate allocation of resources
- Demonstrate theoretical and empirical properties of different controller solutions in a prototype that reflects operational conditions (see also 6.1.1, Context-Aware Assistive Computing, and 6.5.4, *In Situ* Control and Adaptation

6.6.4 Enforcing Quality Attributes

- Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
- Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations (see also 6.5.3, Evolutionary Configuration and Deployment)
- Develop models/algorithms/tools that allow validating key functional properties before and during SW updates (see also 6.5.3, Evolutionary Configuration and Deployment)
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.1.4, Fostering Non-Competitive Social Collaboration)

Objective 4: Balanced Dynamic Resource Alloc. – 1

6.5.4 *In situ* Control and Adaptation

- Create mechanisms such that when the system changes in ways that are visible to warfighters, either existing warfighter views are adapted to new system states or the effects on the warfighter are moderated (see also 6.1.1, Context-Aware Assistive Computing)
- Develop models that represent users and their communities; attach the models to system instrumentation and mechanisms allowing the system to adapt and reflect the model (6.1.3, Modeling Users and User Communities)
- Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces (6.6.3, Understanding People-Centric Quality Attributes)*
- Prioritize information based on mission state, input received, and network state (tie to AIM agents effort?)*
- Provide decentralized bandwidth management for different types of files (NEC2 follow-on?)*
- Develop example applications, middleware, operating system services, and network mechanisms that change their quality-of-service as warfighter context (and needed information) changes* (see also 6.1.1)

6.1.2 Understanding Users and Their Contexts

- Create tools to model and predict whether the system that supports the CROP is matched well to the cognitive capabilities of its human elements
- Develop context-dependent runtime mechanisms to determine whether modeled expectations of warfighters are being met by the running system and, if not, how to rectify the situation
- Develop semantically aware task models, to take into account warfighter needs and state relevant to alternative forms of data presentation, visualization, aggregation, and filtering (see also 6.1.1, Context-Aware Assistive Computing)

Objective 4: Balanced Dynamic Resource Alloc. – 2

6.2.1 Algorithmic Mechanism Design

- Explore control-theoretic methods for handling rapidly changing demands and changing resource availability profiles; explore impact of service policies tuned for different system operating modes (see also 6.5.4, *In situ* Control and Adaptation, and 6.1.1, Context-Aware Assistive Computing)
- Given a lack of centralized control over individual behavior, design the CROP so info contributed to and extracted from it arises from (or is consistent with) the natural self-interests of individuals
- Apply auction mechanisms within the computational infrastructure to determine appropriate allocation of resources
- Demonstrate theoretical and empirical properties of different controller solutions in a prototype that reflects operational conditions (see also 6.5.4 and 6.1.1)

6.6.4 Enforcing Quality Attributes

- Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
- Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations (see also 6.5.3, Evolutionary Configuration and Deployment)
- Develop models/algorithms/tools that allow validating key functional properties before and during SW updates (see also 6.5.3, Evolutionary Configuration and Deployment)
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.1.4, Fostering Non-Competitive Social Collaboration)

Objective 5: Anticipate User Needs - 1

6.1.1 Context-Aware Assistive Computing

- Explore control-theoretic methods for handling rapidly changing demands and changing resource availability profiles; explore impact of service policies tuned for different system operating modes (see also 6.2.1, Understanding Users and Their Contexts, and 6.5.4, *In situ* Control and Adaptation)
- Create mechanisms such that when the system changes in ways that are visible to warfighters, either existing warfighter views are adapted to new system states or the effects on the warfighter are moderated (see also 6.5.4, *In situ* Control and Adaptation)
- Demonstrate theoretical and empirical properties of different controller solutions in a prototype that reflects operational conditions (see also 6.2.1, Understanding Users and Their Contexts, and 6.5.4, *In situ* Control and Adaptation)
- Develop methods for info dissemination based on geospatial location of warfighter*
- Develop capability to alert operators when portions of the mission plan are not executing correctly (related to running estimate serves part of NEC2)*
- Develop example applications, middleware, operating system services, and network mechanisms that change their quality-of-service as warfighter context (and needed information) changes* (see also 6.5.4)

