Research Directions in Requirements Engineering

Betty H.C. Cheng  
Dept of Computer Science and Engineering  
Michigan State University  
East Lansing, Michigan 48824 USA  
chengb@cse.msu.edu  
http://www.cse.msu.edu/~chengb

Jo Atlee  
David R. Cheriton School of Computer Science  
University of Waterloo  
Waterloo, Ontario N2L 3G1 CANADA  
jmatlee@uwaterloo.ca  
http://se.uwaterloo.ca/~jmatlee
Outline

• What is RE?
  – Why is RE difficult?
• RE State of the Art
• Research Strategies
• RE Research Hotspots
• Recommendations
What is Requirements Engineering?

Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems.

It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.

[Zave 83]
Why is RE Difficult?

• RE faces complementary challenges from those faced by the rest of SE community

• Different types of artifacts
  – Downstream SE works in solution space
  – RE works in problem space
### Why is RE Difficult?

<table>
<thead>
<tr>
<th>Issues</th>
<th>Downstream SE</th>
<th>RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Solution space</td>
<td>Problem space</td>
</tr>
<tr>
<td>Starting point</td>
<td>Specified by requirements</td>
<td></td>
</tr>
<tr>
<td>Alternatives to consider</td>
<td>Constrained by requirements, platform, acceptance, priorities, system boundaries</td>
<td></td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Technical, more homogeneous</td>
<td></td>
</tr>
<tr>
<td>Artifacts</td>
<td>Models of SW</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td>Majority of resources</td>
<td></td>
</tr>
<tr>
<td>Strategy</td>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Paradigm Shift</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leverage other disciplines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leverage technology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evolutionary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain-specific</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generalization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Shaw02, Redwine and Riddle 85, Basili93]
Research Strategies
Activity Level

Volume of Activity

Paradigm
Leverage Disciplines
Leverage Technology
Leverage Disciplines
Evolutionary
Domain-specific
Generalization
Engineering
Evaluation

INCREASING MATURITY
## Matrix of State of the Art

<table>
<thead>
<tr>
<th>Research Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elicitation</td>
</tr>
<tr>
<td>Modeling</td>
</tr>
<tr>
<td>Reqts</td>
</tr>
<tr>
<td>Analysis</td>
</tr>
<tr>
<td>V&amp;V</td>
</tr>
<tr>
<td>Reqs Mgmt</td>
</tr>
</tbody>
</table>

### Elicitation
- Activities that enable the understanding of the goals, objectives, and motives for building a system.

### Modeling
- Abstract representations of system’s goals, objectives, motives, and requirements.

### Reqts
- Validation: ensures models and documentation capture stakeholders’ needs.
- Verification: rigorous assessment that requirements satisfy specific properties.

### Analysis
- Managing requirements during construction and evolution:
  - Identifying and documenting traceability links
  - Integration of requirements of different ages

---

*ICSE07 Future of Software Engineering, Future Directions of Requirements Engineering, BHC Cheng, JA Atlee, copyrighted 2007*
# Matrix of State of the Art

<table>
<thead>
<tr>
<th></th>
<th>Elicitation</th>
<th>Notations</th>
<th>Methodologies, Patterns, Strategies</th>
<th>Analysis, Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain descriptions [12]</td>
<td>Patterns [56,87,97,169]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Object models [86]</td>
<td>Modeling facilitators [7,34,95,126]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Formalization heuristics [20,68]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Methodologies [16]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hot Spots

• **Definition**: areas with pressing needs given anticipated challenges posed by emerging systems (e.g., scale, security, tolerance, tighter integration between system and environment)

• **How we selected specific hot spots:**
  – Extensions to the State of the Art:
    • Based on “gaps” in matrix and research strategies
    • Internal factors (push on boundaries)
  – Defining New State of the Art:
    • External factors (from stakeholders from industry, govt, users)
    • Emerging systems
    • Proactive (part of problem solving team)
      – identify promising solutions
      – define the leading edge (or be reactive)
Hot Spots Impact

<table>
<thead>
<tr>
<th>Problems</th>
<th>Requirements Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Notations</td>
</tr>
<tr>
<td>Elicitation</td>
<td>Goals [21, 170]</td>
</tr>
<tr>
<td></td>
<td>Scenarios [1, 35, 49]</td>
</tr>
<tr>
<td></td>
<td>Agents [106, 180]</td>
</tr>
<tr>
<td></td>
<td>Anti-patterns [155, 164, 171]</td>
</tr>
<tr>
<td></td>
<td>Non-functional requirements [31, 79]</td>
</tr>
</tbody>
</table>

|               | Logistics [52]             | Patterns [56, 87, 97, 169]           | Model composition [79] |

| Validation & Verification | Property languages [15, 105] | Simulation [162] | Simulation [162] |
|                          | Model formalisms [24, 53]    | Invariant generation [91] | Invariant generation [91] |
|                          | Object models [86]          | Consistency checking [60, 81, 120] | Invariant generation [91] |
|                          | Sjkfoawienwoania             | Inspection [62, 129] | Consistency checking [60, 81, 120] |
|                          | Akejhvnlaweuvhb               | Model checking [29, 57, 157] | Inspection [62, 129] |

|                        | Feature management [176]       |                          | Impact analysis [101] |

