

第1章 绪论

讲授：任胜兵
中南大学 计算机学院



中南大学

1.5 软件开发方法



中南大學
CENTRAL SOUTH UNIVERSITY

内容提要

开发方法定义

结构化方法

面向对象方法

面向方面方法

形式化方法

方法结合



开发方法有哪些要素？

软件开发方法就是使用定义好的技术（集）及表示符号来组织软件生产过程的方法。

三要素

开发步骤（包括每步相应的技术和符号）



如何做？

软件文档格式



结果是什么？

开发方案评价标准



结果怎么样？

好的开发方法是经过实践检验的方法。

如何理解软件开发方法？

软件开发的核心输出：程序

程序是什么？

程序=数据结构+算法 (Niklaus Wirth)

如何构建程序？

- 结构化方法：从功能（算法）角度看程序 ==> 以功能为中心。
- 面向对象方法：从数据结构（数据）角度看程序 ==> 以数据为中心。
- 面向方面方法：解决面向对象方法的横切关注点问题。
- 形式化方法：用数学的方法分析并构建程序

好的开发方法是经过实践检验的方法



结构化方法如何解决软件复杂性？

结构化方法认为系统是由一些功能的相互联系、相互作用而形成。

指导思想

自顶向下、逐步求精、单入口和单出口，利用抽象和功能分解来处理软件复杂性问题。

方法系列

结构化程序设计 => 结构化设计 => 结构化分析。

方法实现

面向数据流图的方法、IDEF0方法、Jackson方法、LCP方法。

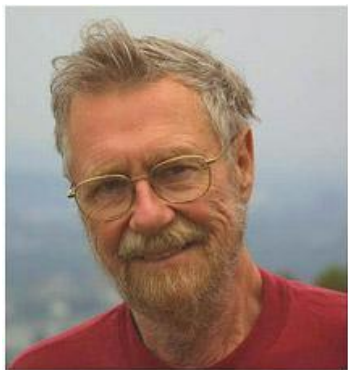
方法特点

- 方法简单实用、技术成熟、应用广泛。
- 不适应规模大的项目及特别复杂的项目。
- 难于解决软件重用（复用）问题。
- 难于适应需求变化的问题。
- 软件维护依然比较复杂。

结构化方法强调以功能为中心。

如何理解结构化方法： 结构化方法起源

Go To Letter



1968年Dijkstra给“ACM通讯”写了一篇短文Go To Statement Considered Harmful, 该文后改成信件形式刊登, 以便早日发表。

Go To语句应从一切高级语言中去掉; 只用三种基本控制结构就可以写各种程序。

Dijkstra被西方学术界称为“结构程序设计之父”。

- 提出“goto有害论”。
- 提出信号量和PV原语。
- 解决了“哲学家聚餐”问题。
- Dijkstra最短路径算法和银行家算法的创造者。
- 第一个Algol 60编译器的设计者和实现者。
- THE操作系统的设计者和开发者。

结构化方法构成的系统是层次。

如何理解结构化方法： 结构化方法发展史

- **逐步设计 (Stepwise design)：** 由尼克劳斯·维尔特 (Niklaus Wirth) 在1971年提出,包括：程序=数据结构+算法。
- 1972年出现的Nassi-Shneiderman图。
- 1974年提出的Warnier/Orr图 - “Logical Construction of Programs”。
- 1974年提出的HIPO - IBM的层次化输入-处理-输出图。
- **Jackson(杰克逊)结构化程序设计：** 由迈克尔·安东尼·杰克逊在1975年提出。
- **结构化设计：** 由赖瑞·康斯坦丁、爱德华·尤登及韦恩·史帝文斯在1975年提出。
- **结构化分析：** 约在1978年由汤姆·狄马克及尤登等人提出。
- **结构化分析及设计技术 (SADT)：** 由道格拉斯·T·罗斯发展。
- 尤登结构化方法：由爱德华·尤登提出。
- 结构化分析及系统规格 (Structured Analysis and System Specification)：由汤姆·狄马克在1979年提出。
- 结构化系统分析及设计方法 (SSADM)：最早是由英国商务办公室在1983年提出。
- 以结构化分析及设计技术为基础的IDEF0，由道格拉斯·T·罗斯在1985年提出。
- Hatley-Pirbhai模型：在 Derek J. Hatley及Imtiaz A. Pirbhai于1988年所著的《Strategies for Real-Time System Specification》中定义。
- 信息工程：约在1990年代由克莱夫·芬克尔斯坦提出，后来因詹姆斯·马丁的推广而广为人知。

