中国科学院50年
科技成果汇萃之一

XYZ系统的目的、意义、作用与应用*

唐稚松

(中国科学院软件研究所 北京 100080)

摘要 XYZ系统是一个基于线性时序逻辑的软件工程系统,由中国科学院软件研究所经过15年的时间设计并实现.它被用于解决某些高技术工程领域的问题.文章介绍了这个系统的目的、意义、作用和应用.

关键词 时序逻辑语言,状态转换,动态语义,静态语义,规范语言,冯诺曼模型,逐步求精,构件,软件体系结构,一致性验证.

中图法分类号 TP311

1 时序逻辑语言XYZ/E研究的目的与意义

20世纪70年代以来,由于半导体线路生产自动化水平提高很快,计算机硬件造价迅速降低,而软件生产仍处于手工编制状态,其可靠性很差、生产率低下,形成瓶颈,严重影响了计算机的应用与发展,并且进一步地也影响其他新技术的发展.因此,长期以来各工业先进国无不以提高计算机软件生产率作为国家关键技术的重中之重.

针对这个问题,美国工业界与西欧学术界各自提出不同的解决方法.美国工业界的方案着眼于技术,认为提高软件生产率应加强支撑软件开发的工具以提高其自动化水平,以及加强面向对象程序设计技术以提高程序模块的可重用性;而西欧学术界则认为软件生产率低主要是由于程序可靠性差所致,其原因是由于命令式程序语言中包含求解过程的细节太多,以至难读懂、难修改,因此,应设计一种直接表示程序含义的形式化规范语言书写程序,然后再自动转换成有效的执行程序.故其关键是关于形式化程序语义理论与规范语言的研究.他们沿着各自的道路,越走越远,形成理论与技术实际严重脱离的现象,最后导致提高软件生产率的问题长期难以解决.我则认为,这两种方案对提高软件生产率都有其重要意义,但不应彼此分离,而应紧密结合起来.而妨碍它们相互结合的深刻原因在于计算的模型.本来计算机是为解题而研制的,其模型是自动机(即图灵机或有穷自动机),其最基本的特征是状态转换.这就是通常所谓的冯诺曼体系的本质.在20世纪70年代以前,不论机器体系、程序语言(命令式高级语言的赋值、循环等也是经过掩饰的表示状态转换),还是当时形式化语法理论也都是围绕着这一模型建立的,故相互间紧密结合,推进了生产率的发展.自从规范语言及形式化语义理论提出后,情况发生了变化.其原因是长期找不到一种以逻辑或代数等理论为基础的语义形式化方法以表示出状态转换机制.但另一方面,计算机硬件体系结构又必须以状态转换方式表示.当时以英国学术界为代表的理论家主张,不但理论及规范语言(当时以指称语义及代数语义为代表)应建立在函数式模型之上,而且计算机硬件体系结构及程序语言的基础也应改为函数式模型.因此,一时间形成非冯诺曼模型研究的高潮,我国的计算机界也受此影响.而我当时对这一潮流却持怀疑的观点.我认为,只要作为物质基础的线路元件是脉冲式离散型的,不论硬件体系还是程序语言,归根到底不可能脱离以状态转换为特征的冯诺曼模型.可是,如何以形式化方法表示状态转换机制的

---

* 作者唐稚松,1925年生,研究员,博士生导师,中国科学院院士,主要研究领域为计算机科学,软件工程.
本文通讯联系人:唐稚松,北京 100080,中国科学院软件研究所
本文1999-02-03收到原稿,1999-03-03收到修改稿

语义则仍是一个未解决的难题,时常挂在我的心上.

