

Almaden Institute on Complexity, April 11-12, 2007

# A Few Thoughts on Software Complexity

*Dr. Alfred Z. Spector  
Independent Consultant  
azs@azs-service.com*

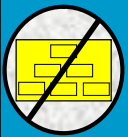
April 11, 2007

# Excerpts from 2/6/2003 Talk @ MIT: *The Conundrum of Systems*<sup>1</sup>

<sup>1</sup><http://www.research.ibm.com/people/a/aspector/presentations/AZSDertouzos.pdf>  
<http://www.csail.mit.edu/events/DLStalks/dlsspector03.html>

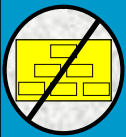
# Outline

- Motivation: Complex software systems, built using widely prescribed techniques, often do not delight their constituencies. They are too complex.
- What is complexity?
- Why are our systems too complex?
- What could Computer Science do to measure and resolve this?
- Should Computer Science undertake this challenge?



# Software Systems Today

- Score high on most metrics:
  - Amount of code
  - # of dependencies
  - # of programmatic interfaces
  - # of layers
  - Administrative interface size & configuration options
  - Non-uniformity
  - Non-orthogonality
  - Defects
  - Documentation
  - # of programmers involved
  - *Added: dynamic change*



# RosettaNet Purchase Orders

- There are **551** XML fields in the PurchaseOrderRequest
- There are **700** XML fields in the PurchaseOrderConfirmation

Excerpted First lines of purchase order confirmation:

```

fromRole.PartnerRoleDescription
|- ContactInformation
|  |- contactName.FreeFormText
|  |- EmailAddress
|- facsimileNumber.CommunicationsNumber
|  |- telephoneNumber.CommunicationsNumber
|- GlobalPartnerRoleClassificationCode
|- PartnerDescription
|  |- BusinessDescription
|     |- GlobalBusinessIdentifier
|     |- GlobalSupplyChainCode
|  |- GlobalPartnerClassificationCode
GlobalDocumentFunctionCode
PurchaseOrder
|- AccountDescription
|  |- accountName.FreeFormText
|  |- AccountNumber
|  |- billTo.PartnerDescription

```

```

...
|- BusinessDescription
|  |- businessName.FreeFormText
|  |- GlobalBusinessIdentifier
|  |- PartnerBusinessIdentification
|     |- ProprietaryBusinessIdentifier
|     |- ProprietaryDomainIdentifier
|     |- ProprietaryIdentifierAuthority
|- ContactInformation
|  |- contactName.FreeFormText
|  |- EmailAddress
|  |- facsimileNumber.CommunicationsNumber
|  |- PhysicalLocation
|     |- GlobalLocationIdentifier
|     |- PartnerLocationIdentification
|        |- ProprietaryDomainIdentifier
|        |- ProprietaryIdentifierAuthority
...

```

Note: RosettaNet is a consortium of major companies working to create and implement industry-wide, open e-business process standards, that will form a common e-business language, globally aligning processes between supply chain partners. (From RosettaNet Home Page.)

aspector@us.ibm.com

02/06/03



# 3 Categories of Complexity

## ■ Classic Complexity ■ Usage Complexity

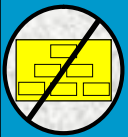
- Time
- Space

## ■ Implementation Complexity

- Logical
- Structural
- Comprehensibility

Task	Pre-Use	Novice	Middle	Expert	Exception
Install					
Configure					
Administer					
Use					

- (Since this talk given, other categories are becoming clearer)



# There are steps to take

- Meaning
- Measuring
- Methodology
- System Architecture
- Science and Technology
- Acknowledgment, Legal & Cultural Change