结构化方法强调以功能为中心。

如何理解结构化方法： 结构化程序理论

- 1966年，Corrado Böhm 和 Guiseppe Jacopini 在Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules 中证明结构定理。
 - 任何逻辑问题均可仅用顺序、选择和循环结构来解决。又称Böhm-Jacopini理论，发表在《Communications of the ACM》上。
- 1965年，唐稚松院士在《数学学报》发表了一篇关于计算机指令系统性质的论文，提出转移指令可用循环替代，它比Böhm-Jacopini定理早一年。

结构化程序理论表明结构化方法的成熟

在80年代起开始广泛使用。当时采用结构程序设计方法的两个最著名项目是：
纽约时报信息库管理系统，含8.3万行源代码，只花了1年；
美国宇航局空间实验室操作的模拟系统，含40万行源代码，只用两年时间就全部完成。

好的开发方法是经过实践检验的方法



面向对象方法如何解决复杂性？

面向对象方法认为系统是由一些对象的相互联系、相互作用而形成。

指导思想

尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界的方法与过程：对象+类+继承+消息通信

方法系列

面向对象程序设计 => 面向对象设计 => 面向对象分析。

方法实现

Coad/Yourdon方法、Booch方法、OMT方法、OOSE方法、IDEF4方法、CRC方法等。

方法特点

- 对象与功能相比，对象更易理解；
- 由对象建立的系统往往更稳定；
- 面向对象方法易学习；
- 面向对象方法易处理需求模糊或变化的情况；
- 由对象建立的系统便于重用；
- 类的封装，使系统易于修改维护。

面向对象方法以对象（数据）为中心。

面向对象方法的不足：Why OO Sucks --- Erlang语言的发明者Joe Armstrong

- 类作为复用单元，有时显得太小；
- 继承会增加类间的耦合性；
- 面向对象方法理论没有结构化方法成熟；
- 面向对象方法比结构化方法复杂。

面向对象方法形成的系统是网状结构

如何理解面向对象方法：面向对象方法起源。

第一个面向对象语言

60年代中后期，Simula语言在ALGOL60基础上研制开发，它将ALGOL的块结构概念向前发展一步，提出了对象的概念，并使用了类，也支持类继承。

(Simula67作者是Ole-Johan Dahl 和 Kristen Nygaard，2001年获图灵奖)

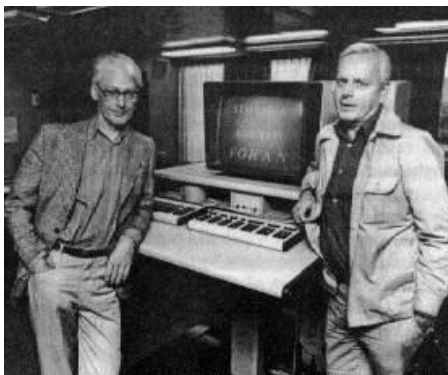
纯面向对象语言

70年代，Smalltalk语言诞生，它取Simula的类为核心概念，并借鉴Lisp语言。由Xerox公司经过持续不断的研究和改进之后，于1980年商品化，它在系统设计中强调对象概念的统一，引入对象、对象类、方法、实例等概念和术语，采用动态联编和单继承机制。

(Smalltalk的作者Alan Kay，2003年获图灵奖)

面向对象源出于Simula67，真正的OOP由Smalltalk奠基。Smalltalk现在被认为是最纯的OOPL。

Dahl and Nygaard at the time of Simula's development



Alan Kay

如果没有继承，就是基于对象的语言。

如何理解面向对象方法：问题空间和软件空间一致

- 强调运用**人类日常的逻辑思维**方法与原则（如抽象、分类、继承、聚合、封装、关联等）。这使得软件开发者能更有效地**分析和解决问题**，并以其他人**容易理解**的方式把分析结果和设计结果等表达出来。
- 无论系统的构成成分，还是它们之间的关系体现的系统结构，都可**直接地映射问题域**。**问题域的成分相对稳定**，这样的映射一方面可以使系统的**软件结构**与所要解决的**问题结构**保持一致，另一方面又可使系统的软件结构相对稳定。



什么是横切关注点?

