Design Informatics
- Report Out -

Moon-Jung Chung, Steven Fenves, S. K. Gupta, Kincho Law, Larry Leifer, Joe Kopena, Andrew Kusiak, Bill Regli (lead), Rob Stone
Discussion Focal Points

• What is cyber-infrastructure?
• Who, how and what will use it?
• Identify cyber-infrastructure needs
  – Driven by engineering design domain
  – Grounded and specific (as possible)
  – Not redundant with existing or generic cyber-infrastructure goals
    • This is an important point
What is Cyber-Infrastructure?

• Network
  – The communications & computing medium
  – Note: this is not specific to design

• Content
  – Data, knowledge, integration, software…
  – Note: in the context of engineering design

• Content creation and management tools
  – Authoring, browsing, discovery, archiving, searching, etc…
  – Note: in the context of engineering design
What is Design Informatics?

• Sure, it’s data, knowledge, ontologies, repositories, etc….
  – Of what?
  – For who?
  – For what purpose?
• It’s also about
  – tools to create, build and manage design repositories, ontologies, etc that address specific engineering needs
  – Instrumentation and logging of repositories so we know how/why they are used
Who are the users of Engineering Cyber-Infrastructure?

- Customers
- Designers
- Tool Builders
Which topics to study?

• There is no one “Repository” with everything
• Probably need 30-40? repositories, each with different focus and architecture and audience
  – These will more manageably partition and explore the problem spaces
  – Makes population more manageable too

• Human-Centric focus
  – The cyber-infrastructure will connect the products, processes, people (customers, designers, tool builders)
Scenario 1: Rationale Capture & Reuse

- Problem: getting at tacit data, background activities, design process capture and learning
- Objective: Rationale Repositories
  - Proactive capture tools, data representations
  - Tools for mapping multi-media to extracted events of engineering interests
  - Connecting NLP, video processing, etc to engineering problems and systems
  - Intelligent filtering tools to handle information overload
    - Human attention is the most precious commodity
Scenario 2: Trust & Security

• Problem: Given that we can capture everything and make it all available, how to control access and views?

• Objective: Tools & Repositories to study Cyber-Trust in engineering contexts
  – Tools, systems, corpa
  – Information hiding, sanitization, obfuscation, abstraction, protection
  – Transparency of process and data
  – Multi-level, cross-domain sharing of engineering knowledge
Scenario 3: Interface & Human Factors

• Problem: Customers will interface with new, customized, cyber-aware products at all stages of their lifecycle

• Objective: Interface and Interaction Repositories
  – Support for communication, collaboration, interaction specifically in engineering contexts
    • Across customers, designers, tool builders
  – How to interface different users of cyber-infrastructure with the content, tools and products?
  – General techniques, evaluation methodologies, metrics, etc
Scenario 4: Knowledge Evolution & Complexity

- Problem: how to capture temporal change, map across engineering ontologies, handle uncertainty, etc?
- Objective: Knowledge Repositories
  - Learn/use logic, Semantic Web, etc
  - Knowledge representations to handle complexity, uncertainty, etc
  - Create instances of products out of well-defined product classes
  - Bridge AI & Semantic Web with engineering
Scenario 5: Deep Queries to Support Design Decisions

• Problem: engineering data/knowledge is buried video, CAD, telemetry from products, audio, databases, etc

• Objective: Repositories to support development of deep engineering queries
  – Partial solutions, probabilistic states, connect across media and representations, knowledge fusion, extraction from semi-structured & unstructured engineering data
  – DB meets engineering
Scenario 6: Software Repositories

- Problem: Choreograph a simulation & lifecycle test for a new, customized, cyber-aware product
- Objective: Service-Oriented Architectures (SOA) for engineering software components
  - Software for tool builders, designers and customers; as well as software embedded in cyber-aware products
  - We need ontologies that leverage and extend web services and semantic web standards
  - Self-integrating, adaptable software for engineering contexts
  - Repositories of code, modules, interfaces
  - Web-accessible agents and services
Where will we get data for repositories & ontologies, etc

- Top down
  - Arecibo, NASA Pathfinder, collider, etc
  - Can NSF create a challenge program with another agency or organization?

- Bottom up
  - Pedagogical design projects, Lego robot classes, etc

- Can’t build it all at once
- Can we identify tractable sub-problems and problem classes?
- How to engage industry to help build the engineering cyber-infrastructure?
National Benefits

- Support faster innovation
- Improved productivity & time to market
- Rapid reaction to market changes
- Delivery of individually customize products
- Increased customer participation in the product lifecycle
- Greater workforce competitiveness
  – Toward knowledge-centric engineering