Carnegie Mellon

Talking About Concerns . . .

James D. Herbsleb
School of Computer Science
Carnegie Mellon University

Carnegie Mellon

What is Modularity?

- Thanks, Mary!
- Thanks, Dick!

institute for SOFTWARE RESEARCH

Carnegie Mellon

Why Modularity?

- **Software** modularity does not matter
- . . . at all
- Except . . .
  - To the extent it modularizes **work**
- **Work** modularity aids human understanding
- **Work** modularity simplifies coordinating people and teams

Carnegie Mellon

Parnas:

Expected Benefits of Modularity

- Reduce need for coordination
  - “separate groups would work on each module with little need for communication”
- Simplify comprehension
  - “it should be possible to study the system one module at a time”
- These effects lower the cost of change
  - “it should be possible to make drastic changes to one module without a need to change others”

Parnas, D. L. On the Criteria to be Used in Decomposing Systems into Modules. *Communications of the ACM*, 15, 12 (1972), 1053-1058, p. 1054.

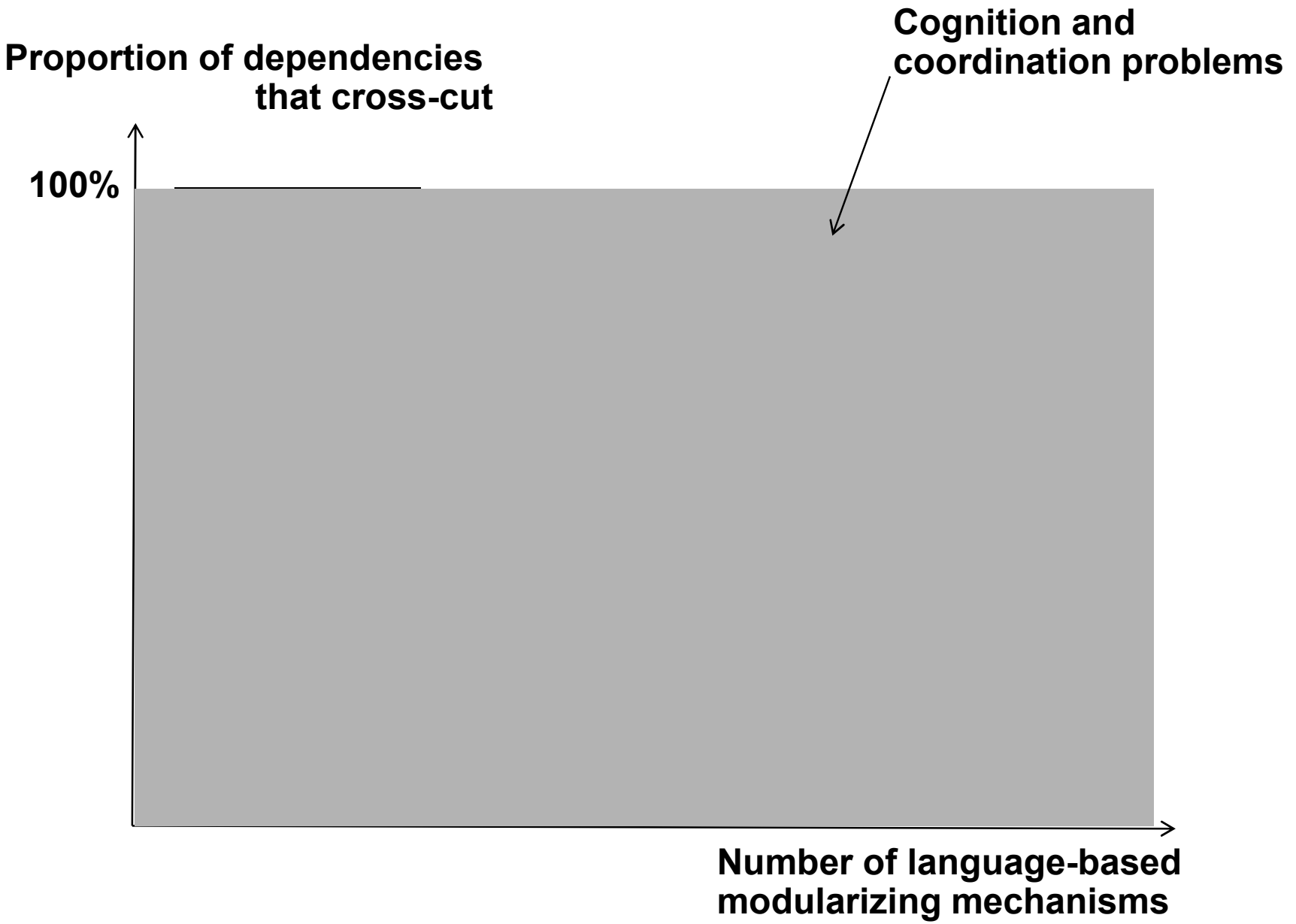
isr institute for SOFTWARE RESEARCH

Carnegie Mellon

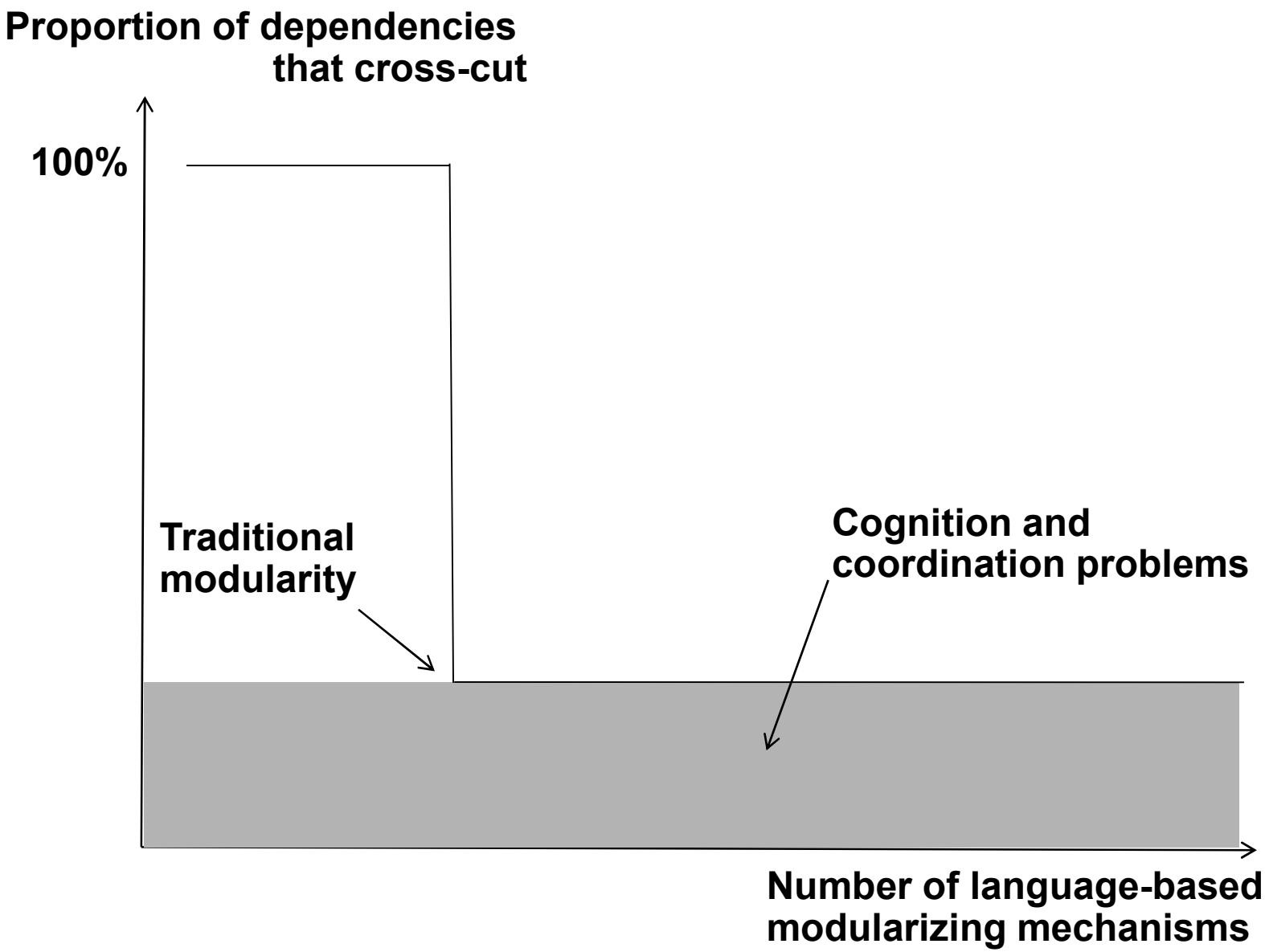
Vision . . .

