§5. 对象技术的现状与发展趋势

5.1 对象技术发展过程及其分析

5.1.1 在实验室里的二十年（1967~1986）

- 与其他计算机技术类似，OO 技术也是起源于解决特定类别的应用问题。针对一类应用－模拟－的要求，1967 年，在挪威 Oslo 大学产生了第一个面向对象语言：Simula 67。

5.1 对象技术发展过程及其分析

5.1.1 在实验室里的二十年（1967~1986）

- 与其他计算机技术类似，OO 技术也是起源于解决特定类别的应用问题。针对一类应用－模拟－的要求，1967 年，在挪威 Oslo [illegible] ula 67。

Ole-Johan Dahl
1931.10.12~2002.6.29

Kristen Nygaard
1926.8.27~2002.8.10

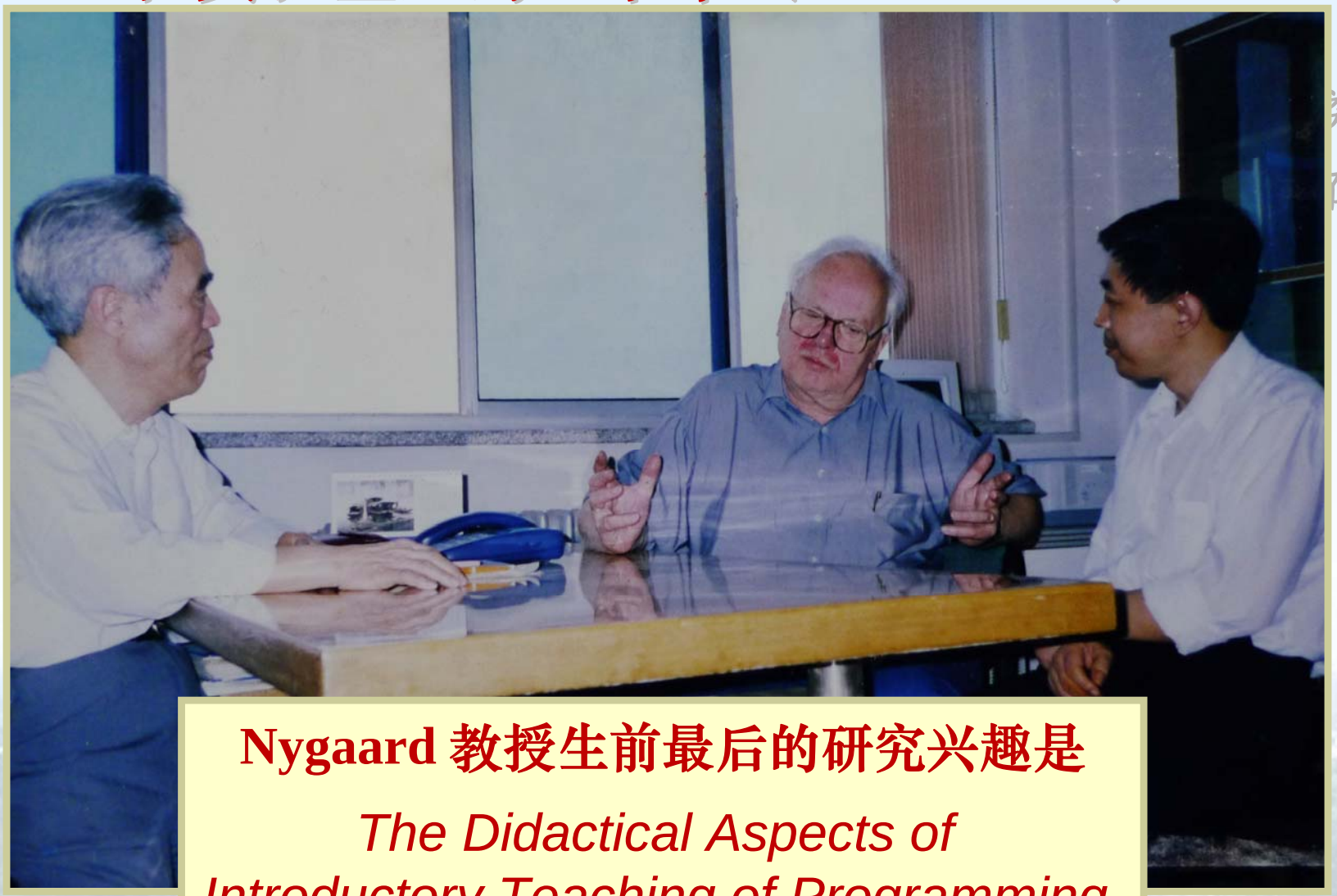
2001年ACM
图灵奖获得者

两位大师
都访问过西电！

5.1 对象技术发展过程及其分析

5.1.1 在实验室里的二十年（1967~1986）

- 类别
  在挪



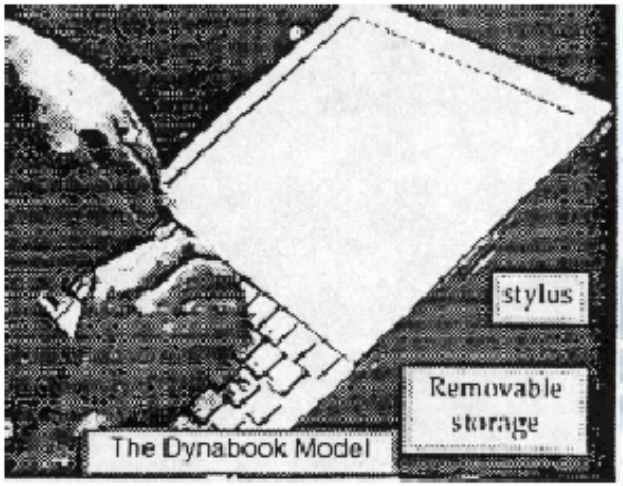
Nygaard 教授生前最后的研究兴趣是

The Didactical Aspects of Introductory Teaching of Programming

5.1 对象技术发展过程及其分析

5.1.1 在实验室里的二十年（1967~1986）

- **Alan Kay** 在研究生学习期间（**1960** 年代后期），就开始构思一种纯粹的个人计算机，他称之为 **Dynabook** – 这是一种以对象为基本元素的笔记本计算机!
- 虽然 **Dynabook** 及其用户环境仅仅发展到规格说明就停止了，但这种大胆的构思将对象从程序结构引伸到更宽的范围，特别是基于图形人机界面（*GUI*）的用户环境。



Dynabook

- Small, handheld
- Wireless networking
- A Personal Computer for Children of All Ages