Extensions to State of Art:
- Methodologies, Patterns, Tools
- Reuse
- Effectiveness of RE Techniques

New State of the Art
Matrix of State of the Art

<table>
<thead>
<tr>
<th>RE Problems Key here:</th>
<th>Requirements Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
<td>Methodologies, patterns, strategies</td>
</tr>
<tr>
<td>RE problem</td>
<td>Human strategies mature; evolutionary for creating new notations for elicited reqts; ever-increasing (higher) level of abstraction (reqts-&gt; goals-&gt; policies)</td>
</tr>
<tr>
<td>Elicitation</td>
<td>Leverage technology video con</td>
</tr>
<tr>
<td>Modeling</td>
<td>Evolutionary; specialization (e.g., viz: Drawing HCI, graphics)</td>
</tr>
<tr>
<td>Requirements</td>
<td>Leverage other</td>
</tr>
</tbody>
</table>

**Paradigm Shift (pushing product-line engr to RE)**

- **Requirements Management**
  - Evolutionary; generalization (model transformations/composition --> understand composition.)
  - Leveraging technology for RE purposes (Logics, formalization mappings; model notation for V&V--not intended for public consumption); aggressive abstraction
- **Validation**
  - Blow up: lots of activity (rainbow: spectrum of types of research)
- **Verification**
  - Leverage other disciplines
- **Elicitation**
  - Evolutionary; specialization (e.g., viz: Drawing HCI, graphics)
Extensions to State of Art

• Methodologies, Patterns, Tools
  – Need more engineering-oriented research to
    • integrate current techniques and
    • Make more widely available, applicable
  – Requirements Reuse (e.g., patterns)
    • Within a domain
    • Across domains

• Effectiveness of RE Technologies
  – Need more evaluation-oriented research
    • Demonstrate utility of current RE techniques on industrial-strength problems
  – Also need comparisons of similar techniques
    • Provide guidance as to when one approach is more appropriate
New State of the Art

• Motivated by anticipated needs
  – External factors (from stakeholders from industry, govt, users)
  – Emerging systems

• Research Challenges:
  – Scale
  – Tolerance
  – Self-Management
  – Cyberphysical
  – Security
New Scale

• **Definition**: new orders of magnitude increase in scale [*ULS Report, SEI 2006*]
  – Size
  – Heterogeneity
  – Decentralized elements
  – Complexity (decision logic)

• **Examples**: (of the future)
New Scale

Example:

Intelligent Transportation and Vehicle Systems
New Scale

Ultra-Large Scale SW-Intensive Systems
New Scale

• **Definition:** new orders of magnitude increase in scale
  – Size (e.g., thousands of sensors, platforms)
  – Heterogeneity
  – Decentralized elements
  – Complexity (decision logic)

• **Example:** ITS and IVS

• **RE Challenges:**
  – Modeling, abstraction, analysis techniques to handle new notions of large scale.
  – Managing requirements with uncertainty in data, processing, platforms
  – Detecting/resolving feature interactions
Self-Managing Systems

• **Definition**: software-based system is aware of its context and must react and adapt to changes in the environment or requirements
  
  [Kramer and Magee, FoSE07]

• **Examples:**
Self-Managing Systems

• **Definition**: software-based system is aware of its context and must react and adapt to changes in the environment or requirements [Kramer and Magee, FoSE07]

• **Examples**: 

• **RE Challenges**: 
  – Determining when adaptation is needed 
  – Supporting changing requirements 
  – Gaining assurance for adaptive systems 
    • Identify assurance criteria 
    • Reasoning technology for adaptive systems 
  – Run-time monitoring of system and environment wrt current requirements
Tolerance

• Definition of sufficient correctness
  – “The degree to which a system must be dependable in order to serve the purpose its user intends, and to do so well enough to satisfy the current needs and expectations of those users” [Shaw WOSS02].

• RE Challenges
  – Size of systems will make it impractical to have complete, consistent, and stable requirements -- need to settle for “healthful systems”
  – Requirements for fault tolerance (cannot wait to address FT at design/implementation)
  – Negative requirements (e.g., unhealthy conditions to avoid)
  – Requirements for diagnostic and recovery mechanisms
Cyberphysical

• **Definition:** Software seamlessly integrated with environment. *[Schaefer and Wehrheim, FoSE07]*
  – Computing and communication tightly coupled with monitoring and control of physical entities in environment

• **Examples:**
  
  **Automated Manufacturing**

  **Handheld/Wearable Computing**
Cyberphysical

- **Definition**: Software seamlessly integrated with environment. [Schaefer and Wehrheim, FoSE07]
  - Computing and communication tightly coupled with monitoring and control of physical entities in environment

- **Examples**:

- **RE Challenges**:
  - Need to model environment (not discrete)
  - Hybrid models (continuous and discrete systems interacting)
  - Unpredictable environmental factors (e.g., humans)
  - Uncertainty aspects:
    - What (when) to monitor,
    - Reacting to dated conditions (delayed information)
    - Noise in sensor data
    - Interpreting the data (SW state or environmental conditions)
Security