- “a vivid mental image; ‘he had a vision of his own death’” *
- “an Explanation of Life Founded upon the Writings of Giraldus and upon Certain Doctrines Attributed to Kusta Ben Luka” *
- “a thought, concept, or object formed by the imagination” **
- “direct mystical awareness of the supernatural“ **

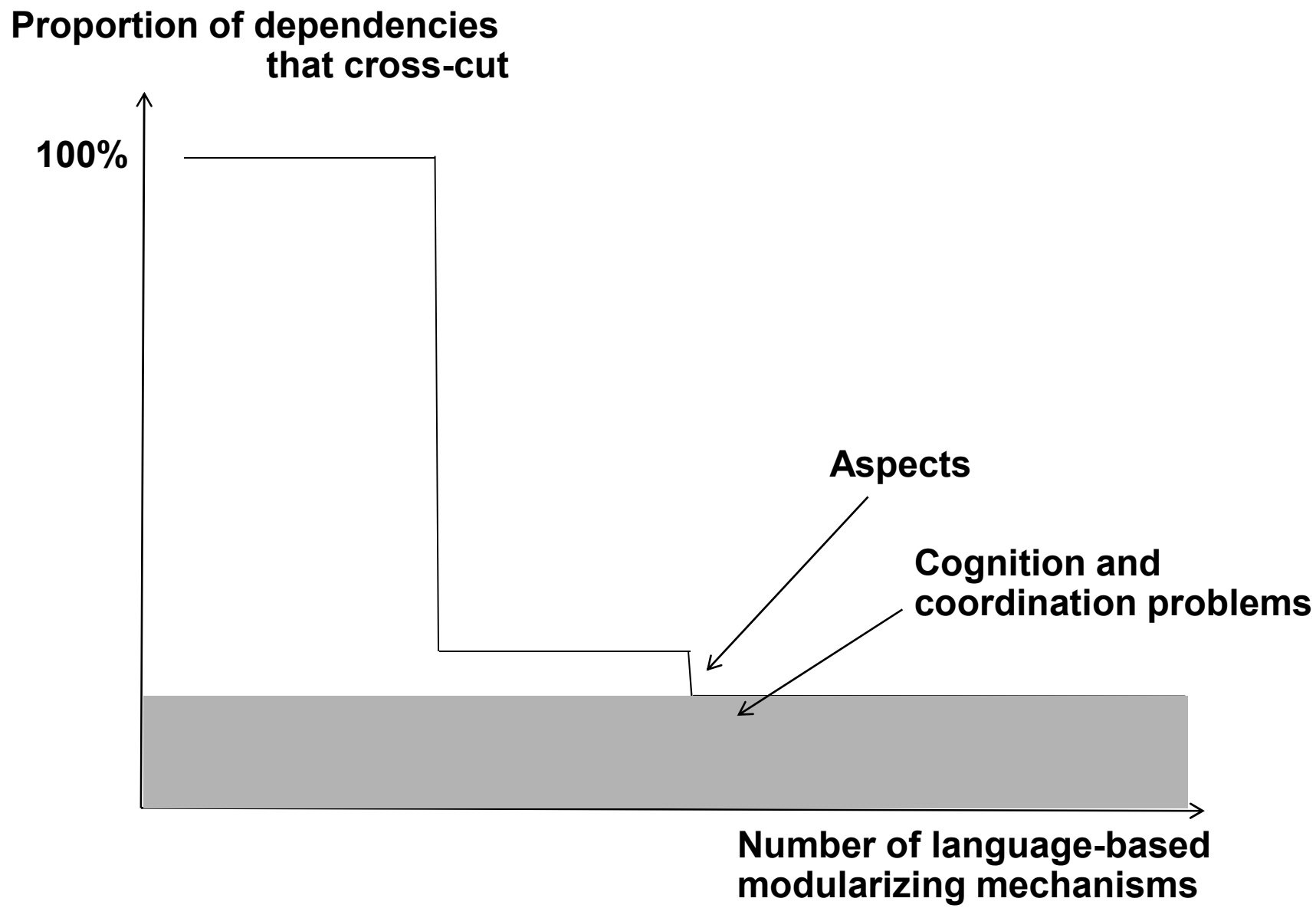
*wordnetweb.princeton.edu/perl/webwn
**Merriam-Webster Dictionary



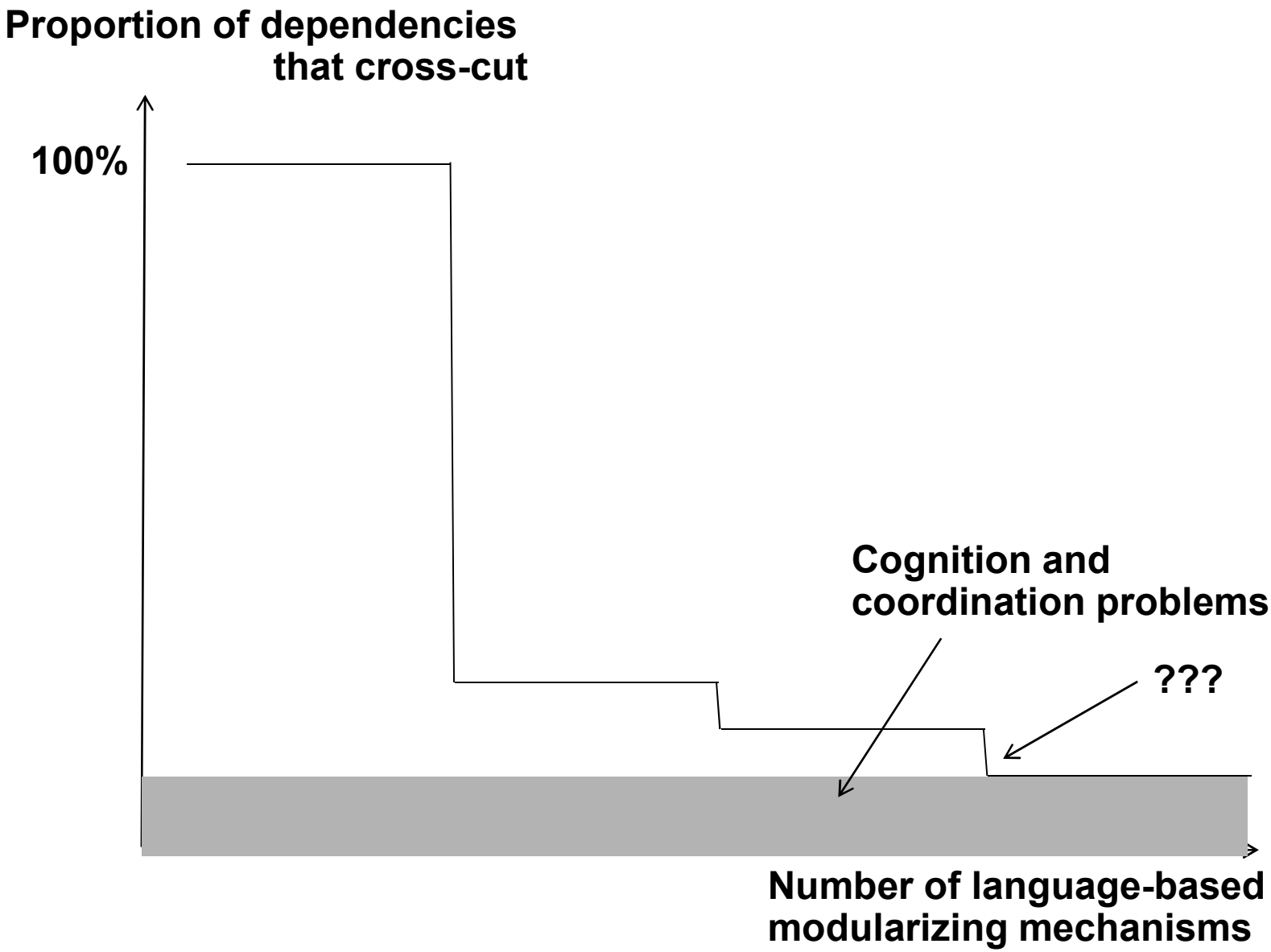
Carnegie Mellon
School of Computer Science