5.1 对象技术发展过程及其分析

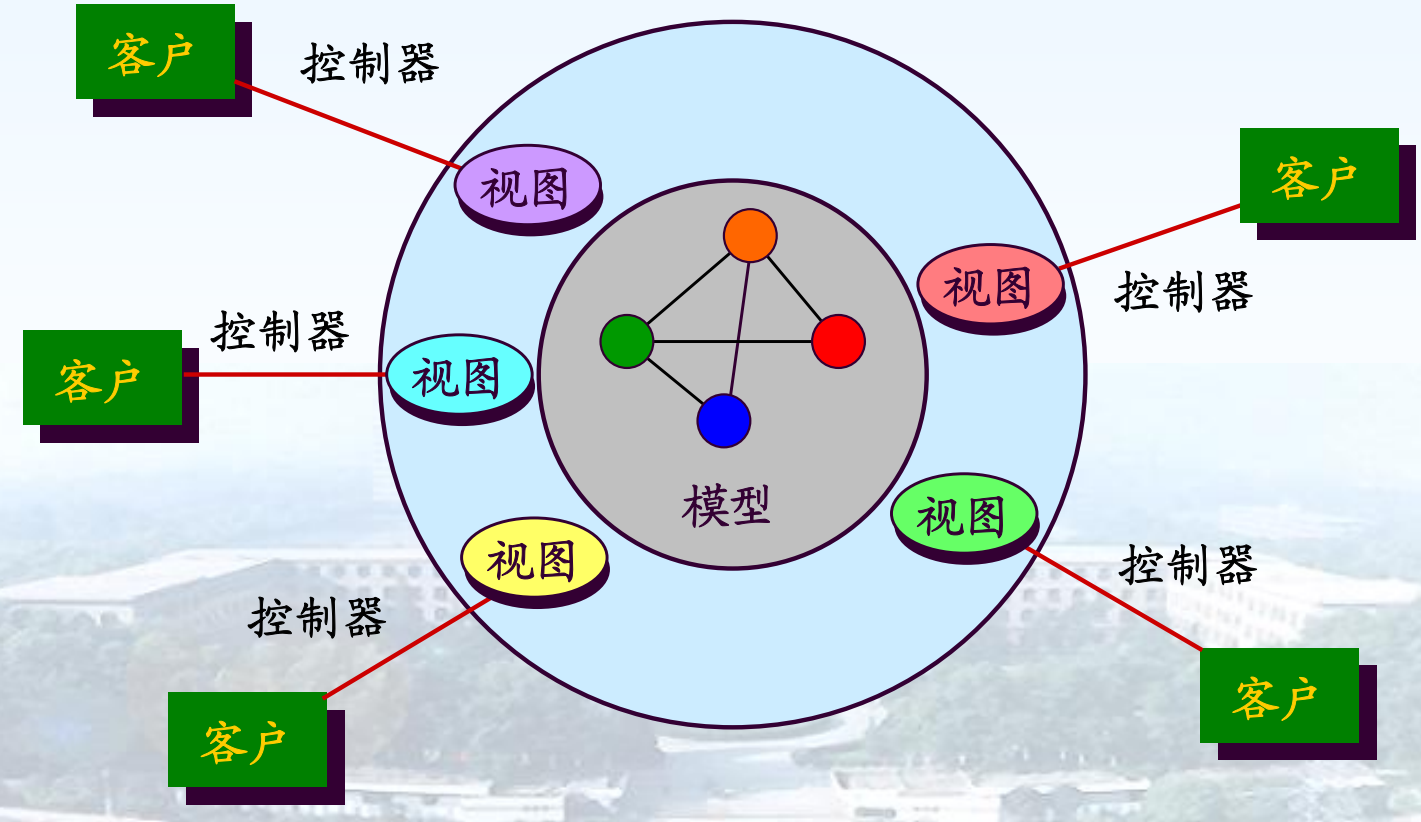
5.1.1 在实验室里的二十年（1967~1986）

- 研究生毕业后，Alan Kay 到 Xerox Palo Alto Research Center（PARC）工作，他关于用户环境的设想被 Dan Ingalls 和 Adele Goldberg 领导的一个工作组采纳，从而以 Simula 67 的对象概念为基础，1976 年，产生了第一个纯粹的面向对象程序设计语言和用户环境－Smalltalk。
- 采纳了 Alan Kay 的理念， Smalltalk 注重以一种新型人机交互方式－MVC－为核心的用户环境，这种交互方式用一种新型的点定位设备－鼠标－来交互式地操纵图形对象。

5.1 对象技术发展过程及其分析

5.1.1 在实验室里的二十年（1967~1986）

- **Smalltalk** 的 **MVC**（*Model-Views-Controller*）结构对以后软件环境的结构设计乃至软件设计产生了深刻的影响。



5.1 对象技术发展过程及其分析

5.1.1 在实验室里的二十年（1967~1986）

- AT&T Bell 实验室的 Bjarne Stroustrop 开始是为了纠正 C 语言的一些错误和改进 C 的一些弱点，引入了强类型化机制，借鉴了 Simula 67、Algol 68 中的一些机制，结果形成了 C++ 语言。
- 同时期还出现了 Objective-C、CLOS、Object-Prolog、Object-Pascal、Object-COBOL 等一批混合型的（*Hybrid*）面向对象程序设计语言。
- 至此，OOP 技术已经基本成型。

5.1 对象技术发展过程及其分析

5.1.2 尝试走出实验室的八年（1987~1994）

- Apple Computer 的一些开发人员在 PARC 访问学习之后，带回了与操作系统及人机交互有关的一些激进的、新的设计理念，形成了 Apple Macintosh 操作系统的基础，这些理念同样也影响了 X Window，Motif，乃至 Microsoft Windows。
- 这一阶段商品化的 OO 语言和工具还不多，如 Digitalk 和 VisualWorks 的 Smalltalk，Microsoft 和 Borland 的 C++ 等。
- 这一时期，面向对象数据库（OODB）是 OO 技术的一个研究热点，但至今基本上没有走出实验室。

5.1 对象技术发展过程及其分析

5.1.2 尝试走出实验室的八年（1987~1994）