Objective 5: Anticipate User Needs – 2

6.1.2 Understanding Users and Their Contexts

- Create tools to model and predict whether the system that supports the CROP is matched well to the cognitive capabilities of its human elements
- Develop semantically aware task models, to take into account warfighter needs and state relevant to alternative forms of data presentation, visualization, aggregation, and filtering (see also 6.1.2, Understanding Users and Their Contexts)
- Develop context-dependent runtime mechanisms to determine whether modeled expectations of warfighters are being met by the running system and, if not, how to rectify the situation

6.1.3 Modeling Users & User Communities

- Develop models that represent users and their communities; attach the models to system instrumentation and mechanisms allowing the system to adapt and reflect the model (see also 6.5.4, *In situ* Control and Adaptation)

6.6.3 Understanding People-Centric Quality Attributes

- Develop models and methods so warfighters have appropriate levels of trust (or mistrust) in the information being presented
- Integrate people-centric models that show how human performance and reliability contribute to overall system performance and reliability, and how human interactions, mediated by the system, affect overall mission success

Objective 6: Increase System Robustness

6.1.2 Understanding Users and Their Contexts

- Create tools to model and predict whether the system that supports the CROP is matched well to the cognitive capabilities of its human elements
- Develop context-dependent runtime mechanisms to determine whether modeled expectations of warfighters are being met by the running system and, if not, how to rectify the situation
- Develop semantically aware task models, to take into account warfighter needs and state relevant to alternative forms of data presentation, visualization, aggregation, and filtering (see also 6.1.1, Context-Aware Assistive Computing)

6.1.3 Modeling Users & User Communities

- Develop models that represent users and their communities; attach the models to system instrumentation and mechanisms allowing the system to adapt and reflect the model (see also 6.5.4, *In situ* Control and Adaptation)

6.6.4 Enforcing Quality Attributes

- Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
- Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations (see also 6.5.3, Evolutionary Configuration and Deployment)
- Develop models/algorithms/tools that allow validating key functional properties before and during SW updates (see also 6.5.3, Evolutionary Configuration and Deployment)
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.1.4, Fostering Non-Competitive Social Collaboration)

ULS Systems Research Topics *In*/Not In Roadmap

6.1.1 Context-Aware Assistive Computing

6.1.2 Understanding Users and Their Contexts

6.1.3 Modeling Users and User Communities

6.1.4 Fostering Non-Competitive Social Collaboration

6.1.5 Longevity

6.2.1 Algorithmic Mechanism Design

6.2.2 Metaheuristics in Software Engineering

6.2.3 Digital Evolution

6.3.1 Design of All Levels

6.3.2 Design Spaces and Design Rules

6.3.3 Harnessing Economics to Promote Good Design

6.3.4 Design Representation/Analysis

6.3.5 Assimilation

6.3.6 Determining and Managing Requirements

6.4.1 Expressive Representation Languages

6.4.2 Scaled-Up Specification, Verification, and Certification

6.4.3 Computational Engineering for Analysis and Design

6.5.1 Decentralized Production Management

6.5.2 View-Based Evolution

6.5.3 Evolutionary Configuration and Deployment

6.5.4 In Situ Control and Adaptation

6.6.1 Decentralized Production Management

6.6.2 Scale and Composition of Quality Attributes

6.6.3 Understanding People-Centric Qual. Attr.

6.6.4 Enforcing Quality Requirements

6.6.5 Security, Trust, and Resiliency

6.6.6 Engineering Management at Ultra-Large Scales

6.7.1 Policy Definition for ULS Systems

6.7.2 Fast Acquisition for ULS Systems

6.7.3 Management of ULS Systems

6.1.1 Context-Aware Assistive Computing

Context-Aware Assistive Computing (CAAC) research enables systems to provide people with the right information and control capabilities at the right time, based on an understanding of user context, i.e., the tasks a user is trying to perform

Why does this research help the CROP?

