电子自动计算-计算机系统

战德臣

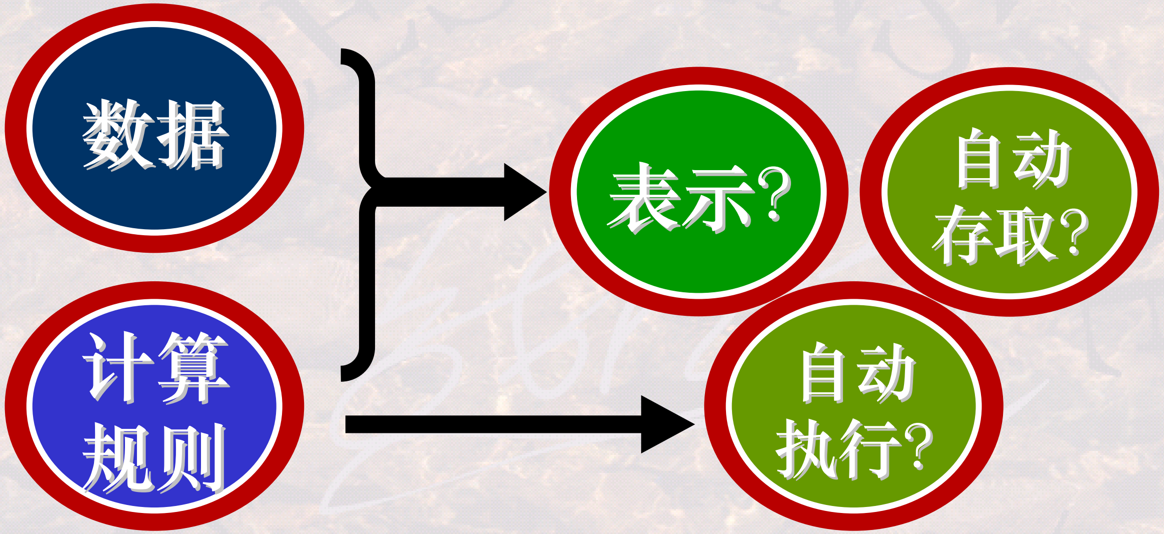
哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

电子自动计算-计算机系统
(1) 自动计算需要解决的问题?

战德臣 教授



二进制 vs. 十进制

由 元器件 到 系统

计算机系统要解决的几个问题

- “控制与计算”----微处理器
- “输入”----如何将外部信息输入到计算机中?
- “输出”----如何将计算机中信息输出到外界(显示或打印)?
- “永久存储与临时存储”----如何将计算机中的信息永久保存或临时保存?

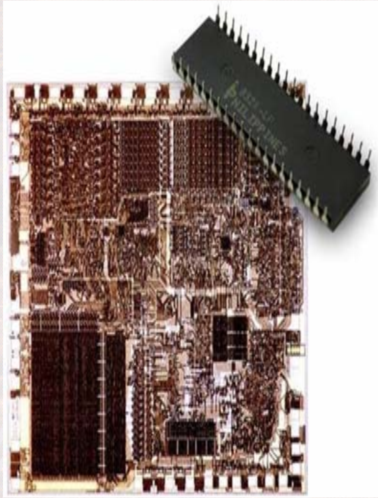
电子自动计算-计算机系统

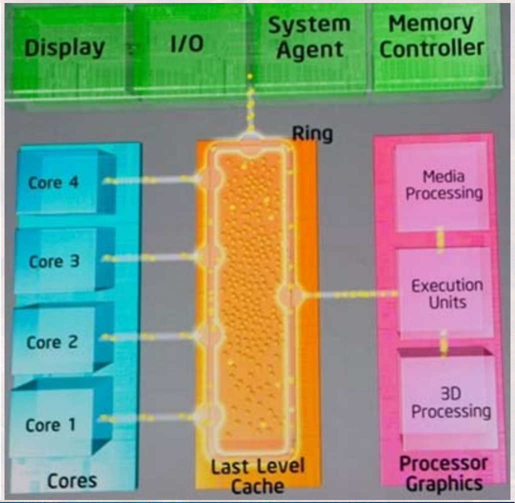
(3) 控制与计算-微处理器-计算机系统的核心?

战德臣 教授

微处理器的发展

- ◆字长：8位➔16位➔32位➔64位
- ◆主频：几MHz➔几百MHz➔几GHz
- ◆晶体管数量：几万➔几百万➔几亿颗
- ◆功能/规模：微处理器➔微处理器+协处理器(浮点运算)➔微处理器+图形处理单元GPU➔微处理器+3D处理器+多媒体处理器➔多核微处理器





8086/8088	Intel 80286	Intel 80386	Intel 80486	Pentium	Pentium II	Pentium III	双核 CORE	八核-十核 CORE i 系列
1974	1982	1985	1989	1993	1997	1999	2005	2010

电子自动计算-计算机系统
(4) 怎样解决大规模快速存储问题?

战德臣 教授

存储设备

发展水平

纳米存储/量子存储

固态硬盘
USB Removable disk

FlashRAM

光盘存储(CD-ROM,
CD R/W, DVD)

磁盘存储(硬盘与软盘)

半导体存储(ROM/RAM)

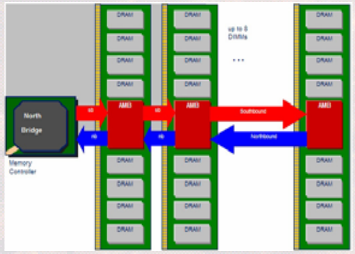
磁带/磁芯/磁鼓存储

汞延迟线

存储设备

类别

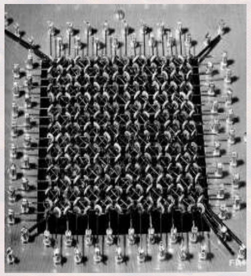
- 体积越来越小
- 容量越来越大
- 访问速度越来越快
- 可靠性越来越高
- 功耗越来越低
- 持久性越来越好



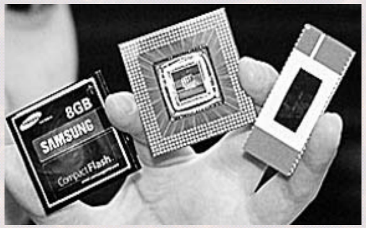
DRAM半导体存储器芯片

固态硬盘

U-Disk



磁芯存储器



FlashRAM存储器

输入设备

发展水平

感知输入
(接触式感知输入
非接触式感知输入)

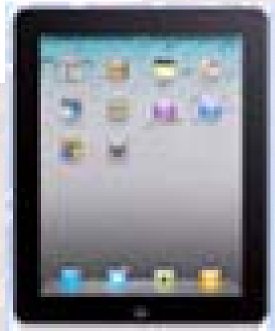
鼠标输入
(点输入)

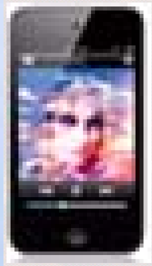
键盘输入
(符号输入)

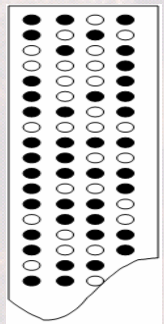
穿孔纸带
(0/1输入)

输入设备

类别







输出设备---显示及显示控制

发展水平

3D显示器：3维图形

数字显示器：高清图形
(液晶、等离子技术)

CRT: 数字光栅扫描显示器
(基于内存的显示: 输出图形)

CRT: 字符发生器
(向量式模拟显示器: 输出字符)

CRT：阴极射线管
(模拟显示器：黑白与彩色)

输出设备
(显示器)

类别

多显示卡并联

- 分辨率越来越高
- 颜色越来越逼真
- 显示速度越来越快(屏幕刷新速度和图形处理速度)
- 越来越薄，越清晰
- 可视角度越来越接近平角

GPU芯片

电子自动计算-计算机系统
(7) 计算机系统各主要部件的发展趋势是什么?