- Apple Computer 的一些开发人员在 PARC 访问学习之后，带
- 这一阶
- 这一时

5.1 对象技术发展过程及其分析

5.1.3 成为主流技术的十多年（1995~目前）

- 1993 年，克林顿政府提出了“信息高速公路”计划，将其技术基础定为：微电子、光电子、面向对象。
- 1990 年代中期，一批新型计算范型（如广域网、多层客户服务器结构的应用系统、图形人机界面、分布系统等）的出现，特别是 WWW，使得 OO 技术进入主流领域，成为软件开发和生产的一种支撑技术。
- 1995 年出现了以新型图形界面进行互联网访问的 Web 浏览器，Netscape 一度主导着技术与市场。

5.1 对象技术发展过程及其分析

5.1.3 成为主流技术的十多年（1995~目前）

- 1995 年 Sun 推出了 Java 语言及其支撑体系，允许通过浏览器访问和下载 Web 上的内容，实现了程序片断在互联网上的迁移和执行。
- 1996 年 Netscape 承诺支持分布对象标准和 Java。
- 在此期间，OMG 制定了分布式互操作标准 CORBA，其核心机制 ORB（*Object Request Brokers*）就是一个面向对象系统，支持在分布式应用之间采用标准的面向对象接口进行互操作。作为ORB的一部分，也为互联网应用提供了互操作协议 IIOP（*Internet Inter-Operability Protocol*）。

5.1 对象技术发展过程及其分析

5.1.3 成为主流技术的十多年（1995~目前）

- 在此期间，**OMG** 推进了 **UML** 和 **UP**（*Unified Process*）的标准化和应用。
- 在此期间，主要的构件模型（*如 Java Beans、EJB、J2EE、COM/DCOM、Active X* 等），都采用 **OO** 技术作为核心技术。
- **1999年，出现了AOP和AspectJ，至今研究不衰。**
- **2000年，Microsoft 推出了 .Net 和 C#。**

5.1 对象技术发展过程及其分析

5.1.3 成为主流技术的十多年（1995~目前）

- 1990年代后期出现了一批敏捷方法（*Agile Methods*），如极限程序设计（*eXtreme Programming，XP*）等。2001年发布的敏捷宣言（*Agile Manifesto*）阐明了四项价值观:
  - 开发人员个体作用与相互间的交流比过程与工具重要
  - 正常运行的软件比广泛的文档重要
  - 与客户合作比合同谈判重要
  - 对变更的响应比遵循计划重要
- 分析认为，敏捷方法适用于较小的开发项目（风险度较低，有一批很能干的开发者，需求变化频繁）。