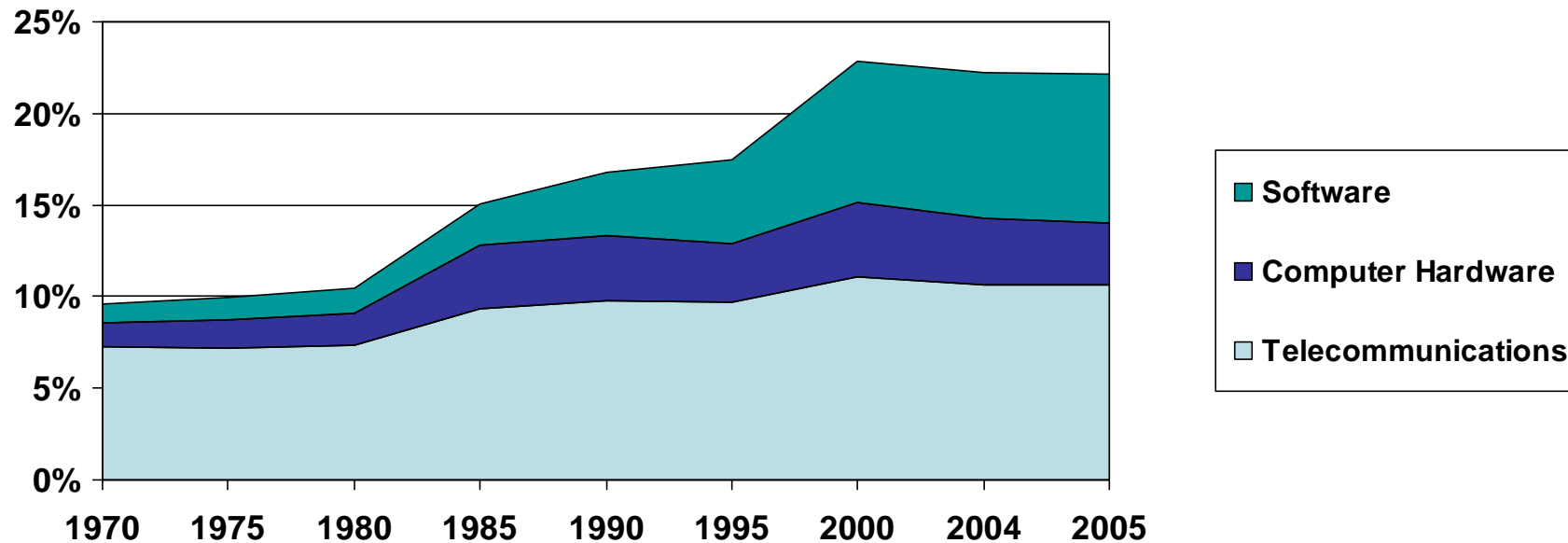
**Excerpts from**  
***Towards a Software Science of Design***<sup>1</sup>  
**Presentation to NSF SoD Workshop 3/1/07**

<sup>1</sup><http://www.cs.virginia.edu/~sullivan/SODPI07/ScienceOfDesign2007SpectorKeynote.pdf>

## Outline

1. Design, Broadly
2. Software Design, Broadly
3. Changing Nature and Impact of Software
4. Successes, but Great Limitations
5. A Software Science of Design
6. Elements of a Research Agenda
7. Observations and Conclusions

## There is More Software to Integrate and Assemble



Source: Table 2.1, Current-Cost Net Stock of Private, Commercial Fixed Assets, Equipment and Software, National Bureau of Economic Analysis

- I/T (particularly S/W) a Growing Share of U.S. Commercial Capital Stock<sup>1</sup>
- Note: Global I/T expenditure per year is roughly  $\$1.5 \times 10^{12}$
- My analysis: Growth in I/T value is probably much steeper; why?
  - In terms of value, I/T has grown much faster due to price deflation
  - Software depreciation/expensing is demonstrably fast

<sup>1</sup>and by implication, the World's.

## Towards a Software Science of Design

**A Software Science of Design would facilitate software production across the lifecycle by helping to understand, codify and integrate currently understood approaches to design, but also by creating better approaches and tools.**

- A broad, integrative mission
- **Key foci:**
  - Cross-fertilization with other design disciplines
  - Holistic approach
  - Scientific analysis (not, “here’s a new approach, it’s real cool ☺”)
  - Education and standardization of approaches
  - Interdisciplinary work
- **Elements probably not included**
  - Domain-specific abstractions and their implementations
  - Elements directly covered in other areas of the field:
    - Programming Language Design
    - Verification
    - Etc.
  - These are admittedly blurry lines
- **Note:**
  - I don’t think that S/W design, broadly, will admit to as much standardization as exists in, say, road design
  - Nonetheless, there can still be families of approaches, parameterized (or meta-), approaches & tools...

## Elements of a Research Agenda

- **Terminology and reference models**
- **Metrics**
- **Modularity and reuse**
- **Workbenches/platforms**
- **Development methodologies**
- **Automated assembly**
- **Design for change**
- **Design for security**
- **Complexity**
- **Education**
- **Team culture**
- **Impact of accounting and taxation**
- **Impact of liability, intellectual property, and anti-trust law**

## Observations and Conclusions

- **The state of software design is not sound.** (1) We don't know enough (2) we rarely *codify* (in the broadest sense) what we know (3) we don't consistently use or educate what we codify.
- **Design is everywhere.** While particularly important in software, there should be a cross-discipline design community looking for synergies in innovation, education, and practice.
- **Design in software is multi-faceted.** It applies to all aspects of the lifecycle. A Science of Design should be broad enough to address all aspects.
- **Design is of growing importance.** Related not only to amount of software, more serious applications of software, greater diversity of it's development across time and space, and post-facto, dynamic assembly, with a greater base of components, design (but not programming) is oft. needed! Implications are manifold.
- **The research agenda has great opportunity.** There are plenty of topics for a broad research agenda. Work should proceed towards categorization and completeness: tops down and bottoms up.
- **Interdisciplinary research is of particular important.** The design of software is not purely a problem of computer science.
- **Establishment of objectives.** We should establish quantifiable objectives.

## Summary

- **Complexity is a growing problem in software-related systems**
  - Given the need for increasing automation and automation in very important/life critical systems, this is ever more important
- **Holistic study of design is an appropriate response**
- **There is a significant, feasible research agenda.**

**Thank you**