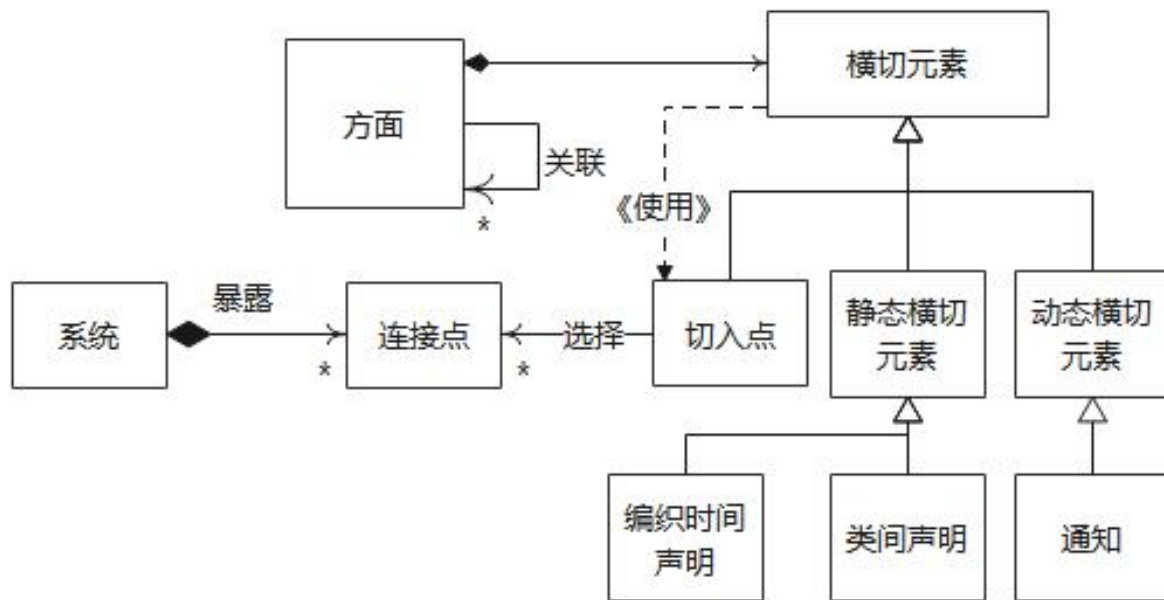
横切关注点：散布在功能模块或对象类中的横切行为。

- 结构化方法以功能为中心，而当数据发生变化时，需要修改许多功能模块。
- 面向对象方法以对象（对象是数据以及与数据有关的操作）为中心，其核心是数据，当需要改变某一功能时，可能需要修改许多对象。
- 面向方面方法利用关注点分离来处理软件复杂性。关注点分离意味着在软件中，每个成分（类、方法、过程等）做且只做一件事。

横切关注点：散布在功能模块或对象类中的横切行为。

- 结构化方法以功能为中心，而当**数据发生变化**时，需要**修改许多功能模块**。
- 面向对象方法以对象（对象是数据以及与数据有关的操作）为中心，其核心是数据，当需要**改变某一功能**时，可能需要**修改许多对象**。
- 面向方面方法利用**关注点分离**来处理软件**复杂性**。关注点分离意味着在软件中，每个成分（类、方法、过程等）**做且只做一件事**（一个关注点）。
- **关注点**可以定义为一个或一组项目信息持有者感兴趣的或对他们**意义重大的事情**。例如，在不同的系统功能中记录操作日志。
- 所谓**方面**，是指涉及到若干功能模块或对象类的系统功能（**横切关注点**）。