战德臣 教授

计算机系统的发展

发展水平 ↑ / 类别 →

微处理器（字长;主频;晶体管数目）

- (64位; >10G; >10亿颗)
- (64位; 10G; 3亿颗)
- (32位; 1G;2800万颗)
- (32位; 60M;300万颗)
- (16位; 20M;10万颗)
- (8位; 5M;1万颗)
- (字长;主频;晶体管数目)

存储设备

- 纳米存储/量子存储
- 固态硬盘
- USB Removable disk
- FlashRAM
- 光盘存储(CD-ROM, CD R/W, DVD)
- 磁盘存储(硬盘与软盘)
- 半导体存储(ROM/RAM)
- 磁带/磁芯/磁鼓存储
- 汞延迟线

输入设备

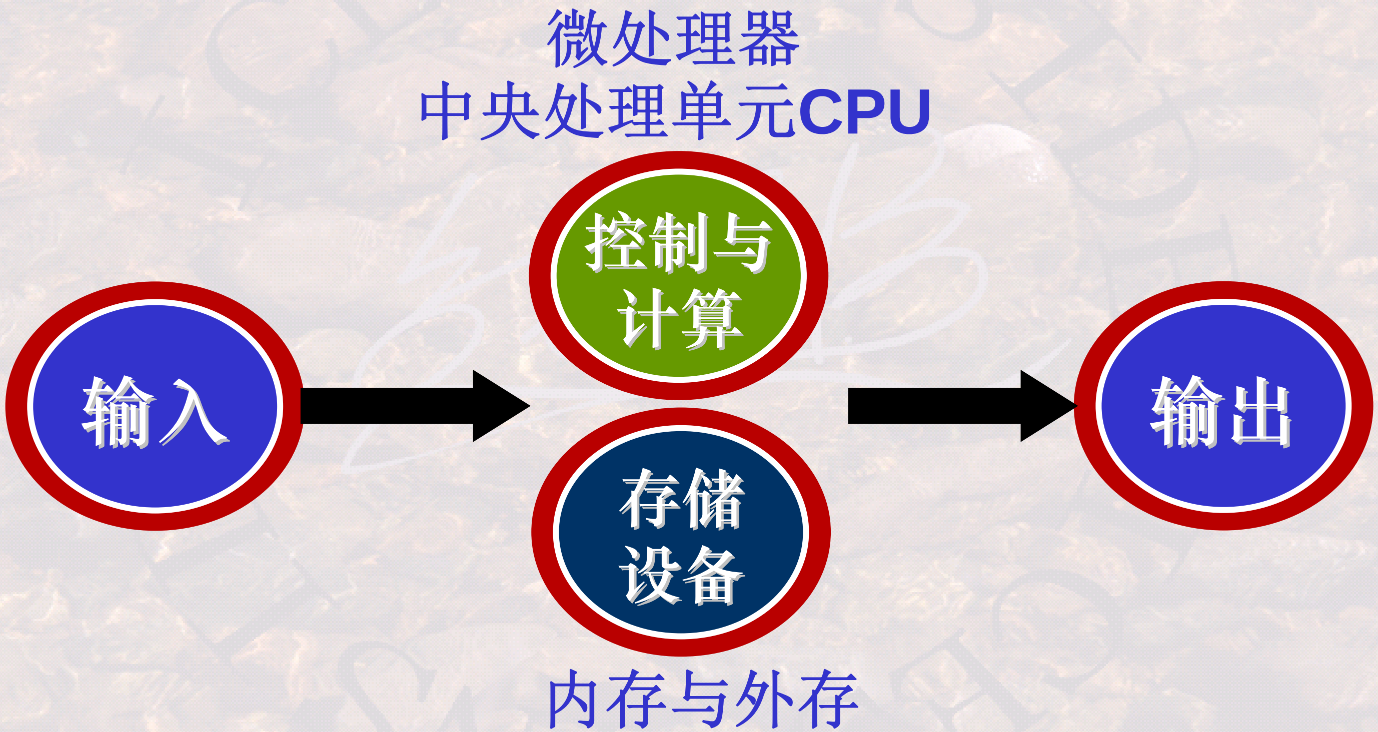
- 感知输入 (接触式感知输入 非接触式感知输入)
- 鼠标输入 (点输入)
- 键盘输入 (符号输入)
- 穿孔纸带 (0/1输入)

输出设备 (显示器)

- 3D显示器
- 数字显示器 (液晶、等离子技术)
- CRT: 数字光栅扫描显示器 (基于内存的显示)
- CRT: 字符发生器 (向量式模拟显示器)
- CRT：阴极射线管 (模拟显示器：黑白与彩色)

电子自动计算-计算机系统
(8) 小结

战德臣 教授



计算系统之发展趋势

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

微型化：可嵌入、可携带

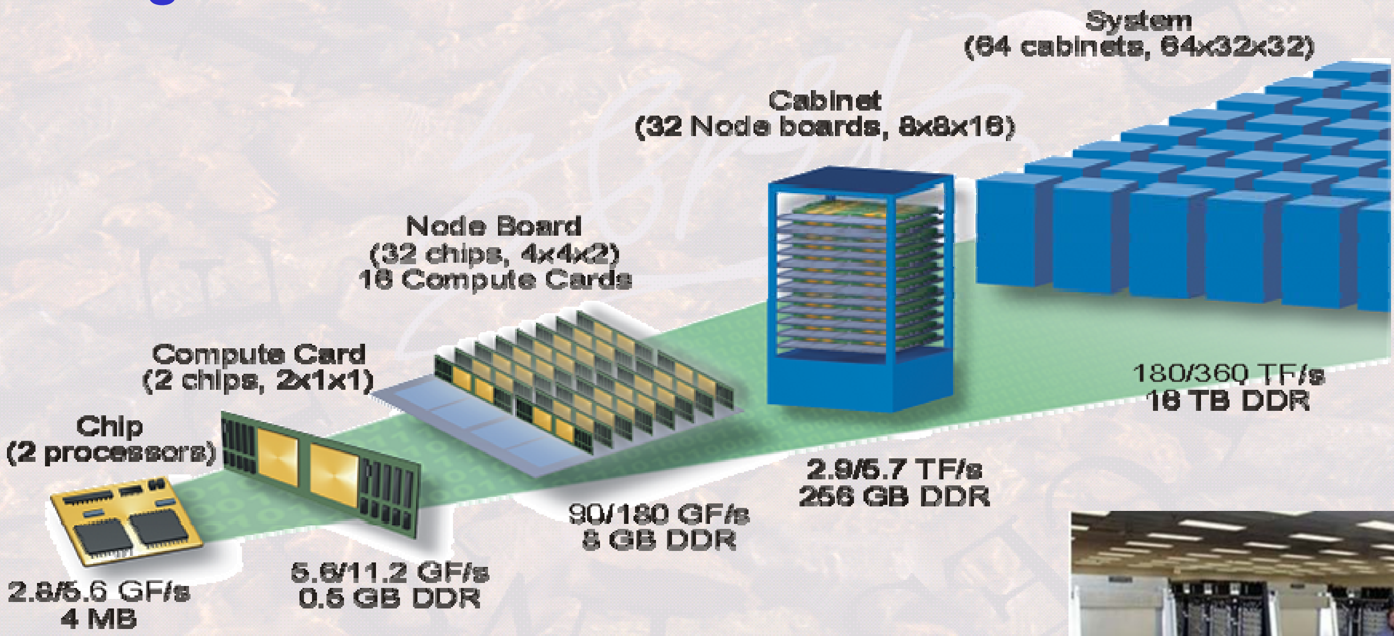
平板电脑-Apple IPAD

世界上最小台式电脑---- 如同拇指大小

智能手机

大型化：可进行大规模、复杂计算

IBM-BlueGene(蓝色基因): Milestone of an Intelligent Machine



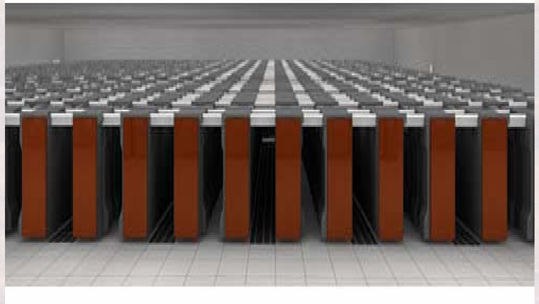
A massively parallel supercomputer using tens of thousands of embedded PowerPC processors supporting a large memory space With standard compilers and message passing environment

计算系统之发展趋势
(2) 大型化趋势是怎样的?

战德臣 教授

大型化：可进行大规模、复杂计算

超级计算机500强



2010.11, 超级计算机**500**强第一名：天河一号**A --** 中国

14336颗**Intel Xeon X5670 2.93GHz**六核心处理器

2048颗我国自主研发的飞腾**FT-1000**八核心处理器

7168块**NVIDIA Tesla M2050**高性能计算卡

总计**: 186368**个核心**, 224TB**内存。

实测运算速度可以达到每秒**2570**万亿次**(**这意味着，它计算一天，相当于一台家用电脑计算**800**年**)**

智能化

理解自然语言，具有自适应性，自主完成复杂功能

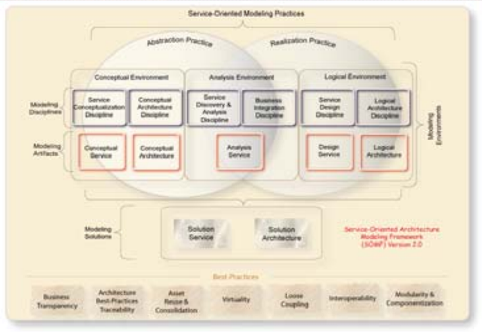
计算系统之发展趋势
(4) 网络化趋势是怎样的?