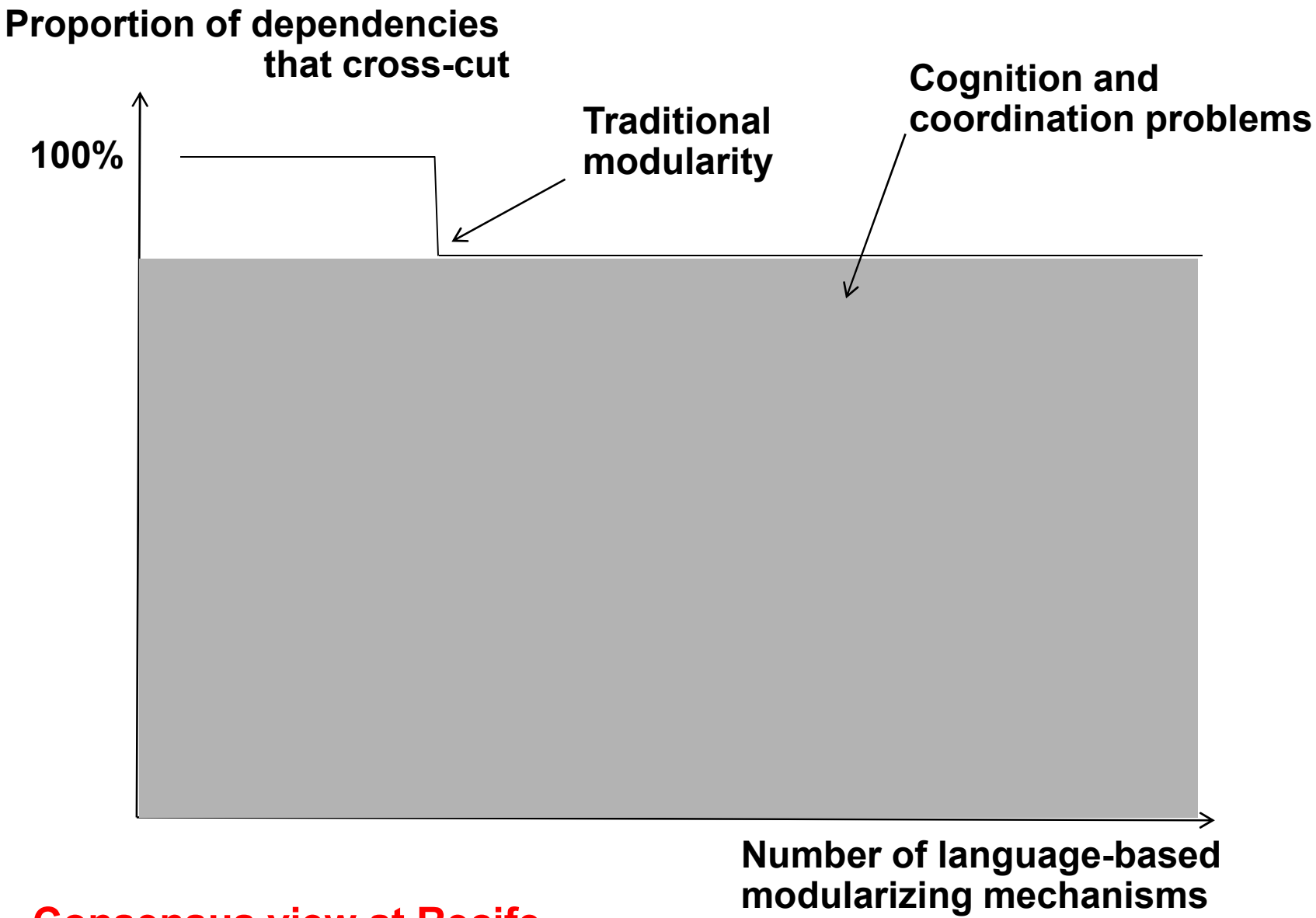
Carnegie Mellon
School of Computer Science