面向方面程序设计的一般模型。

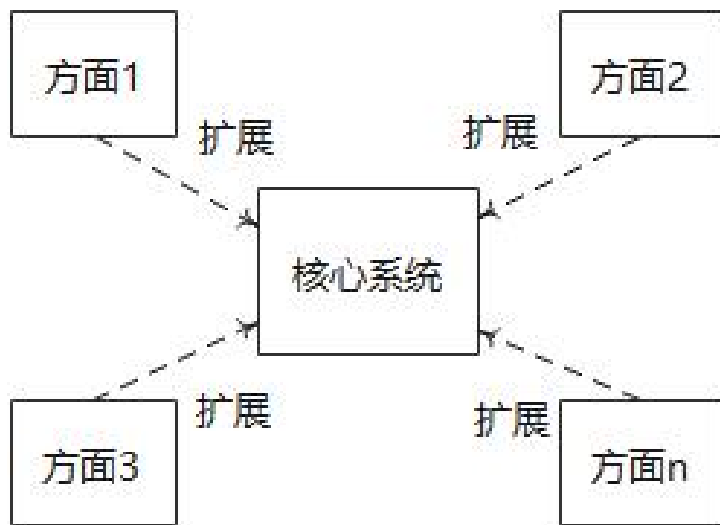


- 切入点是连接点 (Join Point) 的集合，而**连接点**是系统执行过程中明确定义的一个事件，是**横切行为的插入点**。
- **类间声明和通知**表示**横切行为**，类间声明用于在已有类型中**添加新的方法、属性、接口等**，而通知表明在切入点处**增加的行为**。

伊恩·萨默维尔将关注点分成五类。

- **功能性关注点**：是包含在系统中的与特殊功能相关的关注点。系统的核心关注点是那些关注到系统主要目标的功能性关注点，例如火车控制系统中的火车制动。
- **服务质量关注点**：是与系统的非功能性行为相关的关注点。包括性能、可靠性和可用性等特征。
- **政策关注点**：是与管理系统使用的总的政策相关的关注点。包括信息安全关注点、业务规则相关的关注点等。
- **系统关注点**：与系统总体属性相关的关注点。例如系统的可维护性和可配置能力。
- **机构关注点**：是与机构目标和优先级相关的关注点。例如对现存软件资产的利用等。

面向方面方法认为，一个系统是由核心关注点构成的核心系统和若干横切关注点构成的若干方面扩展所构成。



- 面向方面方法的技术有：雅各布森（Jacobson）提出的AOSD（Aspect-Oriented Software Development with Use-Case）方法、克拉克（Clark）提出的Theme方法等。



形式化方法如何去除缺陷？

形式化方法认为系统可以通过数学分析和推理形成。

指导思想

系统可以通过严格的、规范化的数学理论经分析、推理和转换得到。

方法系列

形式化规约方法、形式化分析/推理方法、转换方法。

方法实现

VDM(Vienna Development Method,维也纳)方法、RASIE方法、Event-B方法。

方法特点

- 形式模型完整、一致和无二义性；
- 支持形式推理，便于软件验证；
- 便于软件自动生成；
- 一般人不易接受，需要培训,增加成本；
- 灵活性差；
- 难以与工程化的软件开发过程平滑地结合；
- 支持工具少。

2019年软件学报“形式化方法概貌”