战德臣 教授

网络化

“未来互联网”-Future Internet

Internet of Services



For people

and enterprises

Internet of 3D Worlds

Internet of Things

Internet of Networks

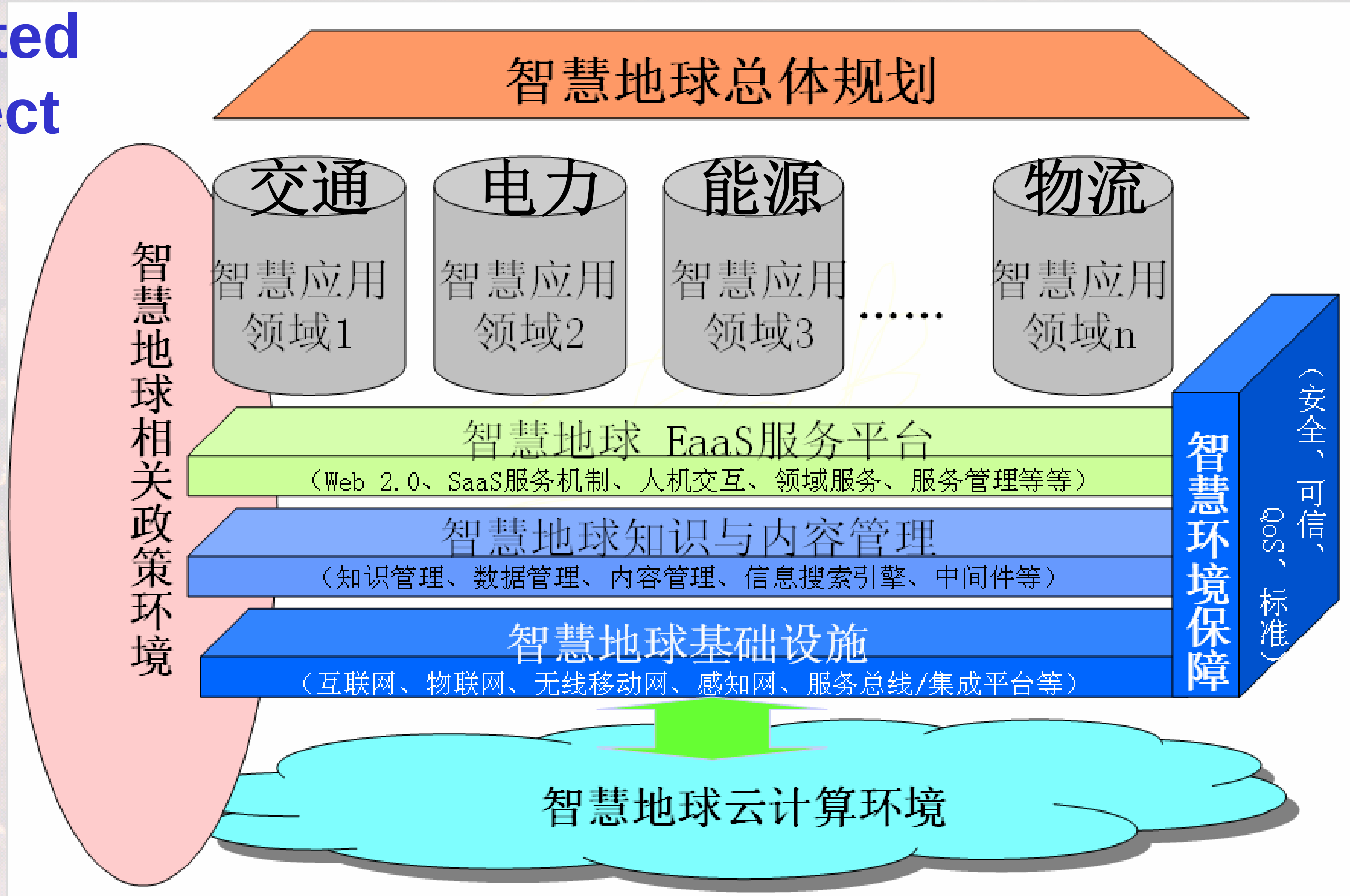
机-机相联，物-物相联，物-人相联，人-人相联

战德臣 教授

网络化

IBM提出智慧地球，**Smart Planet**。

- ◆**Instrumented**
- ◆**Interconnect**
- ◆**Intelligent**



什么是计算思维

战德臣

哈尔滨工业大学　教授.博士生导师

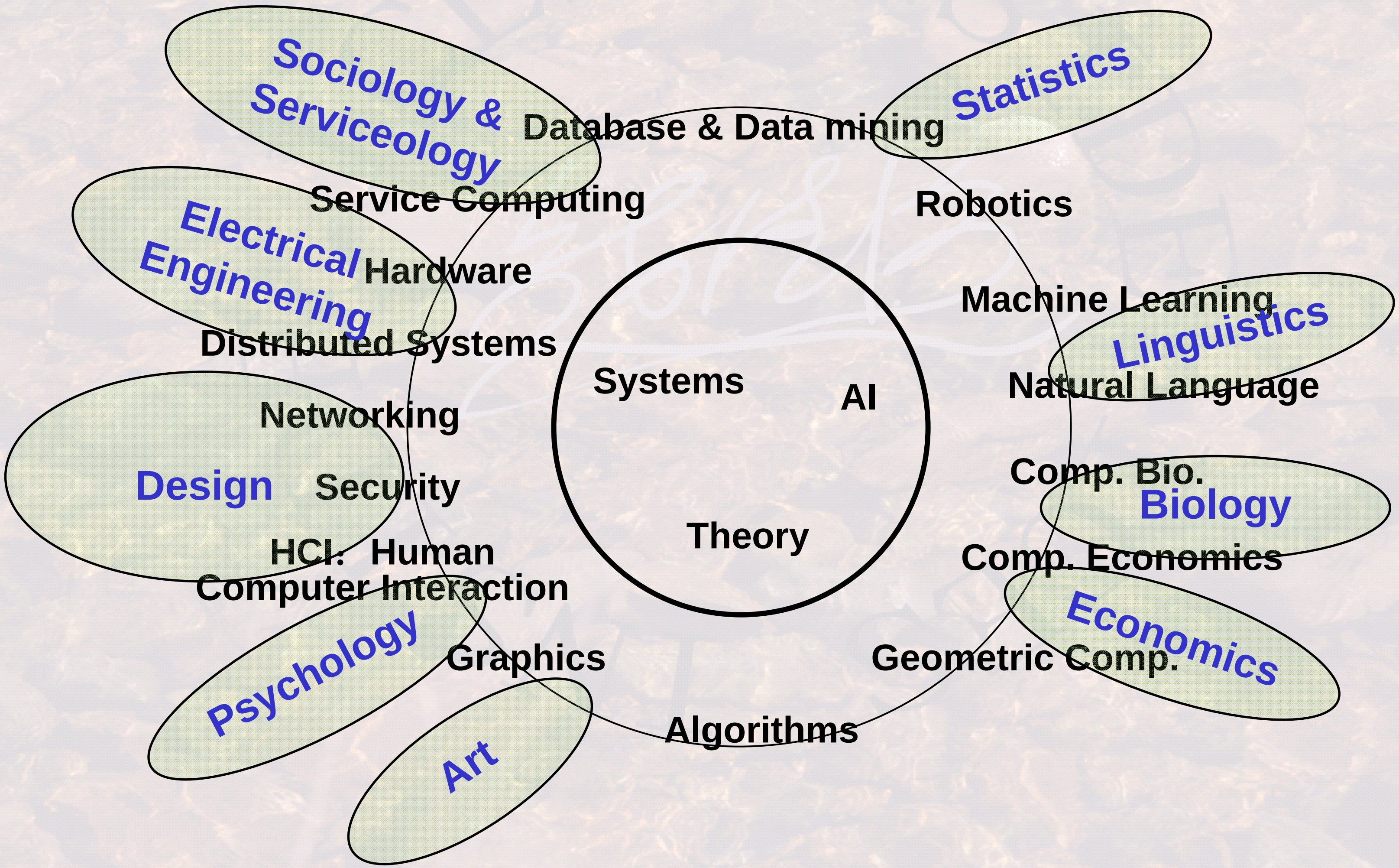
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

什么是计算思维?
(1) 为什么提出计算思维?