- Providing only relevant information to warfighters is an essential element of the CROP capability
- Warfighters in different echelons have to adapt to changing circumstances, and the relevant information in the COP changes as their tasks change or as they try different ways of accomplishing their tasks
- CAAC provides the means for anticipating user needs, making the relevant information available more easily and quickly and determining how the presented types of information should change based on the warfighter's perceived situation
- CAAC enables the system to learn what is relevant to particular classes of warfighters so it can help warfighters find the relevant information

6.1.1 Context-Aware Assistive Computing (1)

6.1 Research

- Explore control-theoretic methods for handling rapidly changing demands and changing resource availability profiles; explore impact of service policies tuned for different system operating modes (see also 6.2.1 and 6.5.4)
- Develop semantically aware task models, to take into account warfighter needs and state relevant to alternative forms of data presentation, visualization, aggregation, and filtering (see also 6.1.2)
- Create mechanisms such that when the system changes in ways that are visible to warfighters, either existing warfighter views are adapted to new system states or the effects on the warfighter are moderated (see also 6.5.4)

6.1.1 Context-Aware Assistive Computing (2)

6.2 Research

- Demonstrate theoretical and empirical properties of different controller solutions in a prototype that reflects operational conditions (see also 6.2.1 and 6.5.4)
- Develop methods for info dissemination based on geospatial location of warfighter*
- Develop capability to alert operators when portions of the mission plan are not executing correctly (related to running estimate serves part of NEC2)*
- Develop example applications, middleware, operating system services, and network mechanisms that change their quality-of-service as warfighter context (and needed information) changes* (see also 6.1.1)

6.1.2 Understanding Users and Their Contexts

Understanding Users and Their Contexts requires research aimed at understanding the drivers of human behavior in the context of system operation

Why does this research help the CROP?

- A commander's goals cannot be achieved without an understanding of the role of humans in the control loop. For example, a battle management system will be more effective if it has been designed with an appreciation of what humans can/cannot do best in contributing to and interpreting the operational picture presented by the system

6.1.2 Understanding Users and Their Contexts

6.1 Research

- Create tools to model and predict whether the system that supports the CROP is matched well to the cognitive capabilities of its human elements
- Develop context-dependent runtime mechanisms to determine whether modeled expectations of warfighters are being met by the running system and, if not, how to rectify the situation
- Develop semantically aware task models, to take into account warfighter needs and state relevant to alternative forms of data presentation, visualization, aggregation, and filtering (see also 6.1.1)

6.1.3 Modeling Users and User Communities

Research focused on **Modeling Users and User Communities** uses field analyses of interactions among user communities and the computational elements of the system to develop models of how ULS systems are actually used and evolved

Why does this research help the CROP?

- A socio-technical ecosystem supporting a CROP is as much about its user communities as its technology. Although existing systems occasionally contain user models, they do not contain explicit models of groups or communities of users and their behaviors. Research is needed to make the systems serve such communities more effectively

6.1.3 Modeling Users & User Communities

6.1 Research

- Develop models that represent users and their communities; attach the models to system instrumentation and mechanisms allowing the system to adapt and reflect the model (see also 6.5.4)

6.1.4 Fostering Non-Competitive Collaboration

Research in the area of **Fostering Non-Competitive Social Collaboration** builds upon the successes of cooperative development models (e.g., open source and open architecture models) to meet the goals of a continuously evolving system at large scale, while maintaining guarantees of reliability, security, performance, etc.

Why does this research help the CROP?