1979～1981年,我应邀到斯坦福大学访问.这时,以色列著名理论家,后来在1997年荣获图灵奖的A. Pnueli教授提出了时序逻辑理论,以Z. Manna教授等为代表的斯坦福专家们正热烈地开展应用此理论表示并发通信进程的活性与安全性的研究.我在研究了他们的工作之后感到非常有意义,但是时序逻辑理论还有另一重要的特性被他们忽略了,即用这种理论可以自然地表示状态转换机制,而且这是动态语义.我首次大胆提出时序逻辑语言XYZ/E(详见文献[1])的概念,并认为它可以作为软件开发过程的统一基础.当时,计算机科学理论家虽然不能指出我的主张有何不妥之处,但对此都普遍感到怀疑.我明知我的这一途径将使自己摆在差不多全世界计算机科学理论主流的对立面,但我却坚信这是使计算机研究理论与技术结合的唯一正确方向,很有意义.这样做,工作量极大,必将耗费我余生的全部精力,并且得不到理论家的支持,但我认为,不必为此计较个人一时得失,坚持下去必将成功.

到了80年代后期,RISC技术与工作站流行,所谓函数式计算机终被淘汰.而函数式语言除LISP及SML在不讲究效率的应用领域,如人工智能及程序验证系统实现等方面,仍保持其生命力外,多数这类语言已无人问津;但以函数式模型为基础的语义理论及规范语言(特别在西欧学术界)仍保持活跃和繁荣.等到90年代前后,即令在这一领域,也已开始有些著名理论家,如Braunschweig的H. D. Ehrich教授、SRI的J. Meseque教授、慕尼黑的M. Wirsig教授,陆续在其范畴论或代数语义的框架中引入状态转换机制,并承认受XYZ/E的影响[2],或者承认因E. Engeler教授或H. D. Ehrich教授的推荐而了解了我们的工作.虽然软件工程界的主要科学家,如马里兰大学的叶祖尧教授、CMU(Carnegie-Mellon University)的N. Habermann教授、Trondheim的Solvberg教授等,早已对XYZ/E作出了很高的评价*,而在理论界直接明确肯定XYZ系统的重要意义则直到1988年才开始.英国理论家H. R. Barringer与D. Gabbay在一篇总结性报告中指出,"在将时序逻辑运用与发展于软件工程方面的重要一步是找到可执行的时序逻辑",并承认XYZ/E是这方面最早的"先驱"(LNCS NO. 335).而更重要的变化则是A. Pnueli教授对待XYZ/E态度的改变.我们虽然于80年初即在斯坦福相识,但他对XYZ/E一直不表态.直到1994年他第一次访问我国,在仔细询问下参观XYZ系统的演示达数小时之后,他对我说:"过去我对XYZ系统一直很怀疑,认为你野心太大,不可能成功.这次参观后,我发现你已经成功了".在他与他的合作伙伴Z. Manna教授的倡议下,1995年在北京召开了《逻辑与软件工程》国际研讨会,以庆祝我70寿辰.在Manna和Pnueli合著的论文正文之前,附有如下一段祝词:"唐稚松教授将时序逻辑概念用得超乎任何人的想像,在他之前没有人认为是可能的"[1].在Pnueli为这次研讨会论文集所写的序言中,他又进一步解释了这一思想.他是这样说的:"我仍记得在最初听到唐教授提出要将时序逻辑作为整个软件开发过程的基础时所感到的惊讶(amazement).随着时间的流逝,这个系统被中国科学院软件研究所一个致力于此的小组不断扩充与实现,这种惊讶也逐渐转变成为钦羡(admiration)".此外,在序中他还强调了"由唐稚松教授所构想并发展的XYZ系统作为先驱所开拓的这一方向"对于软件工程的重要意义[2].接着,在1997年他从以色列启程去美国洛杉矶领取图灵奖的前夕,给我来了一封电子邮件.在谈到他荣获图灵奖时,他说:"由于我一贯认为你是时序逻辑的最强有力与最热衷的推动者之一,我完全相信你应该分享这一使时序逻辑成为具有深远影响的概念所授予的荣誉中的一个有意义的部分".