战德臣 教授

学科的发展，知识的膨胀



什么是计算思维?
(2) 计算思维的提出

战德臣 教授

计算思维，计算的伟大原理

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

Phil. Trans. R. Soc. A (2008) **366**, 3717–3725
doi:10.1098/rsta.2008.0118
Published online 31 July 2008

Computational thinking and thinking about computing

BY JEANNETTE M. WING*

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Computational thinking will influence everyone in every field of endeavour. This vision poses a new educational challenge for our society, especially for our children. In thinking about computing, we need to be attuned to the three drivers of our field: science, technology and society. Accelerating technological advances and monumental societal demands force us to revisit the most basic scientific questions of computing.

Keywords: computational thinking; abstraction; automation; computing; computable; intelligence

1. Computational thinking

Computational thinking is taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing[1] (Wing 2006).

Computational thinking is a kind of analytical thinking. It shares with mathematical thinking in the general ways in which we might approach solving a problem. It shares with engineering thinking in the general ways in which we might approach designing and evaluating a large, complex system that operates within the constraints of the real world. It shares with scientific thinking in the general ways in which we might approach understanding computability, intelligence, the mind and human behaviour.

(a) *Computing: abstraction and automation*

The essence of computational thinking is *abstraction*. In computing, we abstract notions beyond the physical dimensions of time and space. Our abstractions are extremely general because they are symbolic, where numeric abstractions are just a special case.

In two ways, our abstractions tend to be richer and more complex than those in the mathematical and physical sciences. First, our abstractions do not necessarily enjoy the clean, elegant or easily definable algebraic properties of mathematical

*wing@cs.cmu.edu

[1] By 'computing' I mean very broadly the field encompassing computer science, computer engineering, communications, information science and information technology.

One contribution of 19 to a Discussion Meeting Issue 'From computers to ubiquitous computing, by 2020'.

The Great Principles of Computing

Peter J. Denning

COMPUTING IS INTEGRAL to science—not just as a tool for analyzing data, but as an agent of thought and discovery.

It has not always been this way. Computing is a relatively young discipline. It started as an academic field of study in the 1930s with a cluster of remarkable papers by Kurt Gödel, Alonzo Church, Emil Post and Alan Turing. The papers laid the mathematical foundations that would answer the question "what is computation?" and discussed schemes for its implementation. These men saw the importance of automatic computation and sought its precise mathematical foundation. The various schemes they each proposed for implementing computation were quickly found to be equivalent, as a computation in any one could be realized in any other. It is all the more remarkable that their models all led to the same conclusion that certain functions of practical interest—such as whether a computational algorithm (a method of evaluating a function) will ever come to completion instead of being stuck in an infinite loop—cannot be answered computationally.

At the time that these papers were written, the terms "computation" and "computers" were already in common use, but with different connotations from today. Computation was taken to mean the mechanical steps followed to evaluate mathematical functions; computers were people who did computations. In recognition of the social changes they were ushering in, the designers of the first digital computer projects all named their systems with acronyms ending in "-AC", meaning automatic computer—resulting in names such as ENIAC, UNIVAC and EDSAC.

Computing may be the fourth great domain of science along with the physical, life and social sciences

At the start of World War II, the militaries of the United States and the United Kingdom became interested in applying computation to the calculation of ballistic and navigation tables and to the cracking of ciphers. They commissioned projects to design and build electronic digital computers. Only one of the projects was completed before the war was over. That was the top-secret project at Bletchley Park in England, which cracked the German Enigma cipher using methods designed by Alan Turing.

Many people involved in those projects went on to start computer companies in the early 1950s. Universities began offering programs of study in the new field in the late 1950s. The field and the industry have grown steadily into a modern behemoth whose Internet data centers are said to consume almost three percent of the world's electricity.

During its youth, computing was an enigma to the established fields of science and engineering. At first, computing looked like only the applied technology of math, electrical engineering or science, depending on the observer. However, over the years, computing provided a seemingly unending stream of new insights, and it defied many early predictions by resisting absorption back into the fields of its roots. By 1980 computing had mastered algorithms, data structures, numerical methods, programming languages, operating systems, networks, databases, graphics, artificial intelligence and software engineering. Its great technological achievements—the chip, the personal computer and the Internet—brought it into many lives. These advances stimulated more new subfields, including network science, Web science, mobile computing, enterprise computing, cooperative work, cyberspace protection, user-interface design and information visualization. The resulting commercial applications have spawned

Peter J. Denning is Director of the Cebrowski Institute for Innovation and Information Superiority at the Naval Postgraduate School in Monterey, California, and is a past president of ACM. Email: pjd@nps.edu

◆《Computational Thinking》

from CMU，周以真（Jeannette M. Wing），Communications of ACM, Vol.49, No.3, March 2006, Pages 33-35

◆**Computational thinking is a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science.**

◆**Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century.**

– Just like reading, writing, and arithmetic.

– Imagine every person knowing how to think like a computer scientist!

– Computational thinking is not just for other scientists, it's for everyone.

–Thinking like a computer scientist means more than being able to program a computer

◆计算思维的本质就是**抽象(Abstraction)**与**自动化(Automation)**，即在不同层面进行抽象，以及将这些抽象“机器化”。

国内学者/专家的观点

■计算思维是人类应具备的第三种思维

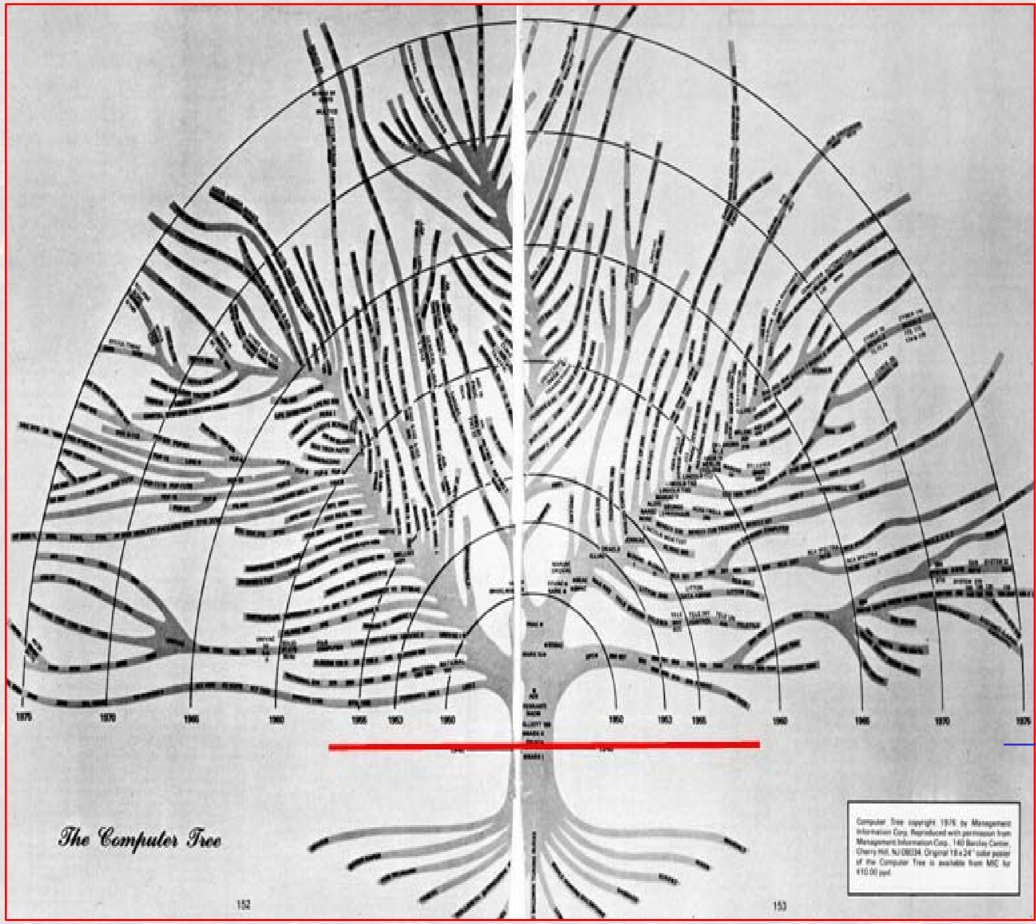
- 实验思维: 实验 ➔ 观察 ➔ 发现、推断与总结. ---观察与归纳
- 理论思维: 假设/预设 ➔ 定义/性质/定理 ➔ 证明. ---推理和演绎
- 计算思维: 设计, 构造 与 计算. ---设计与构造