Carnegie Mellon
School of Computer Science

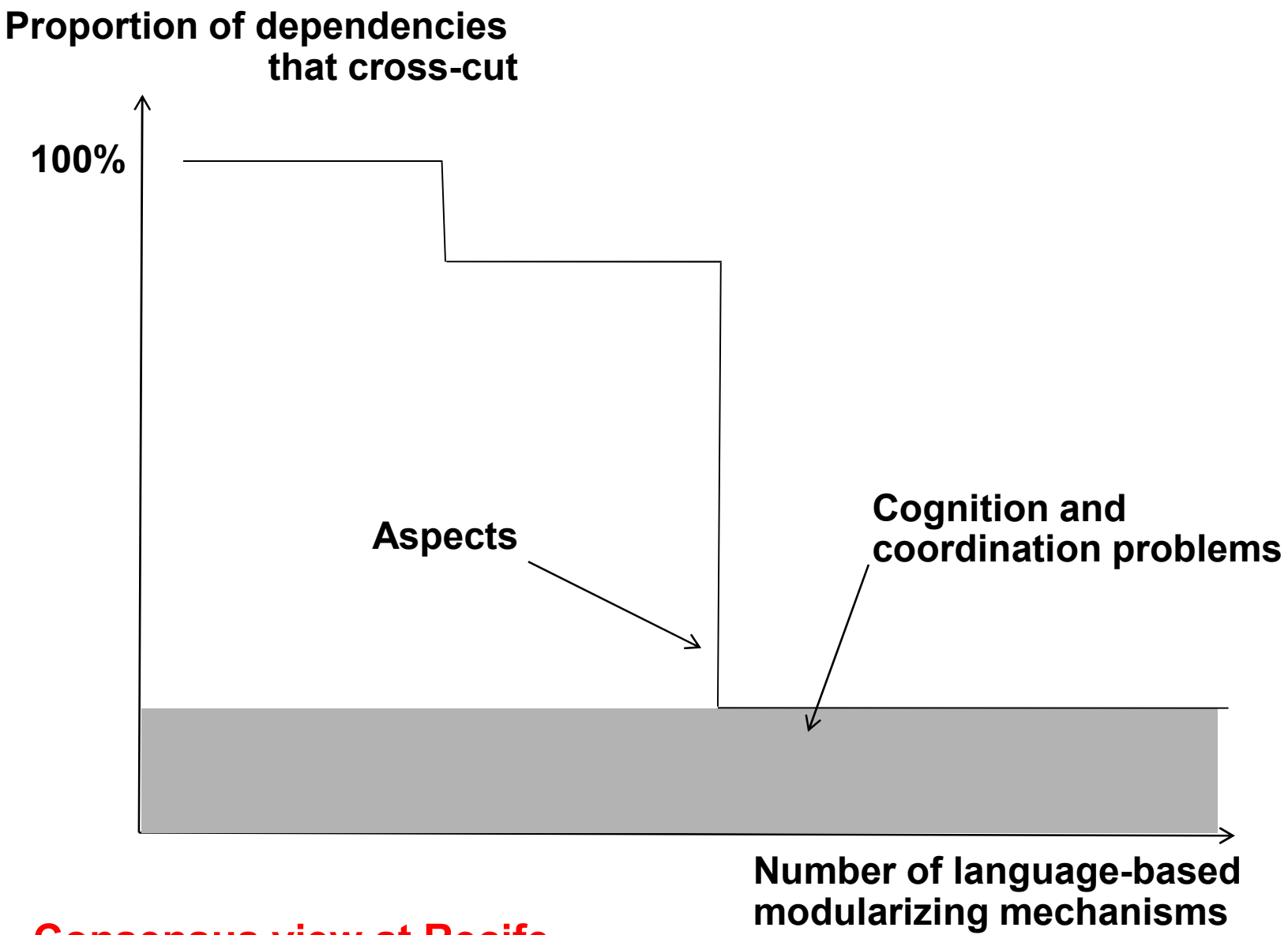


Carnegie Mellon
School of Computer Science



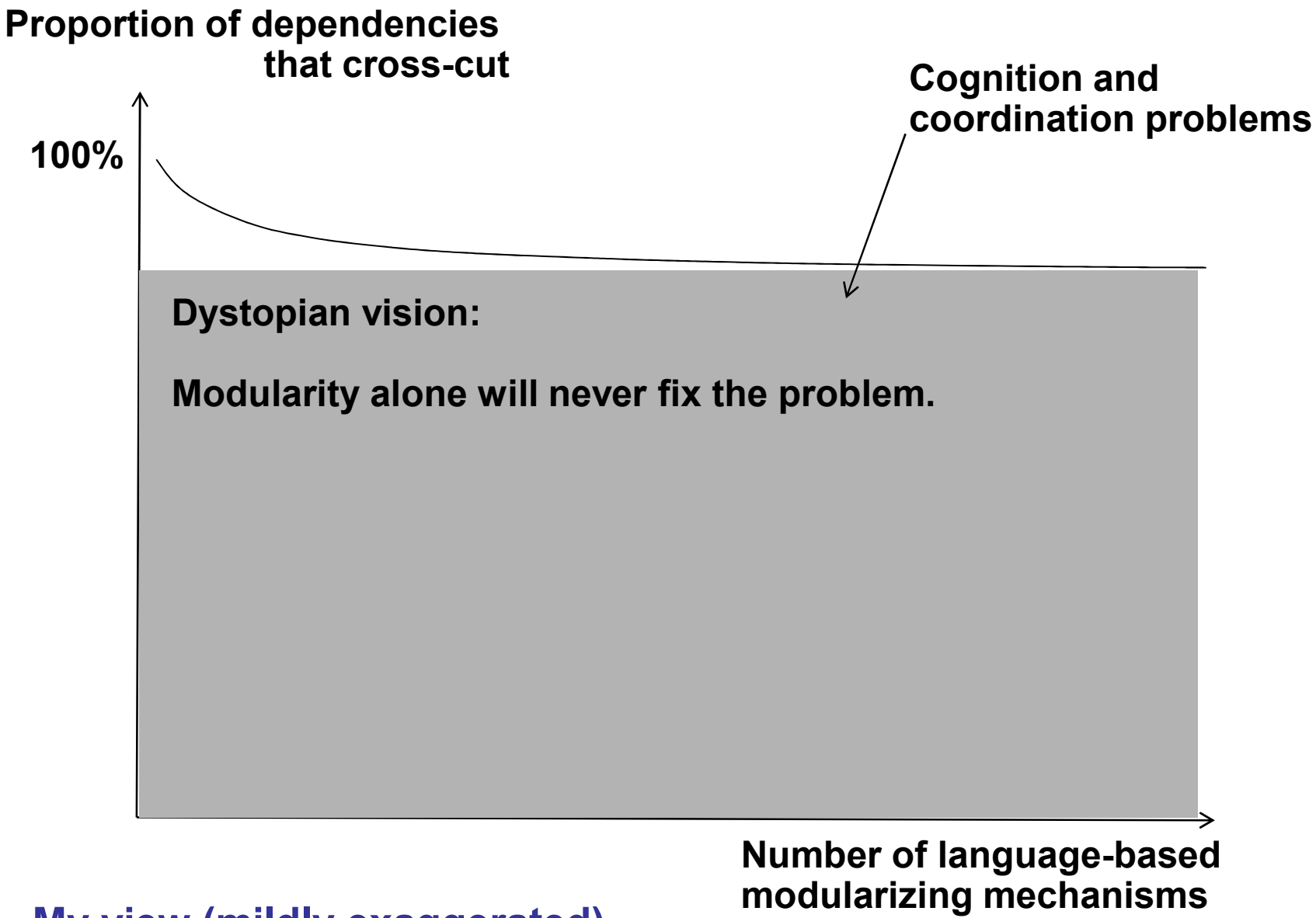
Consensus view at Recife

Carnegie Mellon
School of Computer Science



Consensus view at Recife

Carnegie Mellon
School of Computer Science



My view (mildly exaggerated)

Approaching the Gray Area . . .

- Organizational design, work assignment, and tools set up to bring the right dependencies to the attention of the right people so they can act appropriately

Two Examples . . .

- Organizational design and work assignment
  - Lessons from feature-driven development
- Using information from the environment
  - Learning from human activity

Feature-Driven Development

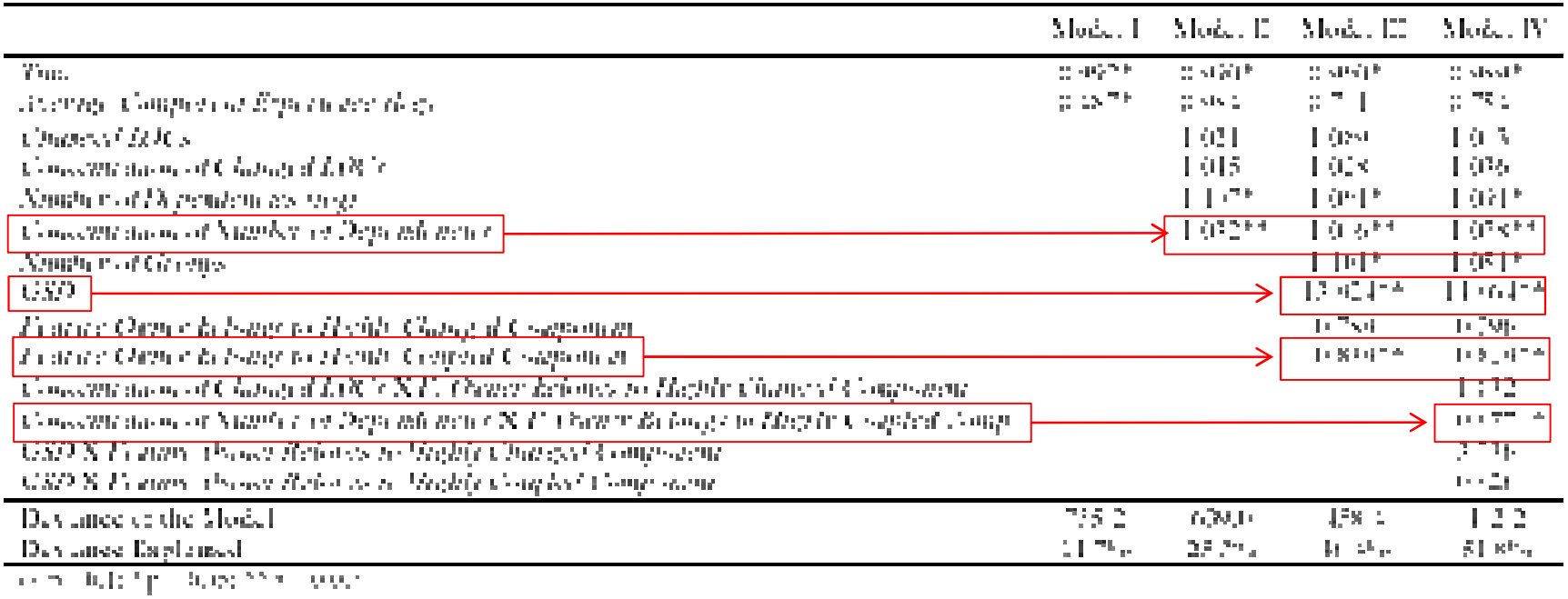
- Unit of functionality in end-user terms
- Feature is the unit of development managed by a project
- Features tend to cut across traditional software entities
- Work often overseen by “feature manager”
- Developers associated with component, assigned to work on particular features

The Study

- Setting
  - Software for automotive navigation system
  - 1195 features
  - 32 months of activity
  - 179 engineers in 13 teams
  - 1.5 M LOC, 6789 source files, 107 architectural components
  - Organization had 5 years of prior experience with feature-driven development
- Architects prepare feature development specification

Carnegie Mellon

What Causes Integration Failure?