推荐一篇论文

- Barry Boehm, A View of 20th and 21st Century Software Engineering, *Proc. ICSE'06*, May, 2006, pp. 12-29.
- 论文以每10年为一个单元，回顾了软件工程领域从1950年代到21世纪前10年主要的成功经验和教训，讨论了将在2010年代乃至2020年以后影响软件工程实践的变化因素，以及评价和应对这些变化因素的一些策略。
- 论文对于我们全面认识软件技术、软件工程的发展历程与发展趋势是很有帮助的，因为作者在讨论重要的标志性成果时，既阐述其正面作用，也指出其负面影响或潜在的问题。
- 作者认为现在存在着一种很危险的倾向：软件生产者把大量的资源投入到尽快将产品投放市场上，却通常不把软件的可信放在首位。他预言这种倾向若持续下去，会催生一场将在目前到2025年之间发生的灾难，其震撼世界的作用类似于"911"事件！

这篇论文中的一张图值得认真琢磨

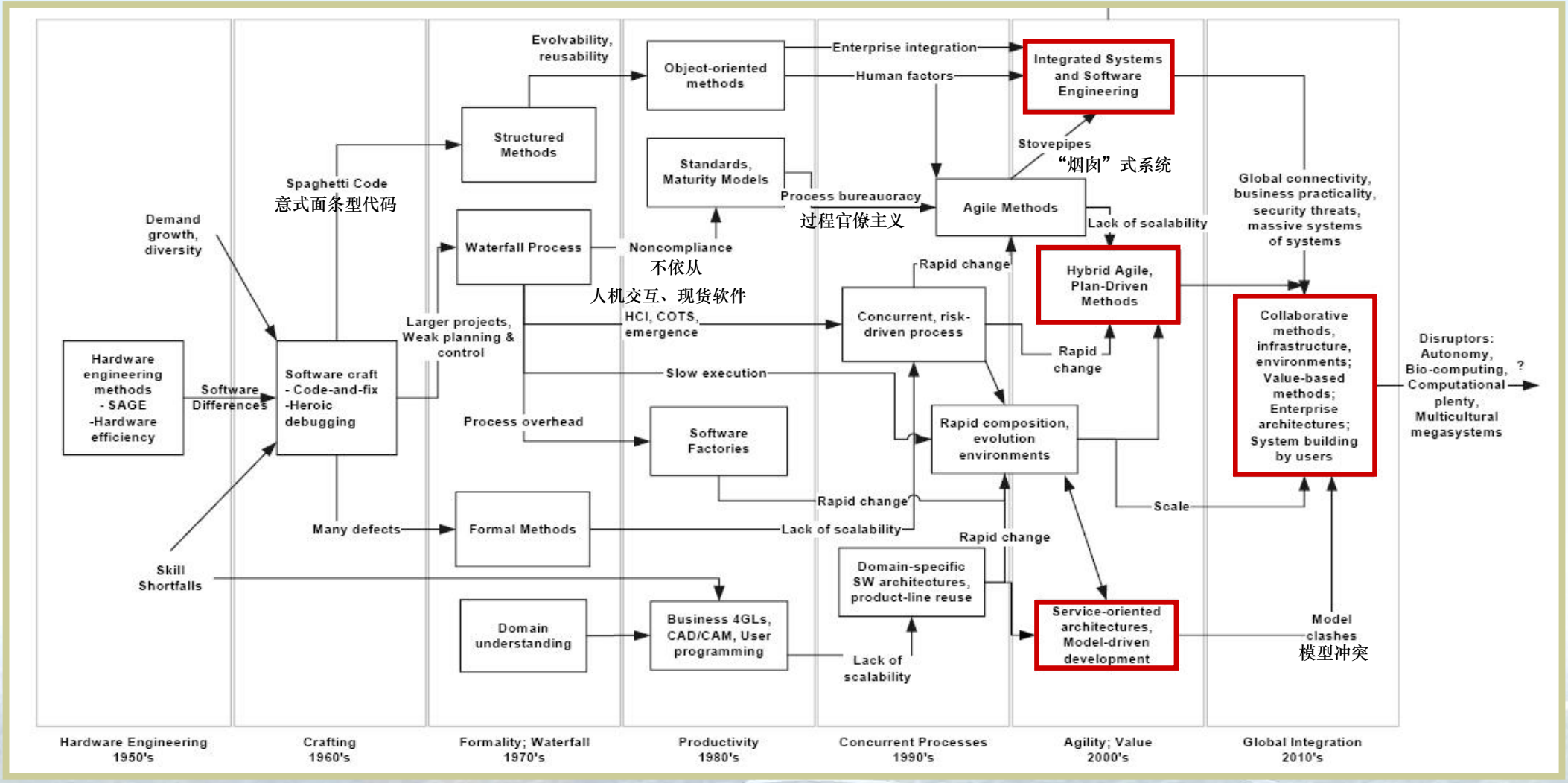


Figure 6. A Full Range of Software Engineering Trends

这篇论文中的一张图值得认真琢磨