计算思维关注的是人类思维中有关可行性、可构造性和可评价性的部分

当前环境下，理论与实验手段在面临大规模数据的情况下，不可避免地要用计算手段来辅助进行。

什么是计算思维?
(5) 从计算学科发展的角度看计算思维

战德臣 教授



科学研究的三大手段：
理论、实验和计算 ➔ 计算科学

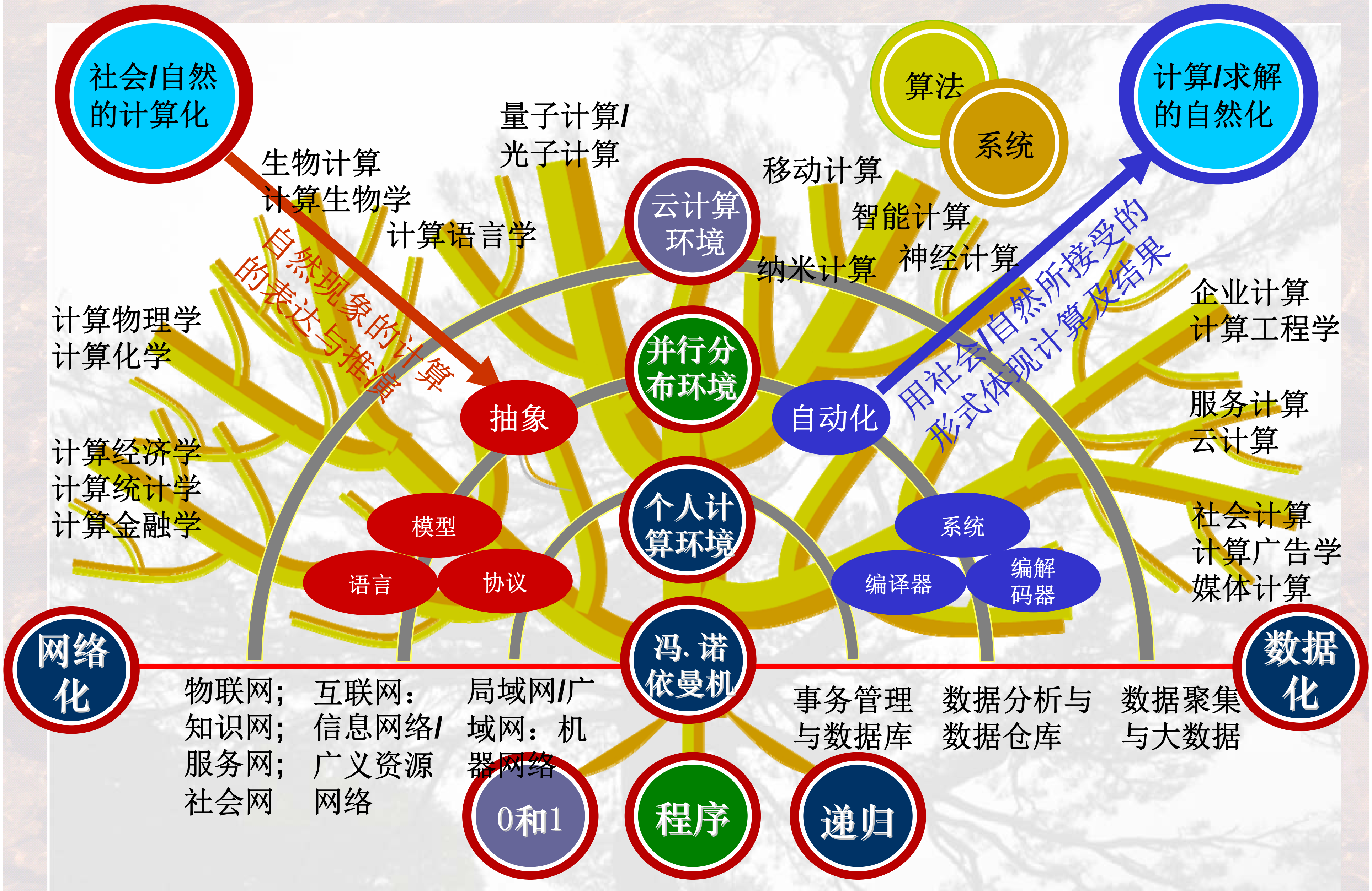
计算机科学与技术➔
软件工程、生物计算、信息安全、… …

构造各种新型“计算机器”
应用各种新型“计算机器”

构造传统“计算机器”

什么是计算思维?
(6) 大学计算思维教育空间---计算之树?

战德臣 教授



大学计算思维教育空间-计算之树

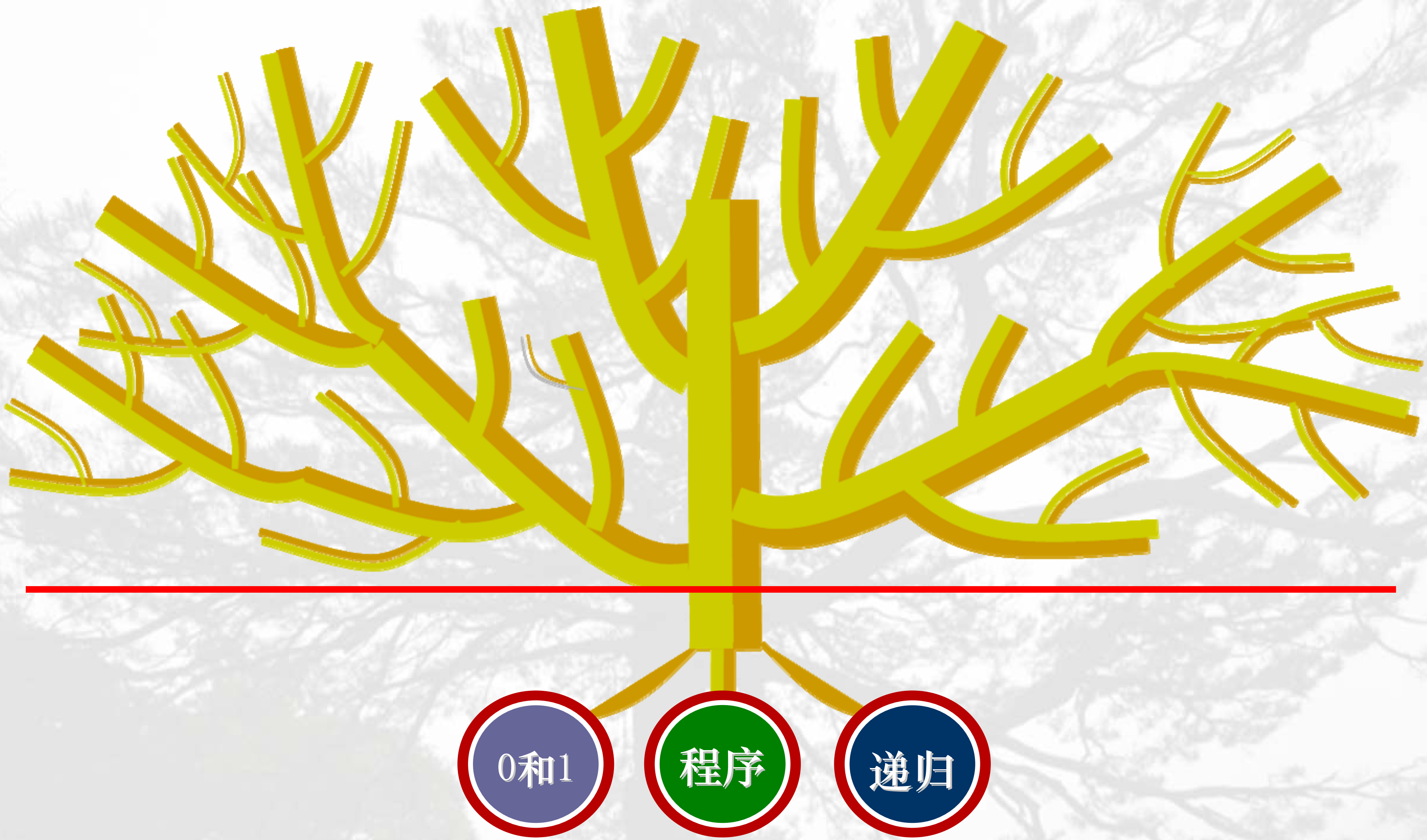
战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