应该指出,随着90年代计算机领域技术情况的发展,我从80年代初以来所持的观点已被证实.以上所述国际上一部分第一流的理论家对待XYZ系统的态度虽已发生彻底变化,但这些都只限于到过我们研究所参观了XYZ系统演示的少数人,国外广大计算机科学技术界对XYZ系统具体内情仍很不了解.许多人认为,这主要是由于一本全面详细介绍XYZ系统的英文书尚未能写出并出版所致,实因我太忙而力不从心.

事实上,时序逻辑语言XYZ/E在语义理论方面的重要意义还远不止于此.还有两个方面至今很少被人所认识:

---

* 美国软件工程先驱、ISSI的叶祖尧教授对XYZ作的书面评语云:"我心中绝对相信唐的XYZ语言将是一次重大突破的基础.……他的工作非常有创造性,而且有使软件生产率取得重大提高的实际应用前景".Solvberg教授也对XYZ作了相近的高度评价.

(a) 这一语言能以统一的程序框架既表示程序的动态语义(即状态转换机制如赋值、转移等),又表示出其静态语义(即前置断言与后续断言规范).由于能做到这一点,它才能表示逐步求精过程,使程序开发过程能一边分步骤进行,一边检验其语义一致性.这样即可以使一个复杂过程分解成许多步,从而使一段复杂的程序变得既便于理解又便于维护,特别是可以避免将程序的规范与其执行程序分割成彼此独立的两大块去进行"马车套在马的前面"式的验证.现今除 XYZ/E 外,其他形式规范都未能将动态语义与静态语义混合写在一段程序之中,故其程序维护与验证非常困难.

(b) 对于通信并发进程而言,所有流行的实际命令式语言仍是建立在原来以状态转换机制表示的、由 Hoare 提出的 CSP 模型之上的.但 Hoare 在 80 年代中发现,他原来所希望的将并行语句的语义用其中包含的各进程的语义的合取(也就是结构化组合)来表示的设想在通信的情况下可能不成立.因此,他在 80 年代介绍 CSP 的书中,将其语义模型由状态转换(即冯诺曼模型)改成了依据 CCS 所发展的进程代数(它是以函数式语义为模型的).这样一来,实际实现的 CSP 与其语义理论中的 CSP 其模型事实上并不一致.这样的漏洞在实际应用中显然就可能导致语义方面的问题,我认为,近年出现的下述两件事是与此相关的.一件是据传闻在上次海湾战争开始时,伊拉克从国际市场上购买的软件系统突然瘫痪;另一件是最近在马来西亚召开经合会首脑会议上作为主席的马哈蒂尔的全部发言突然从所有的录音带上消失,但事前事后检查所有线路都是正常的.对于这样一个关于通信进程语义可组合性的问题,我们已找到了一种解决的办法,能使它在 XYZ/E 的框架内成立.这样一件事的重要实际意义在软件技术界似乎尚未引起足够的注意,人们还以为这只是理论家的纸上谈兵而已.除此之外,多年来我们对时序逻辑语言中各种机制的语义问题都作了详细的理论推敲.一方面保证其功能与常见语言一致,另一方面又保证其理论上的可靠性,而且表示语义的语言与常见程序语言的风格完全一致.在已有的形式语义与规范语言中能做到如此全面考虑的,据我们所知并不多见,它们大多只能表示部分机制的功能,而且甚至已改变该机制的原来意义.以上是时序逻辑语言 XYZ/E 的意义所在.

2　XYZ 系统中软件工程方法与工具的作用与应用

XYZ 系统为时序逻辑语言与软件工程方法与工具相结合而成.后一方面包括以下 3 类:

(I) 基于模块的可视化的图形设计与分析工具.这方面所讨论的程序模块包括基于程序流图的模块(包括过程与进程)、并发通信模块和面向对象模块.每种模块都有与之对应的可视化的图形设计工具,并且每种图形模块设计完成后均可由相应工具自动生成由 XYZ/E 表示其动态语义的程序.这类工具曾是 80 年代国际软件工程研究的潮流.但从事后来看效果并不理想,只能作为一种辅助工具.其理由是:(1) 对于多重模块组成的程序,其集成过程未能自动化,由用户负责既不方便又难以保证其语义正确;(2) 与形式规范脱离,一致性验证不易进行;(3) 未能体现程序设计的逐步求精过程.