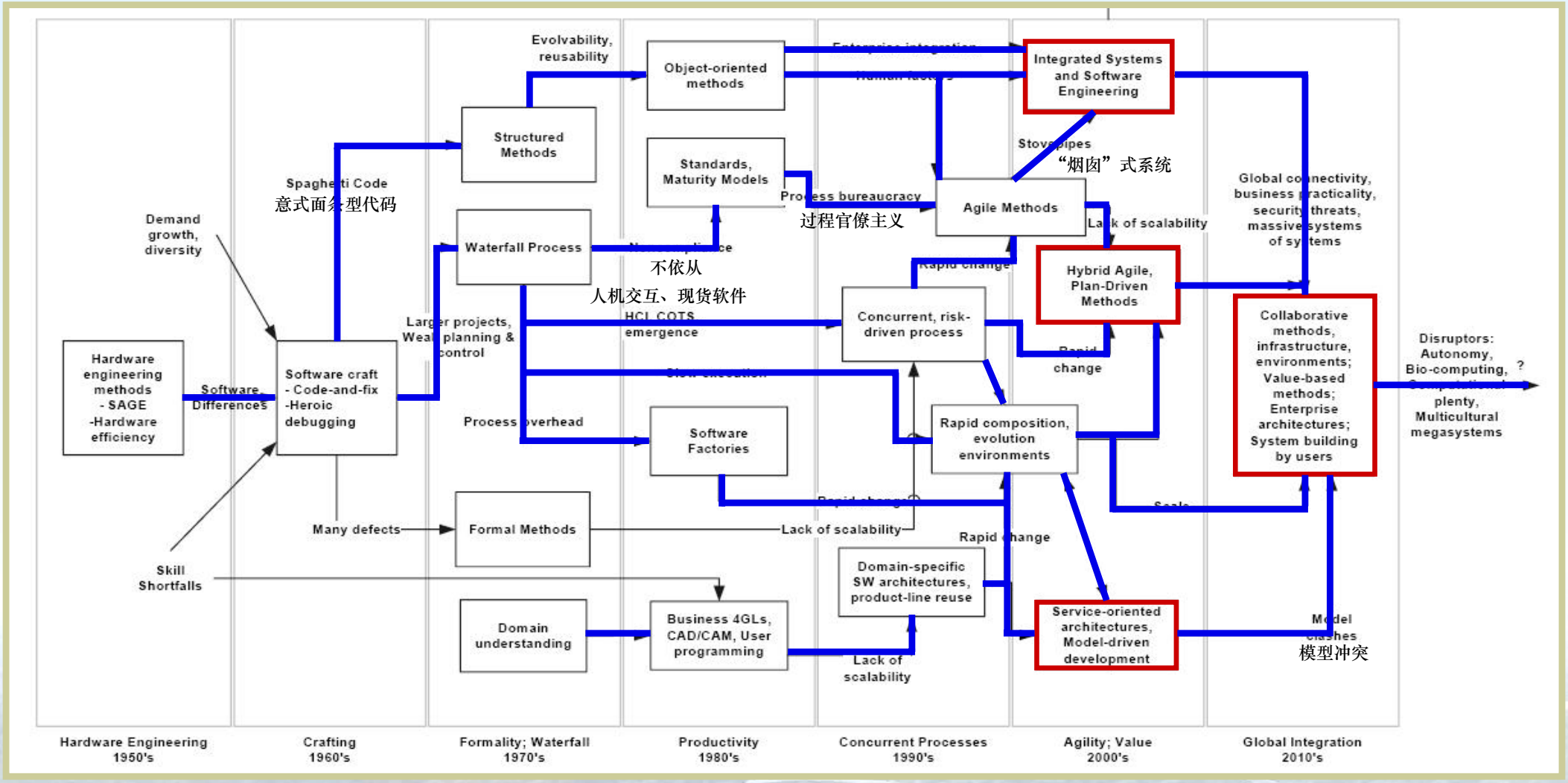


Figure 6. A Full Range of Software Engineering Trends

5.1 对象技术发展过程及其分析

5.1.4 对发展过程的分析

- 在 OO 技术发展的前期，起主导作用的是大学和大公司的开放实验室，其研究的成分大于应用的成分；近年来，推动技术发展的主要是标准化组织和占领市场的大公司产品，其应用的成分大于研究的成分。
- 产生这种变化的关键因素，是由于应用的主体变成了大范围的、边界不可预知的分布对象计算系统，其代表是互联网和 Java。

5.1 对象技术发展过程及其分析

5.1.4 对发展过程的分析

- 有人认为，对象技术的未来在很大范围内取决于正在开发的产品及其厂商是否能够成功。过去，面向对象的产品多为实时系统、工程应用（*特别是 CAD*）、**GUI** 等，而现在大量的厂商在提供各种各样的面向对象产品，如多平台的、分布的、使用多种语言的电子商务系统和信息处理系统等，而 **Java** 则在加速这种趋势的发展。
- 现在的应用要求比过去更迫切、范围更广；现在的研究难度比过去更大；新的研究成果的推广更多地受到了标准化的制约。