计算之树的第一个维度—计算技术的奠基性思维



计算之树的第一个维度—计算技术的奠基性思维

■“0 和 1”思维--符号化→计算化→自动化

➢0和1是实现任何计算的基础；社会/自然与计算融合的基本手段； 0和1是连接硬件与软件的纽带；0/1是最基本的抽象与自动化机制。

■“程序”思维--千变万化复杂功能的构造、表达与执行

➢程序是基本动作(指令)的各种组合，是控制计算系统的基本手段

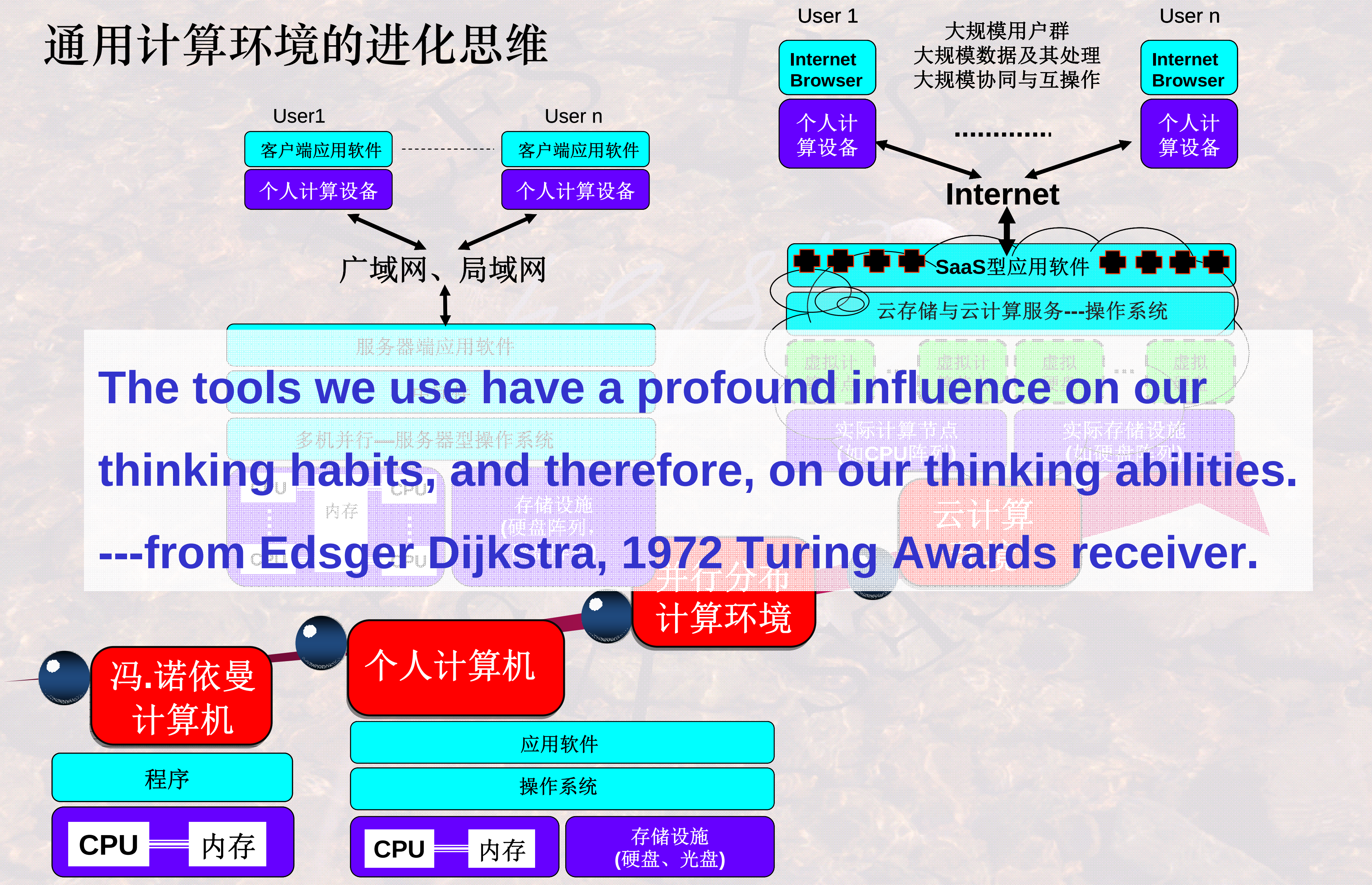
■“递归”思维--无限事物及重复过程的表达与执行方法

➢递归是最典型的构造程序的手段；递归函数是可计算函数的精确的数学描述； 递归函数是研究计算学科理论问题的基础

计算之树的第二个维度—通用计算环境的进化思维

ICES 战德臣 教授

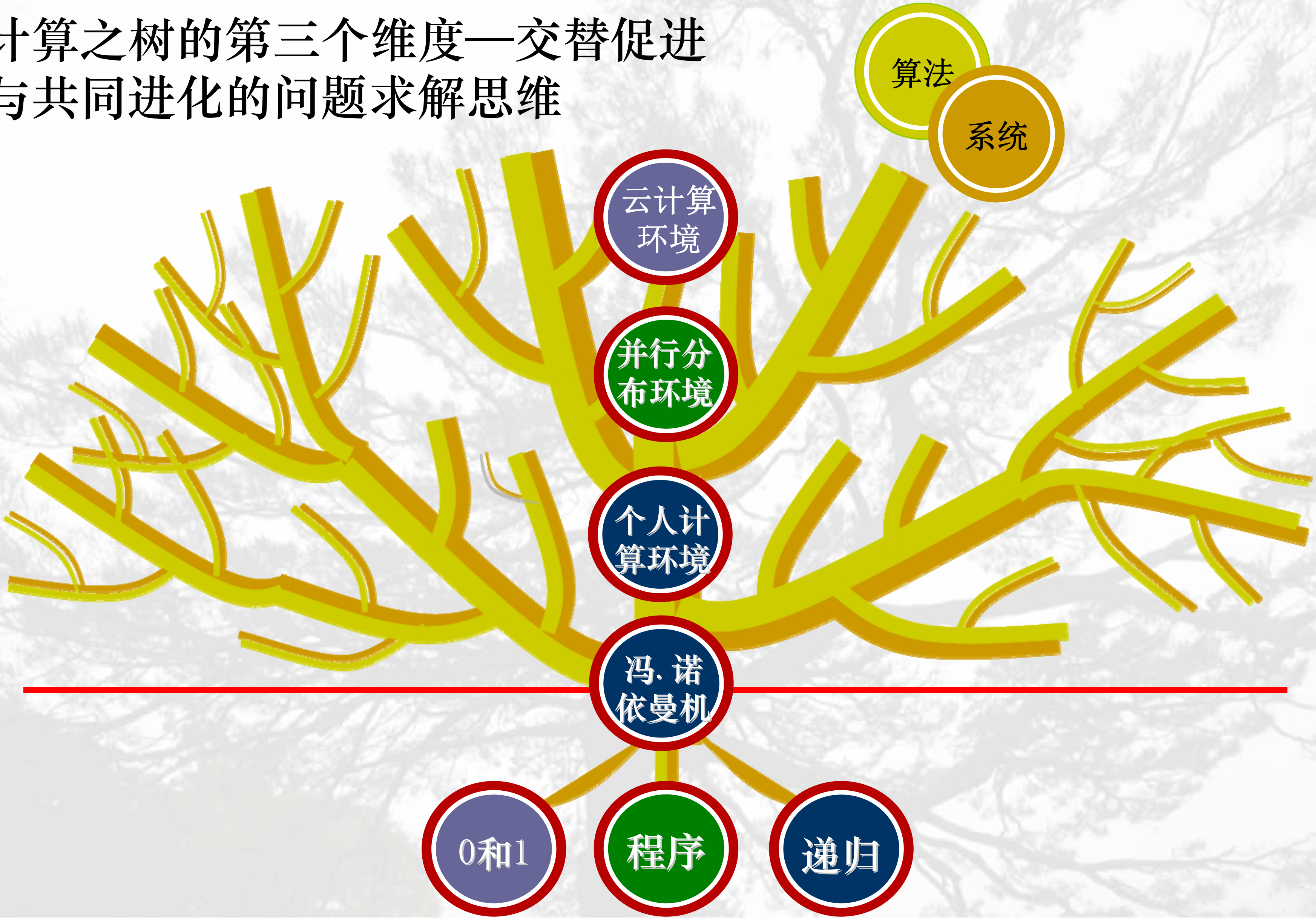
通用计算环境的进化思维



The tools we use have a profound influence on our thinking habits, and therefore, on our thinking abilities.

---from Edsger Dijkstra, 1972 Turing Awards receiver.