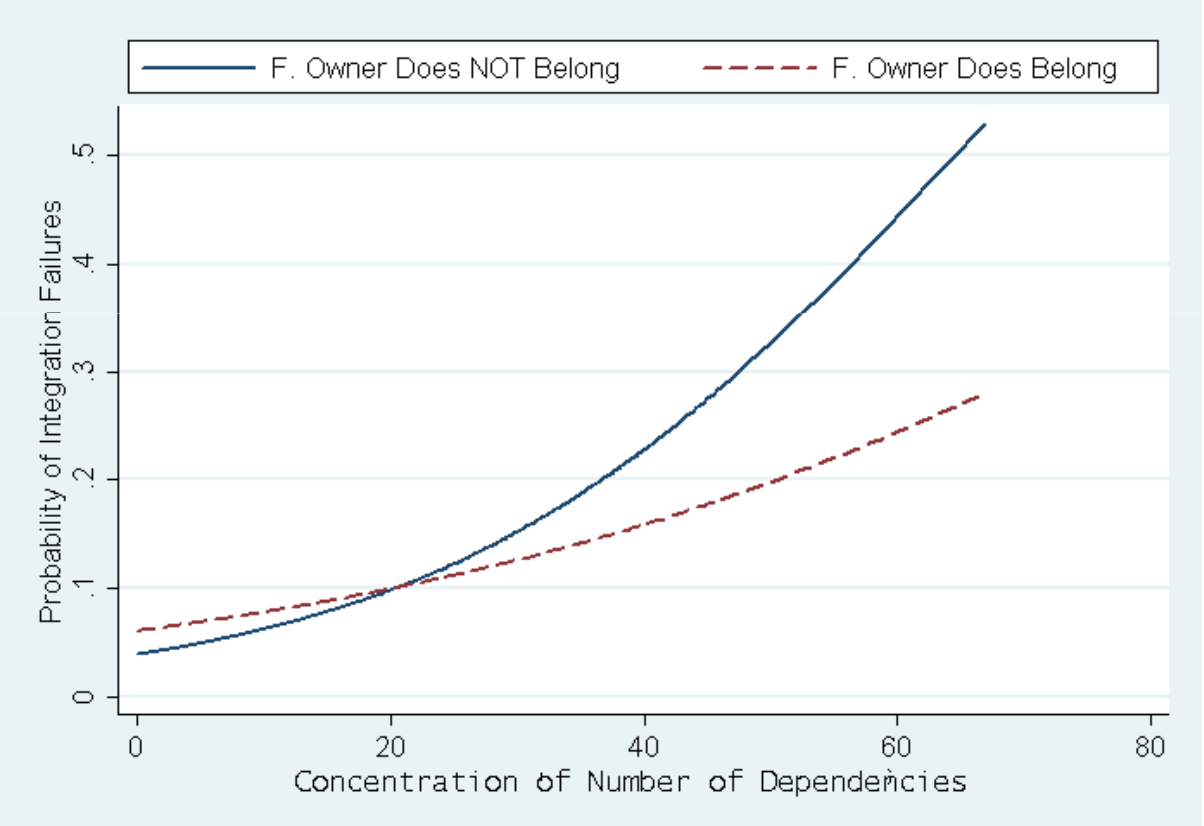
Odds Ratios from Regression Assessing Factors Driving Feature Integration Failures

From Cataldo, M. & Herbsleb, J.D. (2011). Factors Leading to Integration Failures in Global Feature-Oriented Development: An Empirical Analysis. *Proceedings, International Conference on Software Engineering* (to appear).

institute for SOFTWARE RESEARCH

Carnegie Mellon

Ownership Matters!

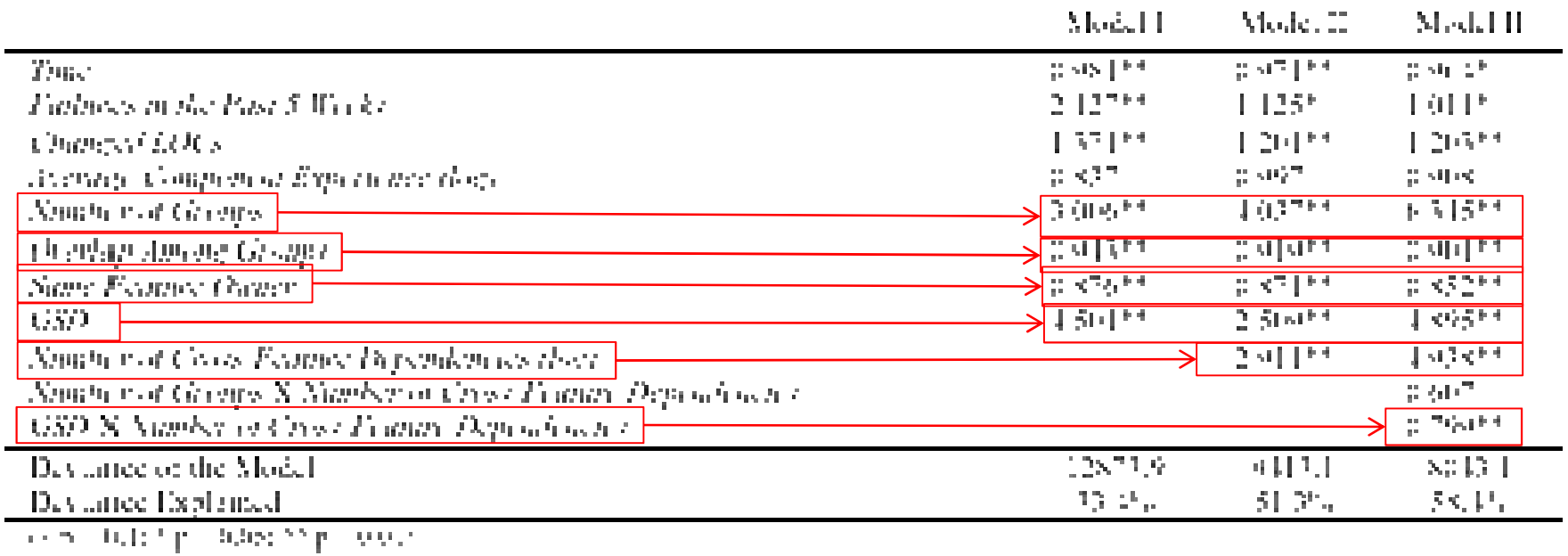


From Cataldo, M. & Herbsleb, J.D. (2011). Factors Leading to Integration Failures in Global Feature-Oriented Development: An Empirical Analysis. *Proceedings, International Conference on Software Engineering* (to appear).