• **Definition:**
  – “A computer is **secure** if you can depend on it and its software to behave as you expect (intend).”
  – ‘**Trust** describes our level of confidence that a computer system will behave as expected.’ (intended)  

[Garfinkel & Spafford, Kasten]

• **Examples:**
  – Healthcare Infrastructure
  – Intelligent Transportation Systems
ULS of the Future

Numerous Security Concerns

Healthcare Infrastructure
Numerous security concerns

Intelligent Transportation and Vehicle Systems
Security

• **Definition:**
  – “A computer is **secure** if you can depend on it and its software to behave as you expect (intend).”
  – ‘**Trust** describes our level of confidence that a computer system will behave as expected.’ (intended)

[Garfinkel & Spafford, Kasten]

• **Examples:**

• **RE Challenges:**
  – Modeling the environment
    • Need for complete (threat) models
  – Malicious entities exist
  – Notations and methodologies
    • Structuring, modeling, and reasoning about security policies
  – Monitoring for security requirements adherence
  – Tools for modeling and analyzing security requirements
Globalization

• **Definition**: globally distributed development teams [Herbsleb, FoSE07]

• **Motivation**:
  - Exploit a 24-hour work day
  - Global work force
Globalization

- **Definition**: globally distributed development [Herbsleb, FoSE07]
- **Motivation**:
  - Exploit a 24-hour work day
  - Global work force
- **RE Challenges**:
  - RE documentation must support distributed downstream development activities (design, coding, testing)
  - Communication support in a distributed environment:
    - Requirements elicitation, modeling, negotiation
    - Team management (geographical, time zone, culture, language)
Recommendations

**RESEARCHERS**

*Extensions to State of Art*
- Engineering/integrating RE Techniques
- Apply to industrial-strength data
- Comparative studies
- Integrate with downstream SE research
- Training the next generation

*Defining New State of Art*
- Looking for paradigm shifts for emerging systems and challenges
- New modeling, analysis, abstraction techniques

**INDUSTRY/PRACTITIONERS**

External factors pose new challenges
- New computing needs
- New application domain needs
- New users
- Leverage new technology

*New Considerations*
- Scale
- Assurance
- Distributed Control
- Autonomic Behavior
Recommendations

Researchers

*Extensions to State of Art*
- Engineering/integrating RE Techniques
- Apply to industrial-strength data
- Comparative studies
- Integrate with downstream SE research
- Training the next generation

*Defining New State of Art*
- Looking for paradigm shifts for emerging systems and challenges
- New modeling, analysis, abstraction techniques

Collaborative Partnerships

- Collaborative research efforts
- Collective identification and definition of RE research challenges
- Sanitized industrial-strength data
- Repositories of RE-related artifacts
- Evaluation of research techniques

[Whitehead, FoSE07, Rombach & Achatz, FoSE07]

Industry and Practitioners

- External factors pose new challenges
- New computing needs
- New application domain needs
- New users
- Leverage new technology

New Considerations

- Scale
- Assurance
- Distributed Control
- Autonomic Behavior

New Considerations

- Collaborative research efforts
- Collective identification and definition of RE research challenges
- Sanitized industrial-strength data
- Repositories of RE-related artifacts
- Evaluation of research techniques

[Whitehead, FoSE07, Rombach & Achatz, FoSE07]
Acknowledgements

• Feedback:
  – Philip K. McKinley, Axel van Lamsweerde, Bashar Nuseibeh, Robyn Lutz, Steve Fickas, Brian Berenbach, Heather Goldsby, Dan Fiedler, Lionel Briand

• Sponsors:
  – U.S. Department of the Navy, Office of Naval Research under Grant No. N00014-01-1-0744, and in part by National Science Foundation grants CCR-9901017, EIA-0000433, EIA-0130724, CCF-0541131, and CNS-0551622, and by Siemens Cooperate Research, and a Michigan State University Quality Fund Concept Grant
  – Natural Sciences and Engineering Research Council of Canada
Take home message...

**Researchers**

*Extensions to State of Art*
- Engineering/integrating RE Techniques
- Apply to industrial-strength data
- Comparative studies
- Integrate with downstream SE research
- Training the next generation

*Defining New State of Art*
- Looking for paradigm shifts for emerging systems and challenges
- New modeling, analysis, abstraction techniques

**Collaborative Partnerships**
- Collaborative research efforts
- Collective identification and definition of RE research challenges
- Sanitized industrial-strength data
- Repositories of RE-related artifacts
- Evaluation of research techniques

**Industry and Practitioners**

External factors pose new challenges
- New computing needs
- New application domain needs
- New users
- Leverage new technology

**New Considerations**
- Scale
- Assurance
- Distributed Control
- Autonomic Behavior

*Whitehead, FoSE07, Rombach & Achatz, FoSE07*

---

ICSE07 Future of Software Engineering, Future Directions of Requirements Engineering, BHC Cheng, JA Atlee, copyrighted 2007