计算之树的第三个维度—交替促进与共同进化的问题求解思维



计算之树的第三个维度—交替促进与共同进化的问题求解思维

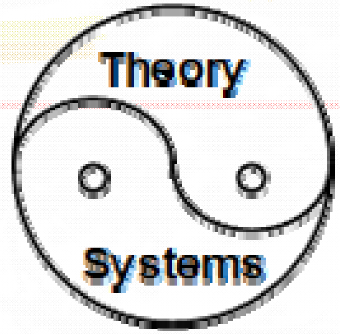
■“算法”：问题求解的一种手段—构造与设计算法

■算法是计算的灵魂；算法强调数学建模；算法考虑的是可计算性与计算复杂性；算法研究通常被认为是计算学科的理论研究。

■“系统”：问题求解的另一种手段—构造与设计系统

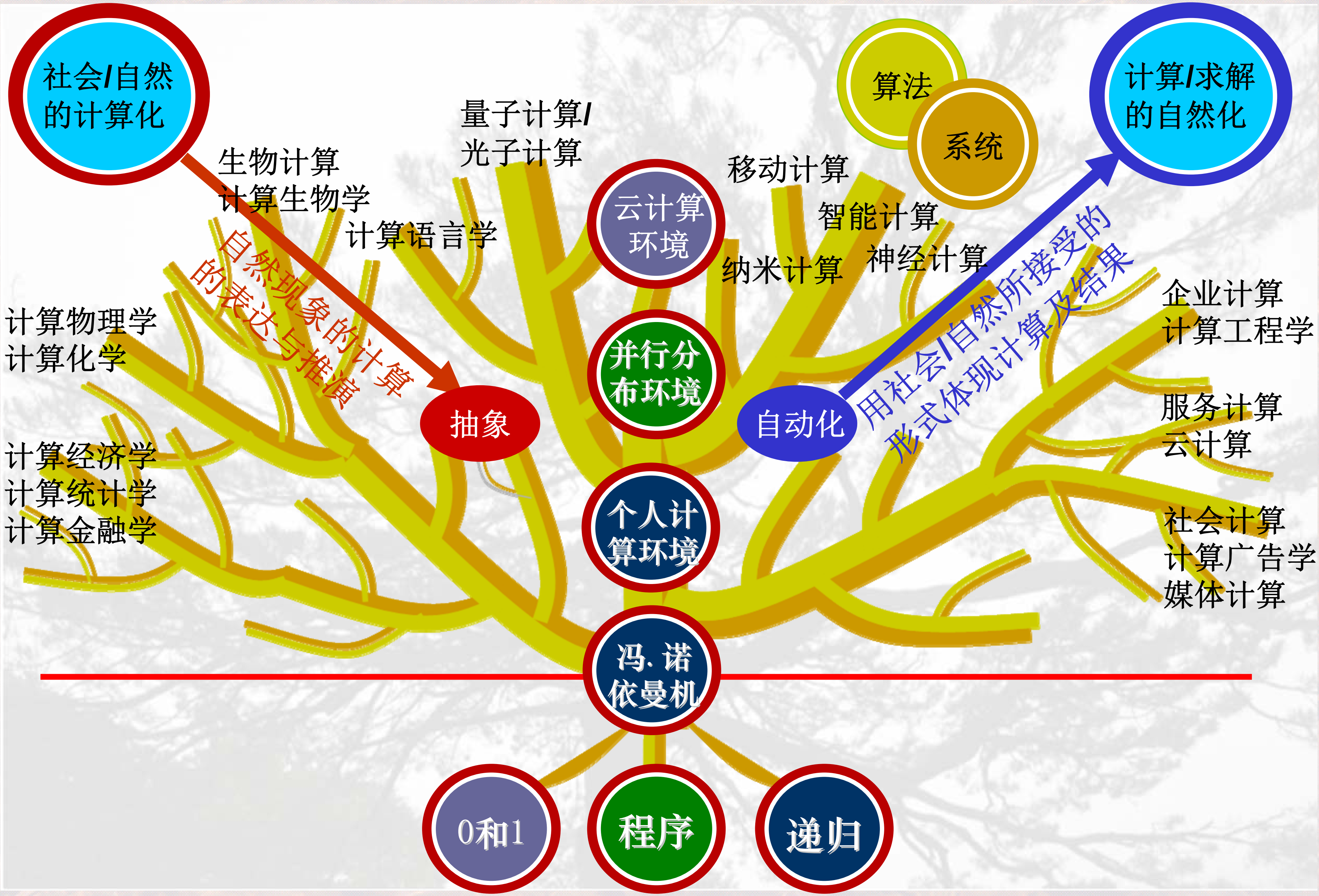
➢系统是改造自然的手段；系统还强调非数学建模；系统考虑的是如何化复杂为简单(使其能够被做出来)；系统还强调结构性、可靠性、安全性等。

系统是龙，算法是睛，画龙要点睛。



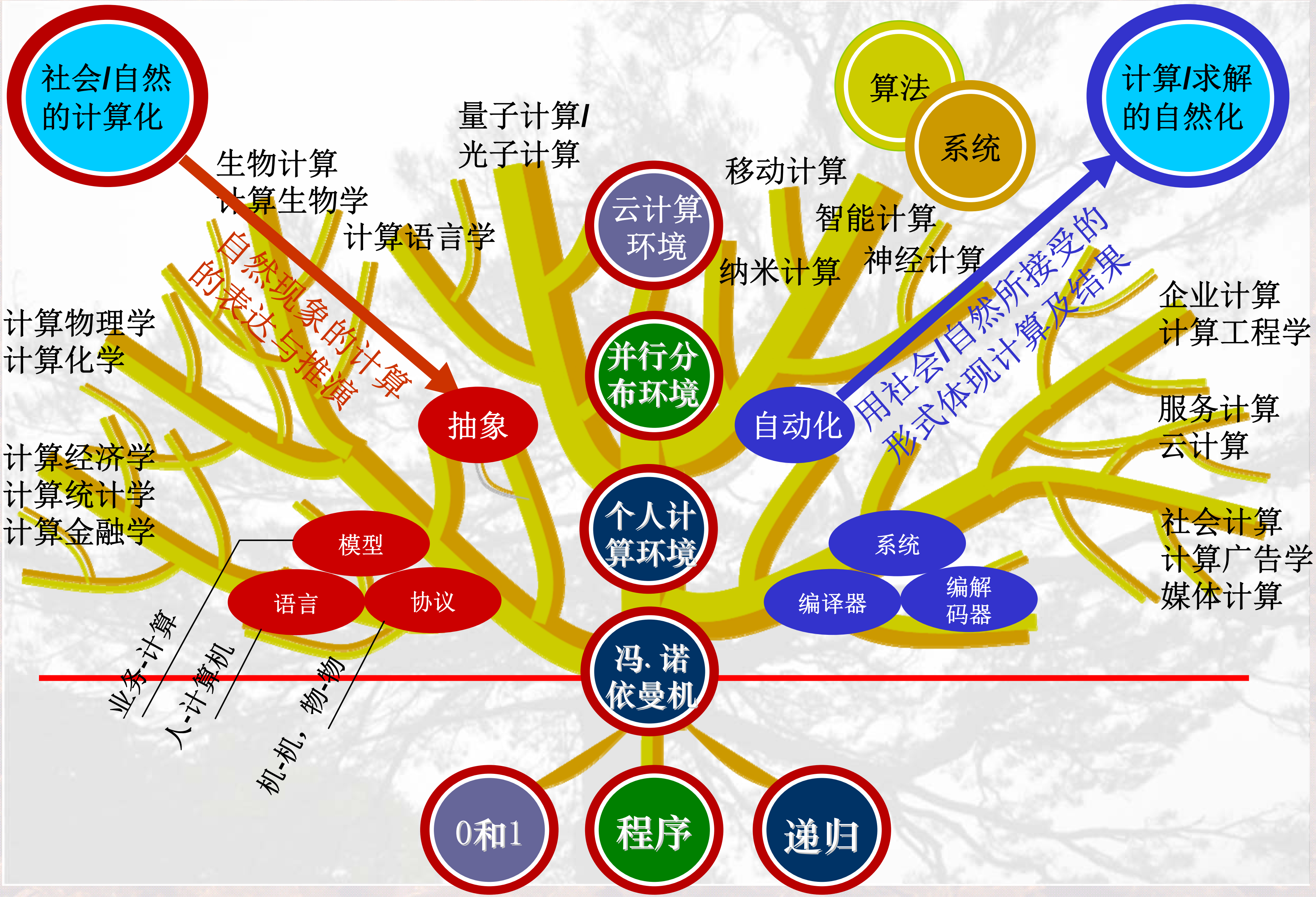
计算之树的第四个维度—计算与社会/自然环境的融合思维

大学计算思维教育空间—计算之树?
(4) 计算与社会/自然如何融合的?