isr institute for SOFTWARE RESEARCH

Carnegie Mellon

Destructive Feature Interaction



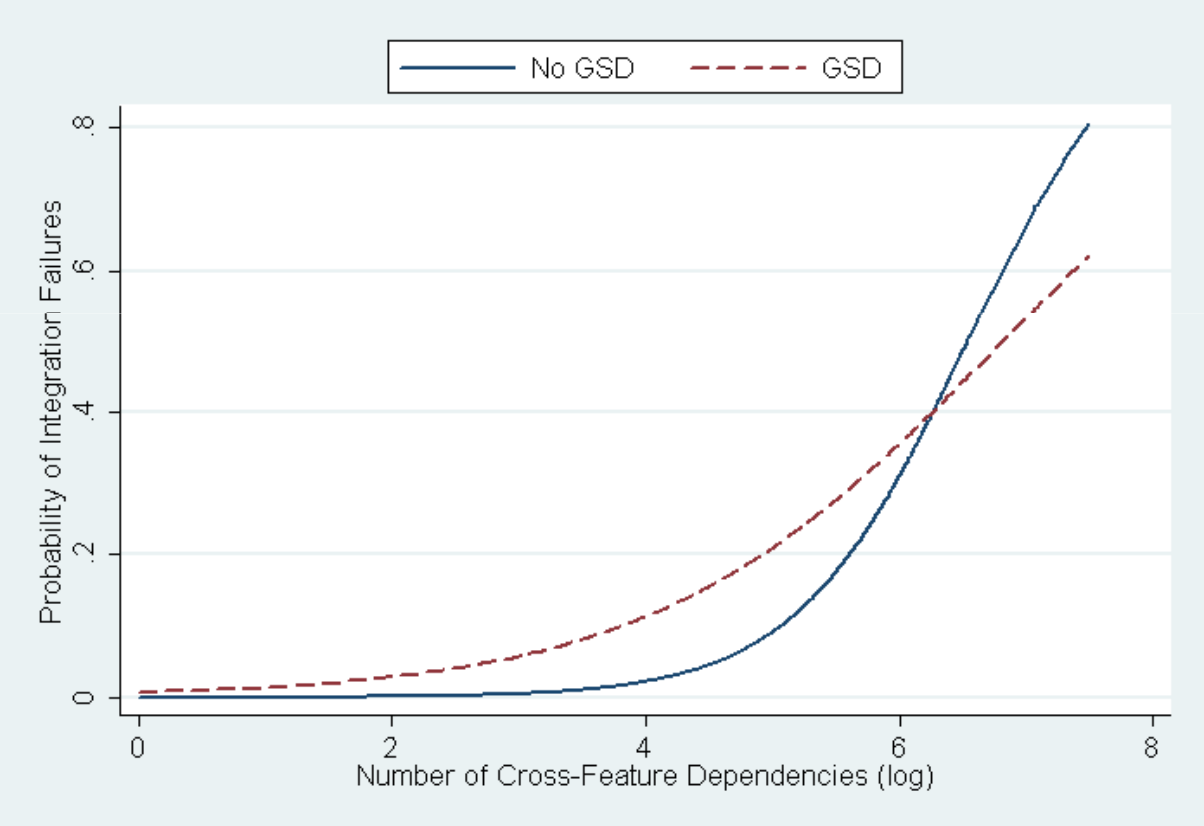
Odds Ratios from Regression Assessing the Impact of Cross-Feature Interactions on Integration Failures

From Cataldo, M. & Herbsleb, J.D. (2011). Factors Leading to Integration Failures in Global Feature-Oriented Development: An Empirical Analysis. *Proceedings, International Conference on Software Engineering* (to appear).

isr institute for SOFTWARE RESEARCH

Carnegie Mellon

Co-location Doesn't Scale



From Cataldo, M. & Herbsleb, J.D. (2011). Factors Leading to Integration Failures in Global Feature-Oriented Development: An Empirical Analysis. *Proceedings, International Conference on Software Engineering* (to appear).

isr institute for SOFTWARE RESEARCH

Carnegie Mellon

Broader Lessons

- Organizational arrangements matter!
- Effects can be quite large
- Effects often are not commonsensical

Carnegie Mellon

Inferring Dependencies from Traces of Human Activity

- Prior work
  - Use files changed together as measure of dependencies
  - Can generate a measure of coordination requirements
  - Validated in a number of settings
- Can we generalize from “files changed together” to “entities discussed together”?

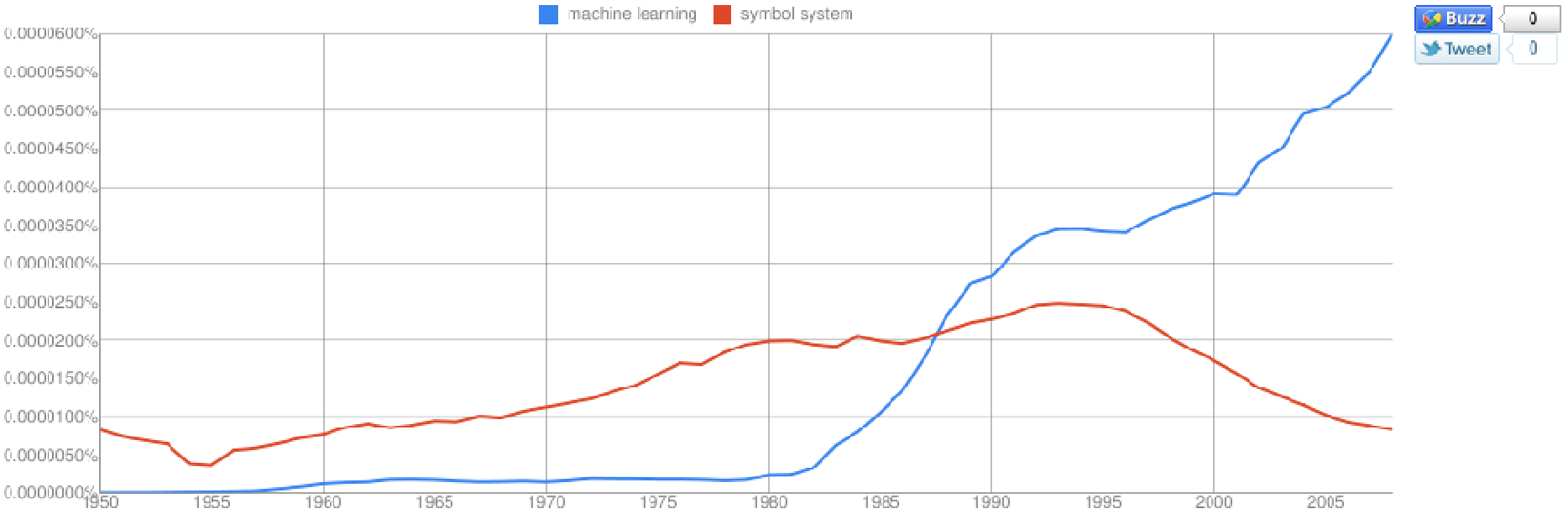
A Brief Digression/Analogy

Google labs Books Ngram Viewer

Graph these **case-sensitive** comma-separated phrases: machine learning,symbol system

between 1950 and 2008 from the corpus English with smoothing of 3.

Search lots of books



Search in Google Books:

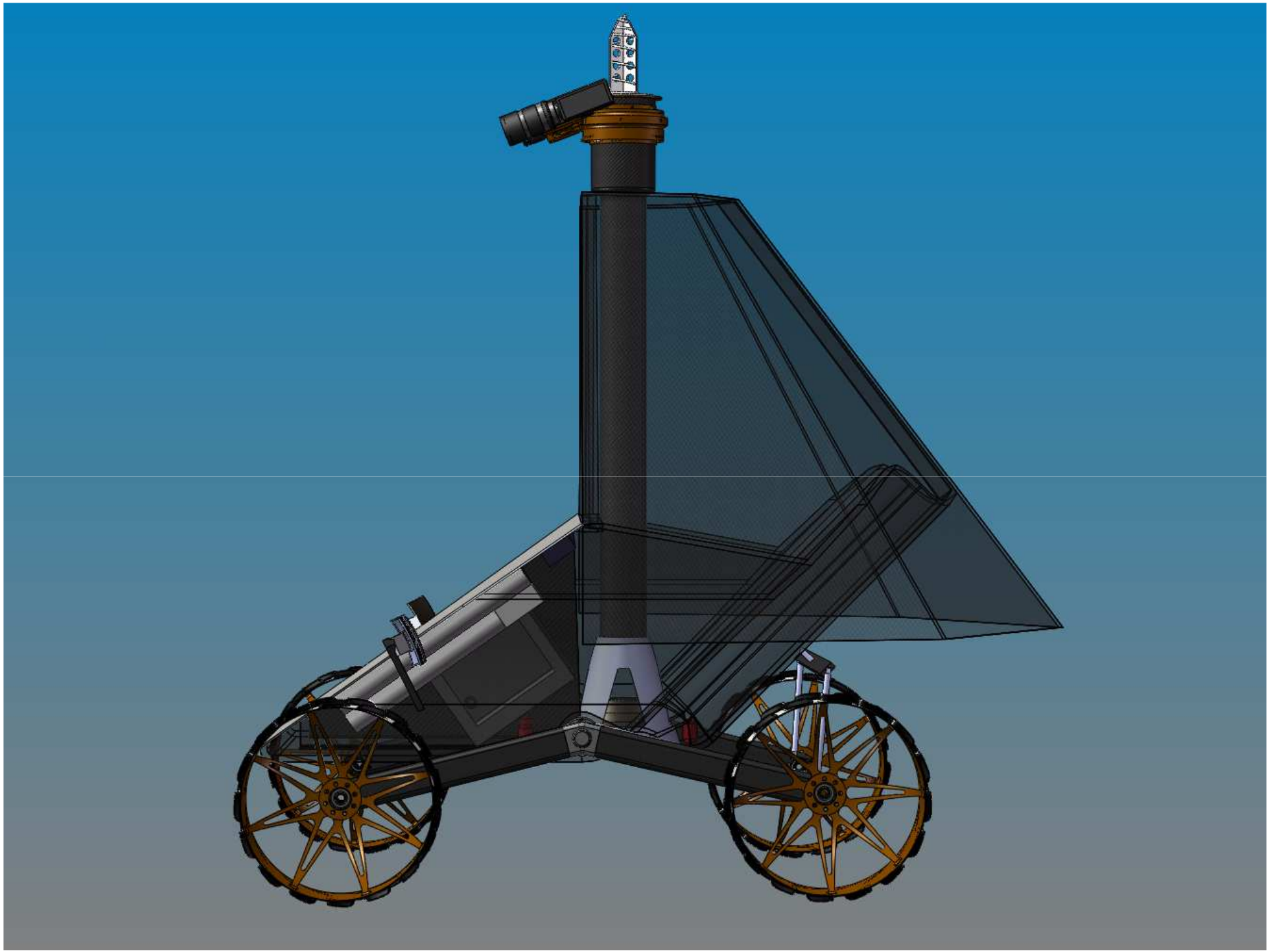
1950 - 1987	1988 - 2003	2004 - 2005	2006 - 2007	2008	machine learning
1950 - 1961	1962 - 1992	1993 - 1995	1996 - 1999	2000 - 2008	symbol system