5.2 从OOPSLA看发展趋势和研究重点

面向对象领域权威的国际学术会议

- OOPSLA（*ACM Conference on Object Oriented Programming, Systems, Languages and Applications*）
- ECOOP（*European Conference on Object-Oriented Programming*）
- TOOLS（*International Conference on Technology of Object Oriented Languages and Systems*）

5.2 从OOPSLA看发展趋势和研究重点

2010年 OOPSLA的重要变化：SPLASH

General Chairs:

William R. Cook	*University of Texas at Austin*
Siobhán Clarke	*Trinity College Dublin*

Cook的开场白：

- *It is my pleasure to welcome you to SPLASH (Systems Programming Languages and Applications: Software for Humanity), the next step in the evolution of the well-known OOPSLA conference. SPLASH is the premier forum for practitioners, researchers, educators, and students who are passionate about improving the state of the art and practice in the development of software systems and applications through improved programming tools and languages.*

5.2 从OOPSLA看发展趋势和研究重点

2010年 OOPSLA的重要变化：SPLASH

- *SPLASH is a new name for the overall OOPSLA conference, which includes workshops, panels, tutorials, co-located conferences, posters, and a doctoral symposium. The OOPSLA name is being retained for the technical research track that is the core of SPLASH. It would have been easier to just rename OOPSLA to be SPLASH, but that would lose continuity with the strong OOPSLA brand. As a result, we have adopted a phased approach where both names will be used for the foreseeable future.*

5.2 从OOPSLA看发展趋势和研究重点

2010年 OOPSLA的重要变化：SPLASH

- *Although SPLASH/OOPSLA has its origin in object technologies, SPLASH is no longer explicitly tied to object-oriented programming. There is an implicit connection, however, since most modern software development incorporates or builds on ideas from object-oriented programming. From its inception, OOPSLA has incubated（孵化） new technologies and practices. Dynamic compilation and optimization, software patterns, refactoring, aspect-oriented software development, agile methods, service-oriented architectures, and model-driven development (to name just a few) all have roots in OOPSLA. SPLASH 2010 continues and strengthens this tradition. SPLASH has as its foundation the most successful software development theories and practices, yet is always striving to find new and better techniques which will define the future of software development.*

5.2 从OOPSLA看发展趋势和研究重点

2010年 OOPSLA的重要变化：SPLASH

- *SPLASH is pleased to host a range of co-located conferences. Onward! is more radical, more visionary, and more open to new ideas, allowing it to accept papers that present strong arguments even though the ideas in the paper may not be fully proven. The Dynamic Languages Symposium (DLS) discusses dynamic languages, including scripting languages. The Pattern Languages of Programming (PLoP) conference explores patterns of software and effective ways to present them. The International Lisp Conference (ILC) is focused on Lisp, a language with a great history and future. The Educators' and Trainer's Symposium and Doctoral Symposium focus on the essential task of educating the next generation of software developers and researchers.*

5.2 从OOPSLA看发展趋势和研究重点

2010年 OOPSLA的重要变化：SPLASH

- *In the end, SPLASH is about people, not technology. While SPLASH inherits SPLA from OOPSLA, it adds a new twist on the end: Software for Humanity. While this may seem an afterthought（马后炮）, I have come to realize over the last year that it is the most important idea in the new name. Our community is strong and diverse. Even as we promote diverse technologies, we share deep values that enable us to work together. One is the simple idea that software can improve the daily lives of humans all around the planet. Like any technology, software has the potential for great benefit and also great harm. Let's try to use it for good.*

5.3 信息技术的泛在化

- 泛在化（*Ubiquitous*）这个概念是**1980**年代中期由施乐研究中心的**Mark Wieser**提出的，他的一句名言是：“最好的技术是那些融入日常工作和生活，成为不可分割的部分，最后自身消失的技术。”
- 信息技术的泛在化主要体现在泛在计算和泛在网络上，也就是无所不在的计算和无所不在的网络。我们过去提的是**E**世界—电子信息世界，现在新的提法是**U**世界—泛在的信息世界。
- 就信息革命的几个关键技术因素而言，计算机技术和微电子技术早就“泛在”了，通信技术也已经“泛在”了，软件技术自然早就“泛在”了，而传感器技术的“泛在”则取决于物联网的进展，看来也是迟早的事。

将来我们靠什么生存？只有不断迎接新的挑战。

大数据

- 大数据（*big data*）是通常用来形容一个公司创造的大量非结构化和半结构化数据。
- 大部分非结构化数据存在于文本文件中，它至少占据了企业数据的80%。
- 2010年全球信息产业中心发布的信息统计结果显示，美国每天使用的信息量为3.6ZB。
  - $1ZB = 2^{10}EB = 2^{20}PB = 2^{30}TB = 2^{40}GB \approx 10^{12}GB$（1万亿GB）
- 尽管大数据并没有指定某个特定数量，这个词常在说到PB和EB级的数据时用到。
- 以管理结构化数据见长的关系数据库在面对大数据时力不从心。
- 如果大数据得不到有效管理，进而得不到充分利用，我们可能不得不花大代价去保存海量的“食之无法下口，弃之可惜”的信息。