- Many good ideas for improvements to the system supporting the CROP come first from individuals and then from groups. To adapt the system to these new needs, the system should be able to take advantage of the productive capability of these users to enable local adaptations that gradually spread to other users of the CROP
- When the CROP needs to be quickly adapted to new circumstances, it should be possible to allow the adaptations to be created by voluntarily assembled groups whose outputs yield a functionally improved, yet still reliable and maintainable CROP

6.1.4 Fostering Non-Competitive Social Collab.

6.1 Research

- Define and test incentive structures for their ability to guide non-competitive development processes
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.6.4, Enforcing Quality Attributes)

6.2.1 Algorithmic Mechanism Design

Algorithmic mechanism design provides interaction rules and incentives such that the actions of self-interested, but rational individuals is more likely to create a desired global outcome, even in an environment of decentralized control

Why does this research help the CROP?

- The clarity of the CROP could be enhanced by the use of market mechanisms to ensure that the right quality and amount of information is shared appropriately
- Appropriate mechanism design could help govern the use of limited system resources such as bandwidth by exploiting the fact that transmitted information is of greater value to some participants than others

6.2.1 Algorithmic Mechanism Design

6.1 Research

- Explore control-theoretic methods for handling rapidly changing demands and changing resource availability profiles; explore impact of service policies tuned for different system operating modes (see also 6.5.4, *In situ* Control and Adaptation, and 6.1.1, Context-Aware Assistive Computing)
- Given a lack of centralized control over individual behavior, design the CROP so info contributed to and extracted from it arises from (or is consistent with) the natural self-interests of individuals
- Apply auction mechanisms within the computational infrastructure to determine appropriate allocation of resources

6.2 Research

- Demonstrate theoretical and empirical properties of different controller solutions in a prototype that reflects operational conditions (see also 6.1.1 and 6.5.4)

6.5.3 Evolutionary Configuration & Deployment

Evolutionary configuration and deployment technologies enable

- Developers and end-users to modify existing systems with new (multiple) versions of components
- Different and evolving configurations to run concurrently in the same operational ULS system
- Trustworthy distribution of software releases and updates

Why does this research help the CROP?

- Evolutionary configuration and deployment technologies help CROP system operators and developers dependably and rapidly modify and extend components contributing to the CROP capability in response to changed technologies and improved understanding of CROP needs and what is possible

6.5.3 Evolutionary Configuration and Deployment

6.1 Research

- Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations (see also 6.6.4, Enforcing Quality Attributes)
- Develop models/algorithms/tools that allow validating key functional properties before and during SW updates (see also 6.6.4, Enforcing Quality Attributes)
- Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces (see also 6.5.4, *In situ* Control and Adaptation)*

6.2 Research

- Provide ability for users to build their own, localized ontologies (perhaps as part of Army's Development Network)*
- Implement SOA so users can build their own services and so service priorities/performance can be tracked across the enterprise*

6.5.4 *In situ* Control and Adaptation

***In situ* control and adaptation** research provides theory and methods to support adaptive realignment of resources in large-scale systems

- "*in situ*" refers to the ability of the system to adapt on-the-fly rather than by dependence on external intervention, e.g., by having system changes made as part of a maintenance or upgrade process

Why does this research help the CROP?

- *In situ* control and adaptation technologies compensate for intermittent deficiencies in the operation of the system and as well as changes in the required QoS by taking advantage of alternative capabilities to provide continued services to the warfighter

6.5.4 *In situ* Control and Adaptation (1)

6.1 Research

- Explore control-theoretic methods for handling rapidly changing demands and changing resource availability profiles; explore impact of service policies tuned for different system operating modes (see also 6.1.1 and 6.2.1)
- Create mechanisms such that when the system changes in ways that are visible to warfighters, either existing warfighter views are adapted to new system states or the effects on the warfighter are moderated (see also 6.1.1)
- Develop models that represent users and their communities; attach the models to system instrumentation and mechanisms allowing the system to adapt and reflect the model (6.1.3)
- Determine how to allow user-defined capability to dynamically interact with existing ontologies and user interfaces (6.6.3)*