Run your own experiment! Raw data is available for download here.

Carnegie Mellon

Text Analysis: Field Robotics

- Project
  - Lunar X Prize competition



Carnegie Mellon

Text Analysis: Field Robotics

- Project
  - Lunar X Prize competition
- No automatically collected version or change data
- Constantly shifting component boundaries and interfaces
- Can we use text analysis to derive dependencies?

Carnegie Mellon

Steps

- Collected data
  - 25 all-hands meetings
  - About 10,000 words each
- Developed code book
  - 6 field robotics articles

Carnegie Mellon

Code Book

Component:	Brief description:
Communications	Communications external to the robot, for control or mission, including operator interface.
External Relations	Acquiring external resources (incl. funding, parts, & purchasing), publicity, investor & media relations.
Internal Relations	Project/program management, HR, task assignments, training, collaboration tools, clarifying norms & expectations.
Mobility Effectors / Actuators	Effectors and actuators that propel the entire robot: e.g. tracks, wheels, shocks, & motors with associated firmware.
Mission Specific Effectors / Actuators	All other motors, gears, & moving parts that don't move the robot as a whole, e. g. camera mast rotation motor.
Perception software / computing	Software, and any dedicated hardware, for: terrain mapping, environmental modeling, and/or object detection. Camera/lens zoom, shutter, and focus control software.
Planning software / computing	Mission task planning, including the overall mission plan and computing resources for semi-autonomous execution.
Power	Includes batteries, solar cells, switches, power cables & controls.
Sensors	Camera; thermal, ultrasonic, tactile, radar/sonar range sensors; Inertial Measurement Unit, GPS, & any wiring or processing going from sensors to controls.
Shared / general computing	Includes general purpose processors / onboard computers (e. g. avionics box). Abbreviated "gpp."
Structure	Chassis, fasteners (e.g. Frangibolt, weld joints), radiator, payload, paints, reflectors.

isr institute for SOFTWARE RESEARCH

Carnegie Mellon

Steps

- Collected data
  - 25 all-hands meetings
  - About 10,000 words each
- Developed code book
  - 6 field robotics articles
- Manual coding of decision discussions
  - Tested inter-rater reliability
    - QAP correlations .80

Carnegie Mellon

Text Pre-Processing

Step	Description
1	Removed contractions (e. g. changing "what's" to "what is").
2	Applied a Krovetz (dictionary-based) stemmer to covert terms into morphemes
3	Removed common English terms (e. g. 'the'), replacing them with placeholders ('xxx').
4	Removed punctuation.
5	Turned meaningful bigrams into unigrams (e.g. 'solar cells' became 'solar_cells').

Carnegie Mellon

Steps

- Collected data
  - 25 all-hands meetings
  - About 10,000 words each
- Developed code book
  - 6 field robotics articles
- Manual coding of decision discussions
  - Tested inter-rater reliability
    - QAP correlations .80
- Created thesaurus

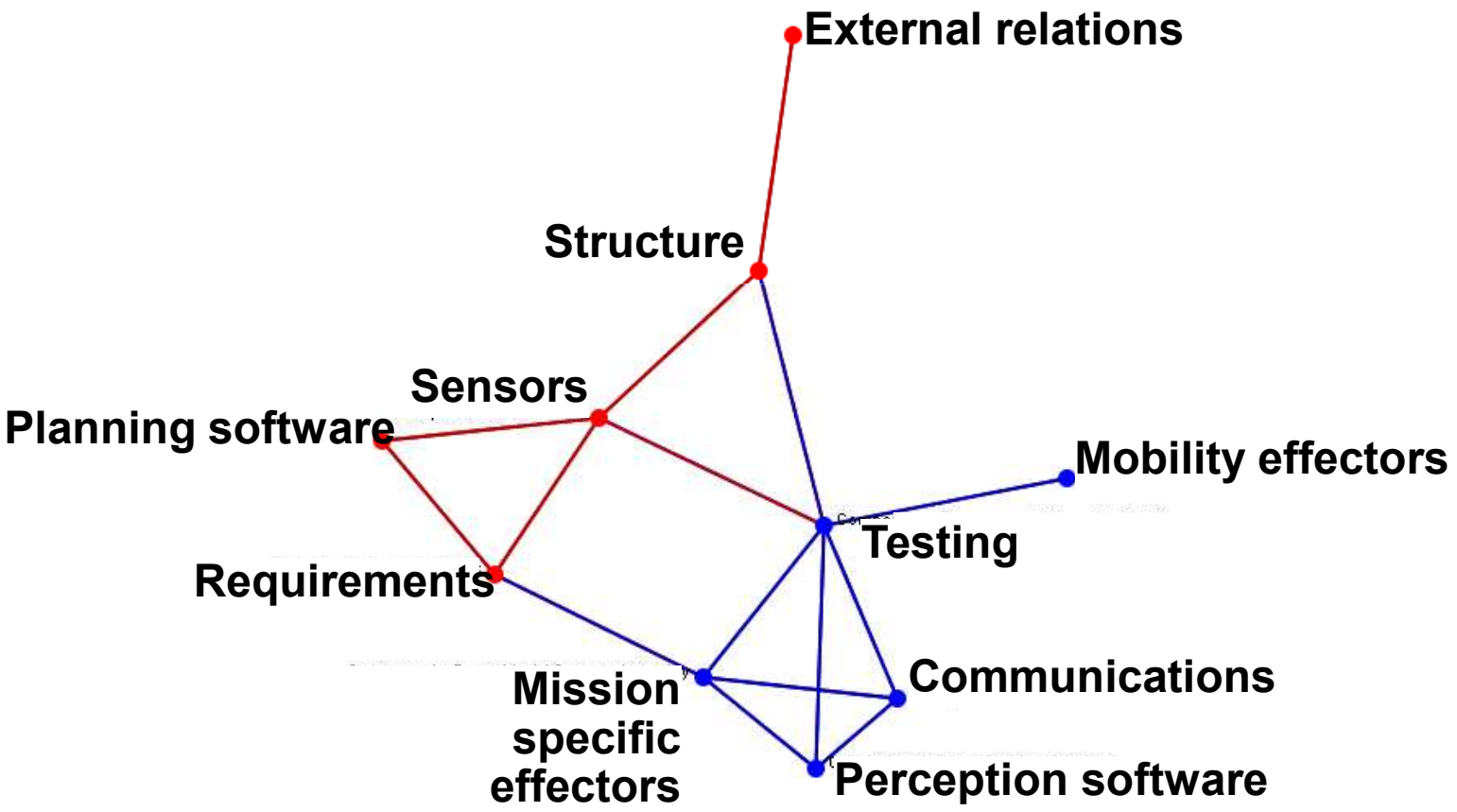
Carnegie Mellon

Link Identification

- Co-occurrence of terms
- Human coding: same decision
- Selected sliding window size
  - Size 15 had best agreement with hand coding
  - Threshold established
- QAP correlations comparable to human-human agreement (~.8)
- Sets of links based on topics