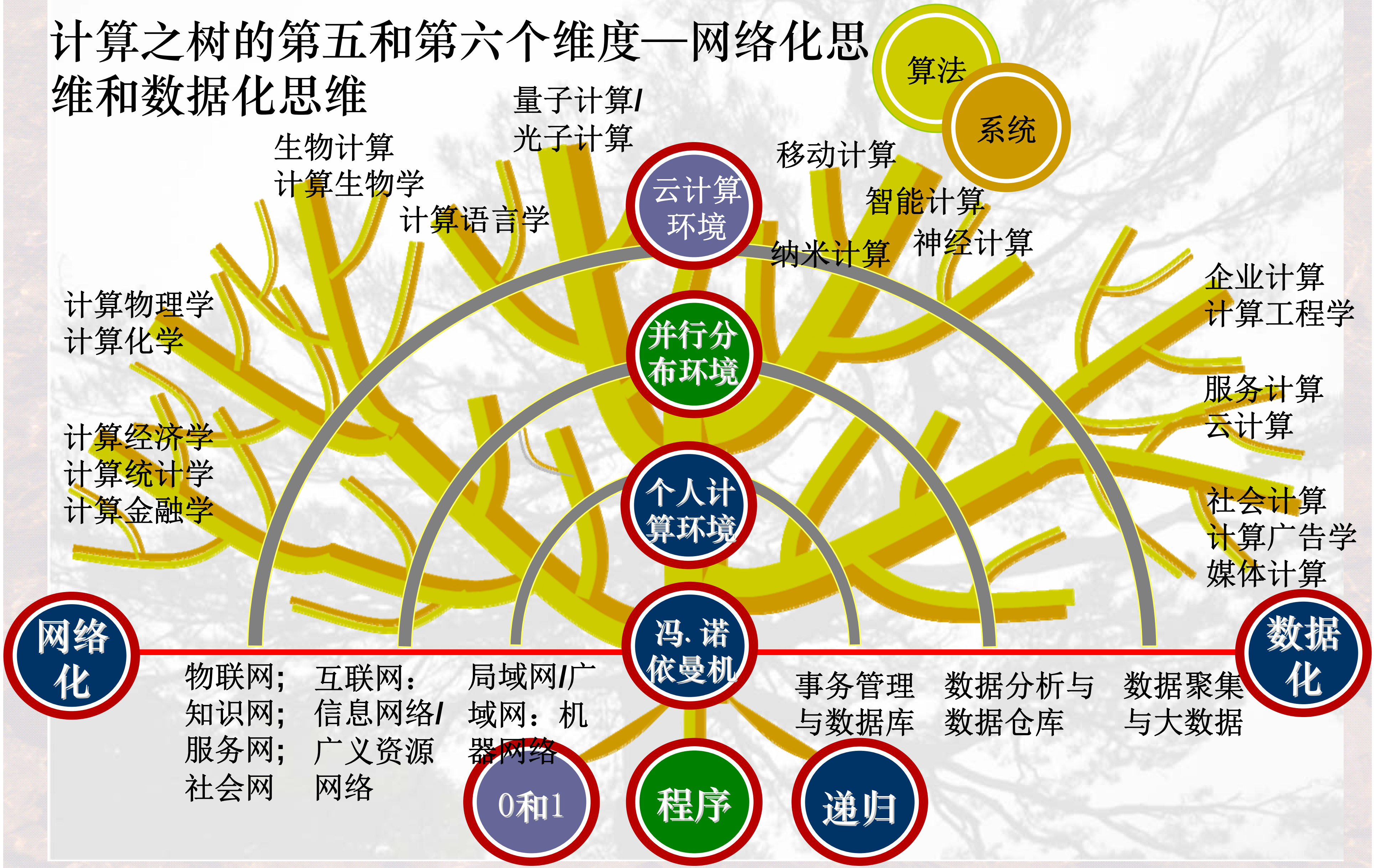
大学计算思维教育空间—计算之树?
(4) 计算与社会/自然如何融合的?

战德臣 教授



ICES 战德臣 教授

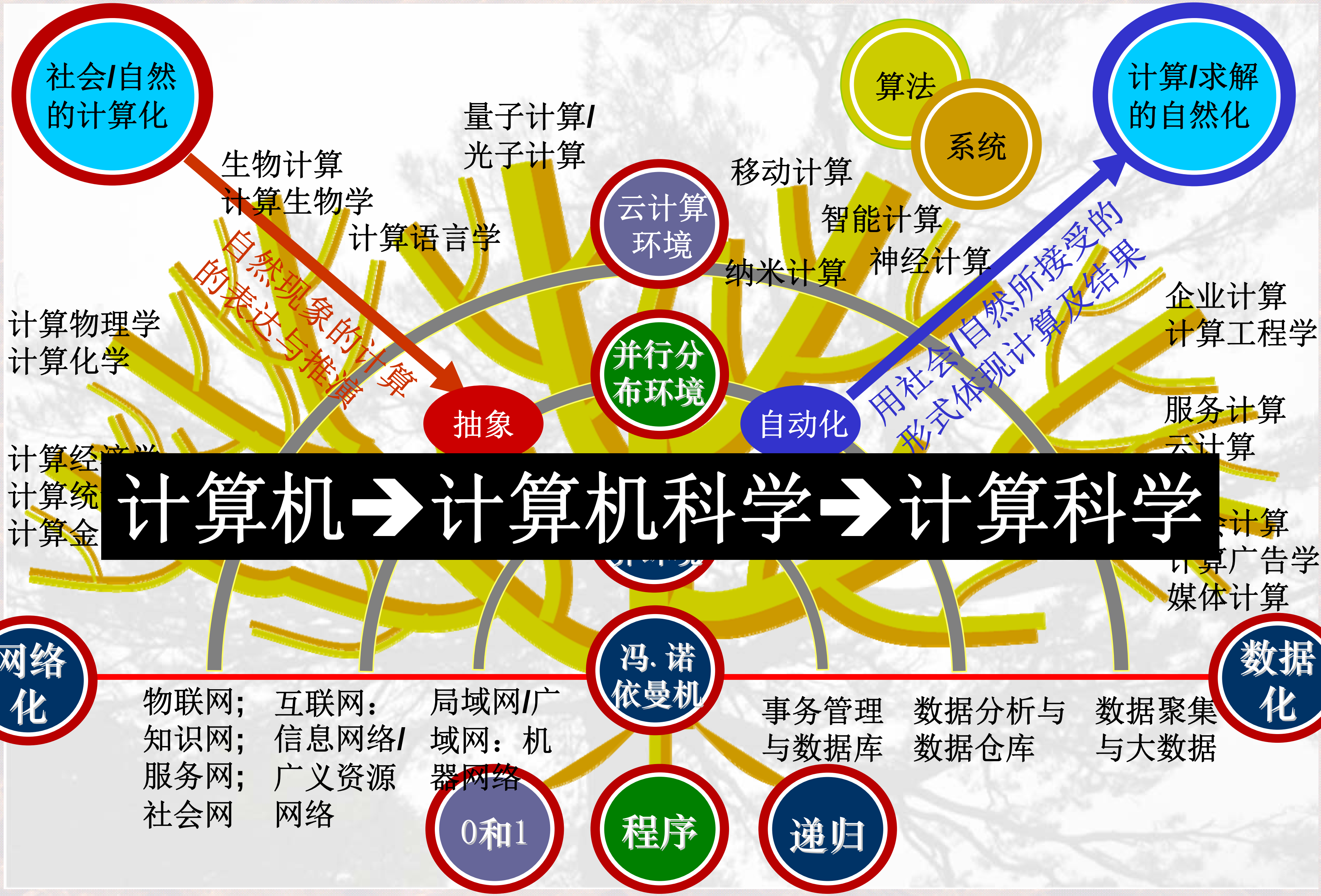
计算之树的第五和第六个维度—网络化思维和数据化思维



大学计算思维教育空间—计算之树?

(5) 为什么网络化思维、数据化思维很重要?

战德臣 教授



◆计算思维的学习方法

（1） “知识/术语”随着“思维”的学习而展开，“思维”随着“知识”的贯通而形成，“能力”随着“思维”的理解而提高。

（2）从问题分析着手，强化如何进行抽象，如何将现实问题抽象为一个数学问题或者一个形式化问题，提高问题表述及问题求解的严谨性。

（3）通过图示化方法来展现复杂的思维可以一目了然；通过规模较小的问题求解示例来理解复杂问题的求解方法；通过从社会/自然等人们身边的问题理解到计算科学家是如何进行问题求解。

（4）追求“问题”及问题的讨论，通过逐步地提出问题，使自己从一个较浅的理解层次逐步过渡到较深入的理解层次，通过不同视角和递阶的讨论，使自己理解和确定前行的方向。

（5）宽度与深度相结合，从宽度学习开始，深度学习结束，既能够使自己理解相关的思维与知识，还能够有助于建立起较为科学的研究习惯与研究方法。

（6）思维蕴含在案例中，案例蕴含着思维。

阅读书籍、阅读文献、网上搜索、梳理思路(记笔记)