(II) 面向形式规范的逐步求精与语义一致性检验过程的支撑系统.这一方法和工具恰与(I)相反,它着重强调了形式规范、逐步求精过程及语义一致性.我们已应用这一方法在表示几个国际著名问题的解题过程中取得了较为成功的效果.一个是应用时序逻辑语言 XYZ/E 表示了由 Abrial 提出的蒸汽锅炉实时混成控制系统的开发全过程,从建立物理模型开始,表示其安全性、活性组成的形式规范、逐步求精过程、每一步语义一致性的验证以及最后用可视化图形表示的有效的实现,全过程均在 XYZ/E 的框架内表示并进行光滑的过渡.能做到如此程度,在关于这个著名问题的国际文献中过去还未见过.此外,我们也应用 XYZ/E 语言及其 Hoare 逻辑验证系统,表示并论证了 L. Lamport 与 M. Broy 曾在 TLA 框架内论证的 RPC-Memory 问题,效果也颇令人满意.由于有了这些试验作基础,我们现在正开始与有关单位合作,应用这一方法与工具表示并验证我国宇航方面一个大型实时控制系统的开发全过程;类似地,我们也正准备与有关专家合作,研制一个具有一定复杂性的 CIMS 系统并论证其正确性.此外,我们还将这一方法应用于软件技术其他的新领域,比如动画片与多媒体.数年来,我们与北京邮电大学的专家合作,已应用 XYZ/E 语言实现了这类新技术的典型例子.现在,进一步的问题是如何将形式规范、逐步求精方法及语义一致性检验这些理论方法也引入到这些新技术领域中(如果成功,影响将很大).过去,在这方面世界上几乎无人做过任何有意义的尝试,难度极大.最近,有一位研究生已令人满意地表示出这类新技术程序的轨迹语义(已引起国内有关专家的注意).另一位研究生也已找到一种表示这类程序的轨迹语义的

形式规范、逐步求精过程及语义一致性检验的方法.看来在这方面取得突破已为期不远.但这一方法仍有一个重要不足之处,即未能与软件模块结合.

(III) 基于组件并面向软件体系结构的逐步过渡并检验其语义一致性过程的方法与工具.关于软件组件(或构件)及软件体系结构的研究是两个新兴的软件工程研究方向.有人(Perry-Wolf)称"90年代是研究软件体系结构的时代".我认为,这类研究如果脱离形式语义基础将是不可靠的.而XYZ/E能以统一框架同时表示静态语义与动态语义,故可在这方面起到关键作用,而且由此形成的方法与工具正好可将上述(I)中的软件模块化特征与(II)中的形式规范与逐步求精特征结合起来.为此,我们提出一种以可视化图形表示的软件体系描述语言XYZ/ADL,其核心为组件.每一组件由两方面构成:其外部界面即静态形式规范,其内部结构即体系.体系由其子组件界面与联结关系(connections)所组成.每一步设计即将其组件的外部界面(规范)替换成其相应的体系结构,这就形成一步过渡.体系内部又包含子组件,还需将其外部界面的规范过渡到相应的内部体系结构,如此构成逐步过渡过程即逐步求精过程.每次体系结构画出后,系统即可自动生成相应的表示其语义的XYZ/E程序(由联结关系的动态语义与子组件规范的静态语义混合而成).用此XYZ/E程序与原组件界面的静态规范相配合即可论证一次过渡的语义一致性.故它包括了(I)、(II)两类方法与工具的两方面的特征.因此,(III)中所谈到的正在进行研制的应用问题(如航大实时系统及CIMS系统等)均将同时应用(III)中方法作为另一方案进行研制,然后将两方案的结果进行比较,以见到两种方法的效果.此外,北京邮电大学计算机系王任艾波教授和由他指导的周莹新博士在《软件学报》(1998年第11期)上发表的论文中也介绍了他们应用XYZ/E表示软件体系结构问题所得出的令人信服的结果,并在此基础上提出一整套软件工程方法论,且试图将之应用于电信领域:电信信息网、智能网、管理网的研制.我认为这是XYZ系统一项非常有意义的应用.此外,我们还即将与西安邮电学院专家合作,开展应用XYZ/E研制软硬件结合的嵌入式系统.