形式化方法就是数学方法。

最早可以追溯到20世纪50年代的编译技术的研究，由BNF自动生成程序分析工具。

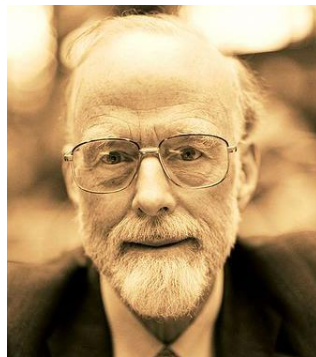
20世纪60年代后期，出现了一系列形式化方法

Petri网方法



Petri

VDM/VDM++ CSP/CCS



Hoare



Milner

Z/B/Event-B



Abraail

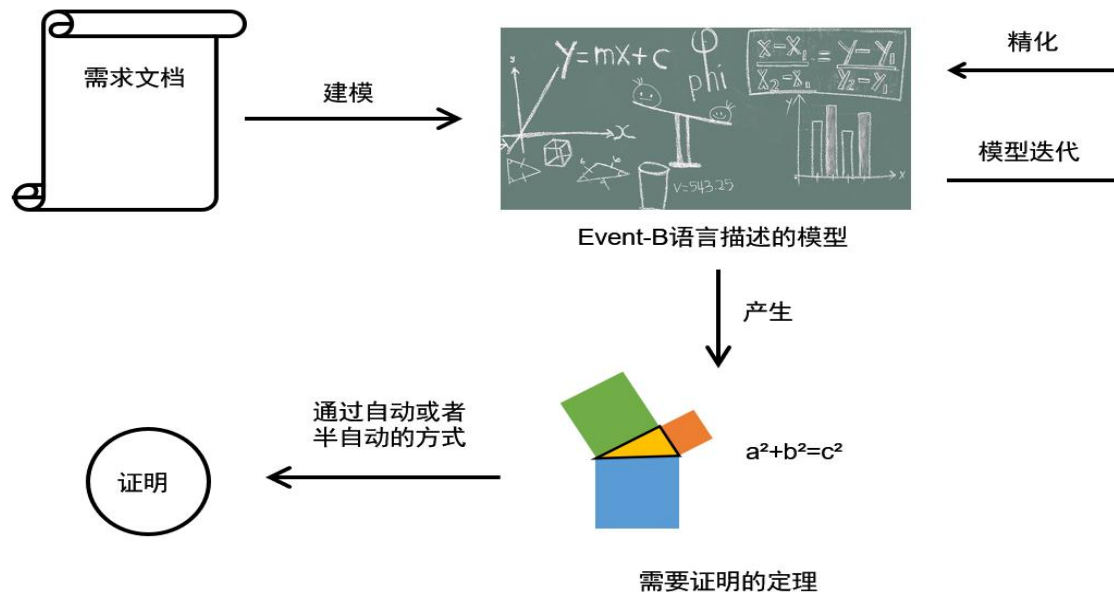
2019年软件学报“形式化方法概貌”

形式化方法可以充分消除缺陷

形式化方法：描述做什么，不描述如何做。

形式化方法的一个重要研究内容是形式规约：是对程序做什么的精确数学描述。

形式化方法的另一重要研究内容是形式验证：正确性证明、模型检测、模拟。





如何利用各种方法的优势？

利用各种方法的长处，从而实现优势互补。

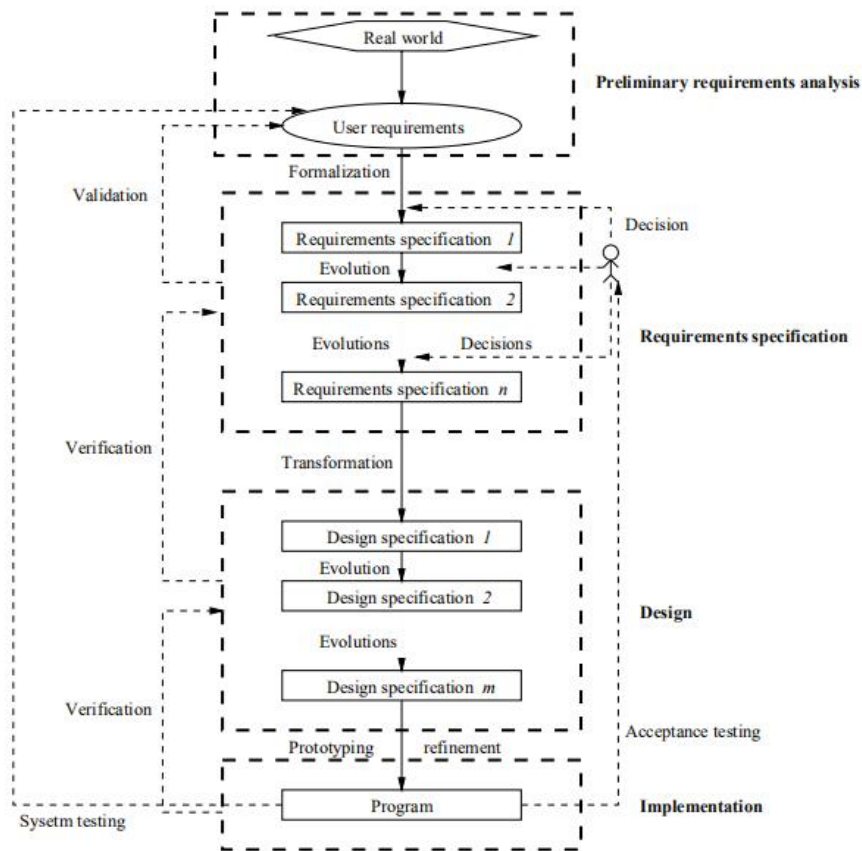
S. Liu (刘少英, 日本法政大学) 等人提出了一门SOFL (Structured Object-based Formal Language) 语言和一种集成了结构化方法、面向对象方法和形式化方法于一体的SOFL开发方法学。