大数据

- 大数据的例子:
  - 网络日志，**RFID**，传感器网络，社会网络，社会数据，互联网文本和文件;
  - 互联网搜索索引;
  - 呼叫详细记录，天文学、大气科学、基因组学、生物地球化学、生物和其他复杂和/或跨学科的科研、军事侦察、医疗记录;
  - 视频监控记录，视频档案;
  - 大规模的电子商务 。
- 适用于大数据的技术，包括大规模并行处理（*MPP*）数据库、分布式文件系统、分布式数据库、云计算平台、互联网和可扩展的存储系统。
- 这个领域目前在技术上领先的是**Google**。

云计算

- 云计算（*cloud computing*）是基于互联网的相关服务的增加、使用与交付模式，通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源。云是网络、互联网的一种比喻说法。过去在图中往往用云来表示电信网，后来也用来表示互联网和底层基础设施的抽象。
  - 狭义云计算指IT基础设施的交付和使用模式，指通过网络以按需、易扩展的方式获得所需资源；
  - 广义云计算指服务的交付和使用模式，指通过网络以按需、易扩展的方式获得所需服务。这种服务可以是IT和软件、互联网相关，也可是其他服务。它意味着计算能力也可作为一种商品通过互联网进行流通。

5.4 物联网带来的冲击

- 物联网（*the internet of things*）的概念是**1999**年提出来的。顾名思义，物联网就是“物物相连的互联网”。这有两层意思：第一，物联网的核心和基础仍然是互联网，是在互联网基础之上延伸和扩展的一种网络；第二，其用户端从计算机延伸扩展到了物品，以实现物品与物品之间的通信和信息交换。
- 物联网是通过射频识别（*radio frequency identification*，*RFID*）装置、红外感应器、全球定位系统（*GPS*）、激光扫描器等信息传感设备，把物品与互联网相连接，进行通信和信息交换，以实现智能化识别、定位、跟踪、监控和管理的一种网络。

5.4 物联网带来的冲击

2011年第三届全国高校软件工程专业教育年会

- 李未院士：物联网、云计算和群体软件工程
- 严成文（*IBM Rational 中国软件开发中心总经理*）：**Software, Everyware.**（注意：不是 *everywhere*）
  - **1980年代汽车中有5%与软件相关，2010年有近50%与软件相关。**
  - **F35有600万行代码，GM Volt（通用公司的一款汽车）有1000万行代码。**
- **2011年8月27日《华尔街日报》文章：软件将吃掉整个世界?**

5.4 物联网带来的冲击

李未院士：物联网、云计算和群体软件工程

- 物联网：一个面向特定领域或行业的、拥有超量数据的复杂信息应用系统。
  - 物联网五大功能（括号中的数量级表明了其超量特征）：感知（10^7）；处理（10^{12}）；通信（10^{12}）；存储（10^{18}）；控制（10^7）。
- 云计算：一种基于互联网的大众参与的计算模式。
  - 实质：对用户屏蔽有关计算、存储、通信和控制的底层操作细节。
  - 云计算为物联网的实现，特别是其软件系统的实现提供了一种解决方案。

5.4 物联网带来的冲击

李未院士：物联网、云计算和群体软件工程

- 物联网建设对软件工程的挑战：
  - 物联网是涉及一个行业信息的应用系统，具有超（超大规模系统）、变（不间断的、持续演化和部署）、散（分散性）的特点。
  - 物联网应用系统的规模：远超过6000万行的Vista。
  - 传统软件工程：精英化（软件专业骨干）；计划性（预先规划需求，自顶向下开发）；封闭性（开发过程不对外开放）。
  - 软件和软件工程面临的挑战：有硬件，缺软件；有软件，少数据；有数据，难共享。

5.4 物联网带来的冲击

李未院士：物联网、云计算和群体软件工程

- 软件工程将进入新的历史时期:
  - 结构化软件工程
  - 面向对象软件工程
  - 敏捷软件工程
  - 面向Agent的软件工程（2000）
  - 群体软件工程（2010）

5.4 物联网带来的冲击

李未院士：物联网、云计算和群体软件工程

- 群体软件工程
  - 目标：用于有效开发以云计算为解决方案的可靠的超量信息系统。
  - 理念：开发过程从封闭到开放；开发软件从精英到大众；开发组织从工厂到社群；开发方法从机器工程（机械工程）到社会工程。
  - 开发原则：使用者即设计者；使用者即开发者；使用者即维护者。
  - 群体软件工程的内容：群体软件结构；屏蔽原理；群件原理；用户身份的多重性原理；开发者竞争选择原理。