6.5.4 *In situ* Control and Adaptation (2)

6.2 Research

- Demonstrate theoretical and empirical properties of different controller solutions in a prototype that reflects operational conditions (see also 6.1.1 and 6.2.1)
- Prioritize information based on mission state, input received, and network state (tie to AIM agents effort?)*
- Provide decentralized bandwidth management for different types of files (NEC2 potential follow-on)*
- Develop example applications, middleware, operating system services, and network mechanisms that change their quality-of-service as warfighter context (and needed information) changes* (see also 6.1.1)

6.6.3 Understanding People-Centric Qual. Attr.

Research on **Understanding People-Centric Quality Attributes** addresses how the human element affects system quality attributes such as performance, reliability, safety, etc.

Why does this research help the CROP?

- Since people are integral parts of the system, group performance, reliability, and security will affect system performance, reliability, and security. Without adequate models of group behavior in the context of the system, it will be difficult to anticipate and adapt to the consequences of such behavior
- The data that forms the operating picture comes from people as well as from sensors. This research will help in modeling how different human-supplied data varies in different situations so the system can present the most trustworthy picture of the situation.

6.6.3 Understanding People-Centric Quality Attr.

6.1 Research

- Develop models and methods so warfighters have appropriate levels of trust (or mistrust) in the information being presented
- Integrate people-centric models that show how human performance and reliability contribute to overall system performance and reliability, and how human interactions, mediated by the system, affect overall mission success

6.6.4 Enforcing Quality Attributes

Research to **Enforce Quality Attributes** provides

- ways of maintaining desired levels of reliability, performance, security, etc. in the face of system modifications and normal failures
- ways to satisfy new and possibly situation-specific quality requirements

Why does this research help the CROP?

- Sensor-rich systems on ad hoc, dynamically (re)configured networks, in hostile and fast-changing environments will have demanding but variable quality requirements (time, security, etc.)
  - For example, unanticipated track volume may violate assumptions underlying prior, predictable system performance.
  - Quality attribute enforcement dynamically adds resources to preserve assumptions as invariants, or adapts the analytic models to a changing reality

6.6.4 Enforcing Quality Attributes

6.1 Research

- Develop quality attribute enforcement protocols, their associated quality attribute theories, and the means for dynamic (online) adaptation of both
- Investigate certification techniques that can ensure adaptive systems only operate within safe, correct, and stable configurations (see also 6.5.3, Evolutionary Configuration and Deployment)
- Develop models/algorithms/tools that allow validating key functional properties before and during SW updates (see also 6.5.3, Evolutionary Configuration and Deployment)
- Produce a variety of non-competitive development models relevant to the types of adaptations needed for the CROP while ensuring that these development models can achieve the level of needed quality (see also 6.1.4, Fostering Non-Competitive Social Collaboration)

Roadmap Structure and Development Process

Start with: a needed ULS system warfighter **capability**

Make: **Observations** about this capability

- Example: user needs change dynamically

Use: ULS systems **perspective** (contrasted with conventional approach)

Identify: **Technical challenge** (related to ULS systems perspective)

- Contrast with the “usual” technical challenge

Restate challenge as: **Research objective**

Cite: ULS Systems report **Research Topic**

Define **Research Initiatives**: Several supporting each research objective

Roadmap Intent

Motivate Research

- The roadmap shows how an individual research initiative (a 3-4 year effort of $1M/year) supports one or more ULS-system technical challenges

Help evaluate the ULS systems relevance of existing or planned research

- The roadmap structure explicitly shows a ULS system perspective

Prioritize research funding

- The roadmap provides a basis for determining which research is most critical/relevant/impactful for achieving a future ULS systems capability

Framework for incorporating additional ULS systems research

Software Engineering Institute | Carnegie Mellon