SOFL uses **structured methods** for **requirements analysis** and specification and an **object-based approach** for **design** and **implementation**。

During both the structured and object-based development of the system, **formal methods** can be applied to **provide high quality specifications and verifications** of various levels of the system.。

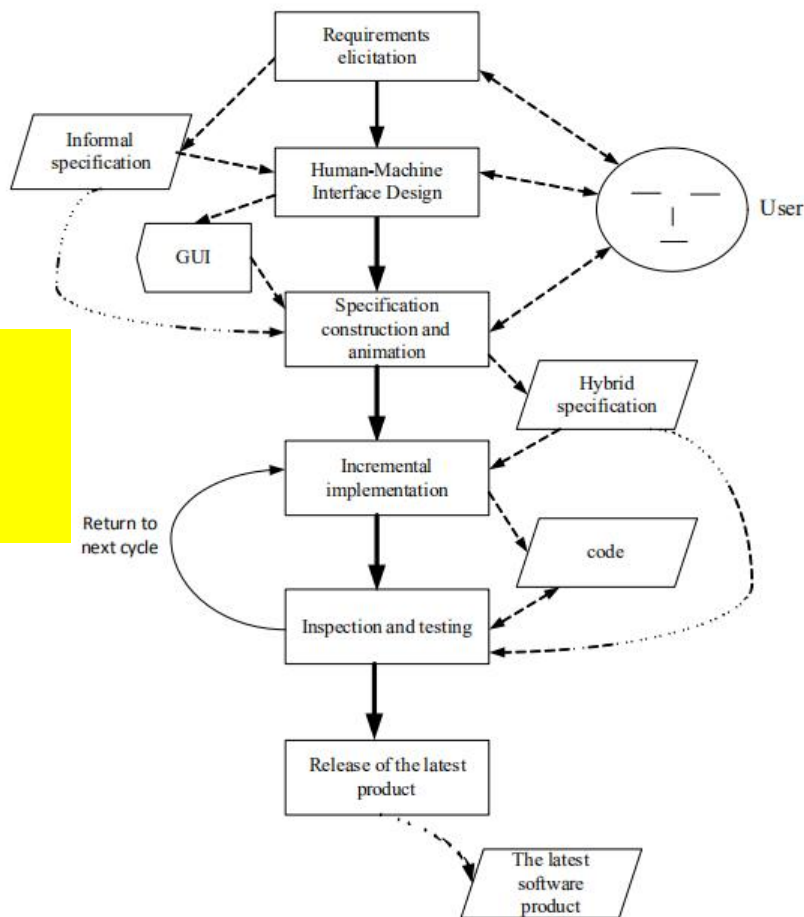
在软件开发过程中也需要结合各种方法。

SOFL开发过程



在软件开发过程中也需要结合各种方法。

Agile-SOFL开发过程



Agile Formal Engineering
Method for Software
Productivity and Reliability

在软件开发过程中也需要结合各种方法。



中南大學
CENTRAL SOUTH UNIVERSITY

**感谢各位聆听！
祝大家学习愉快！**