5.4 物联网带来的冲击

李未院士：物联网、云计算和群体软件工程

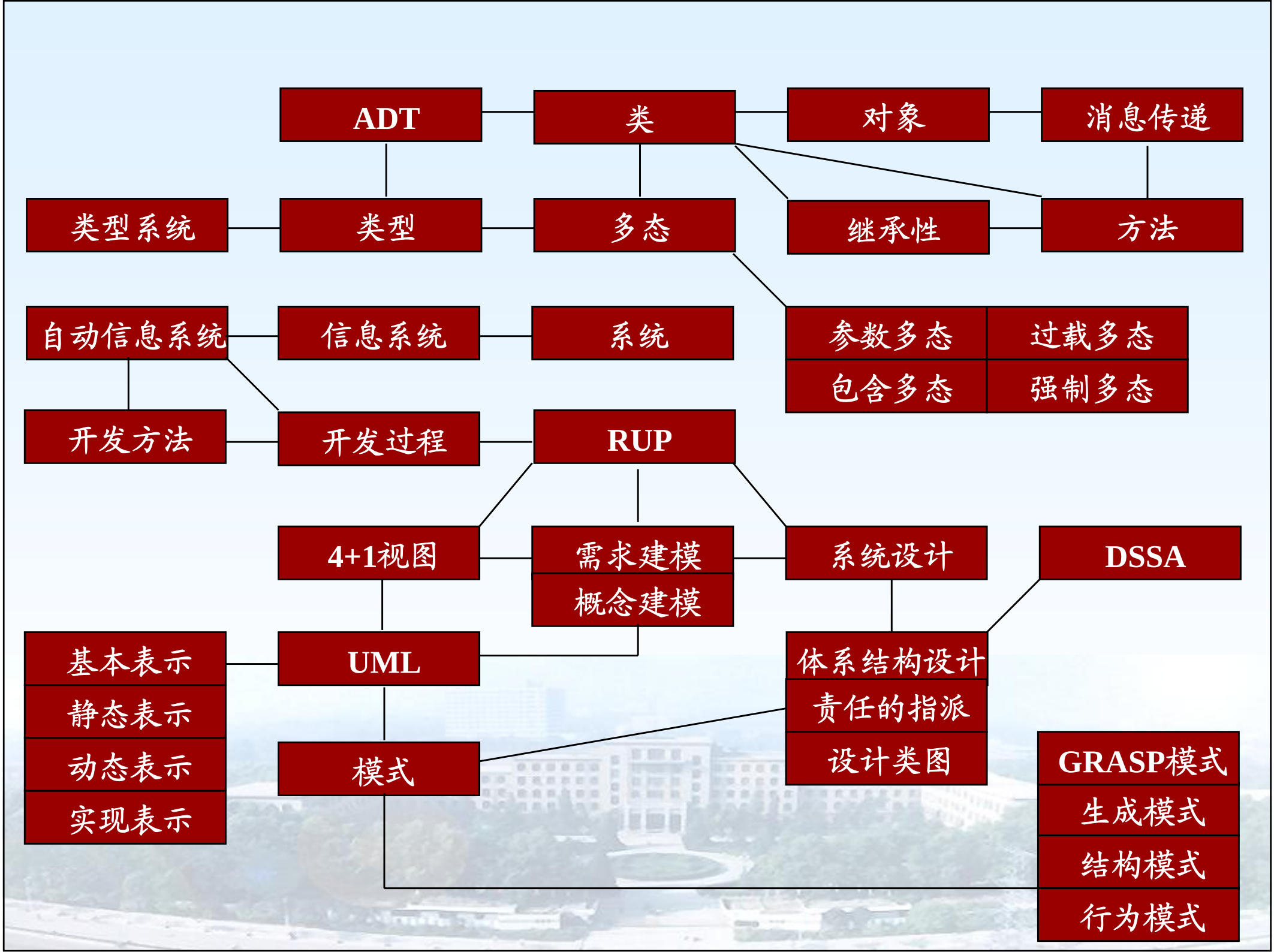
- 群体计算雏形：
  - 人肉搜索
- 群体软件开发实例：
  - Apple的App Store；Google的Andriod Market。
  - 社会群体广泛参与，3年时间拥有了47万个应用软件。
- 李未、杨学军等六人成立了面向超量信息系统的群体软件工程研究小组（北航、国防科大），正在进行群体软件工程的研究。

5.4 物联网带来的冲击

李未院士：物联网、云计算和群体软件工程

- 我的思考：
  - 我们可能要面对物联网带来的“漫山遍野”的软件、“三教九流”的开发人员、“云里雾里”的软件结构。
  - 这种趋势很可能不以我们的意志为转移。
  - 这种趋势很可能会颠覆很多现有的理念和技术。
  - 即使是颠覆，也不会是无继承性的。
  - 要做好准备，没准儿机会就在这里。

课程总结


ADT
类
对象
消息传递
类型系统
类型
多态
继承性
方法
自动信息系统
信息系统
系统
参数多态
过载多态
包含多态
强制多态
开发方法
开发过程
RUP
4+1视图
需求建模
概念建模
系统设计
DSSA
基本表示
静态表示
动态表示
实现表示
UML
体系结构设计
责任的指派
设计类图
模式
GRASP模式
生成模式
结构模式
行为模式

本课程到此结束，欢迎大家提出意见与建议。

谢谢大家！