Carnegie Mellon

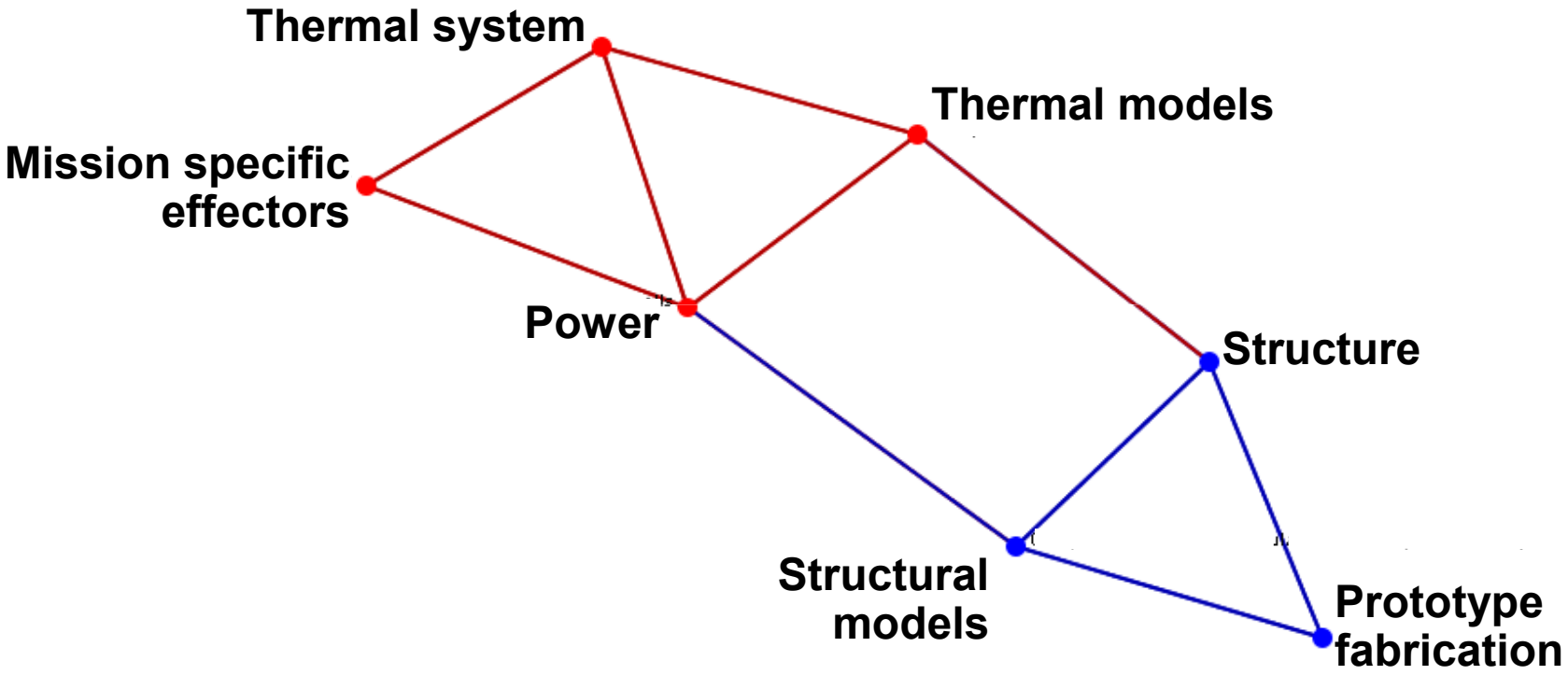
Optics



institute for SOFTWARE RESEARCH

Carnegie Mellon

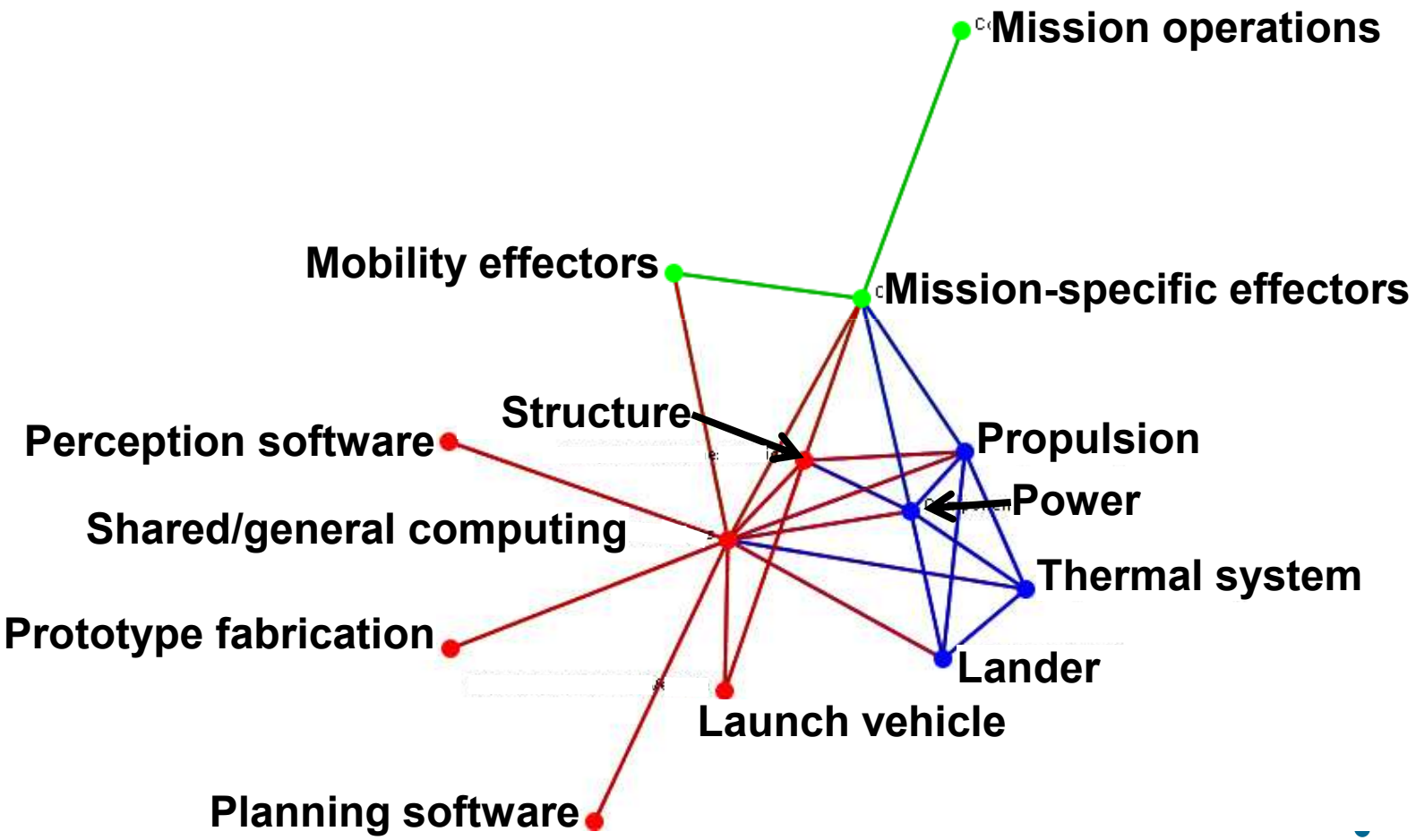
Thermal



institute for SOFTWARE RESEARCH

Carnegie Mellon

Avionics



institute for SOFTWARE RESEARCH

Carnegie Mellon

Concluding Vision

- The gray area – work that cross-cuts language constructs – is here to stay
- Use organizational tactics
- Use computations over artifacts generated by development activities
- Explore new data sources, including documents and conversation
  - Activities reveal knowledge
  - Analysis can often make it actionable