这里应说明一个问题.在XYZ系统中,以逐步求精方法表示的软件开发过程被置于较为关键的地位.这样的过程与一般软件工作者所习惯的实际软件开发过程是什么关系呢?事实上,实际软件开发过程由先后3个阶段构成:即形成需求,由需求得出(形式)规范,由规范导引设计并编制出可在计算机上有效运行的正确程序.其中每一阶段都可能因为认识不够全面或判断有失误或因客观情况的变化而需要补充与修改.一般来讲,这是一个由底向上不断试探与修改的过程,但逐步求精都是一个由顶向下的分解过程,这二者应如何协调配合?事实上,这种现象在数学问题求解时也常出现.众所周知,人们在数学问题求解时,实际上常起作用的是Polya所总结过的那种试探性过程.往往是先构造典型的例子或反例,对其进行分析修改,找出解题规律.在这一过程中总不免判断有误,要求进行老鼠走迷宫式的回溯,再向前再回溯等等.最后找到解决问题的合适途径之后,再将总结出的结果按照逻辑关系将这一过程的各部分整理成定义、公理、引理、定理、推论的形式.事实上,后面这种理性的架构并非实际解题过程的直接反映,而是在经过试探性解题过程,找到了较好的解题途径之后,对其进行总结、分析、整理,找出各组成部分的逻辑依赖关系所得的结果.它反映了解决该问题的合适途径、各部分之间的内在逻辑关系.有了这样的描述,以后遇到同类问题即可进行单刀直入的处理,免走弯路;或者对性质相近的问题只要找出其不同之处即可直接对相应的部分进行适当的修改并论证其语义一致性,而不必一切从头开始进行试探.这后一点对于程序维护尤为重要,因为程序开发事实上是一个不断修改论证的过程,甚至在一个软件系统交付使用多年以后,仍可能因外在条件的改变而要求修改.所以在解题过程中,试探法与经过分析组成合理的逻辑结构这两方面事实上是交替应用的.这个两过程并不矛盾,而是互相补充的.因此,这种由规范导引的逐步求精方法(特别是与记录开发历史的工具相结合),对提高可维护性的作用是非常有意义的.而90年代以来关于软件体系结构的研究,更为表示这种形式规范导引的逐步求精(过渡)方法找到一种合适的技术基础,即构成一种将软件的功能与结构的分别表示这两个方面结合在一起的方法,因此它可以提高软件的可靠性与可维护性.

目前,XYZ系统已开始进入走向应用与走向世界的阶段.一本介绍XYZ系统的中文专著《时序逻辑程序设计与软件工程》的上册(介绍时序逻辑语言)于1999年1月由科学出版社出版.其下册(介绍软件工程)需待2000年才能出版.至于英文专著,因人力所限,看来尚待时日,这不可避免地将对迅速扩大XYZ系统的国际影响起到消极作用.

最后,还应指出,XYZ系统的基础是在哲学思想上从西欧理性主义与北美实用主义的片面性中解放出来,

走中国传统哲学思想与西方逻辑分析方法相结合的道路.日本软件工程学会主席岸田孝一先生对此极为赞许,他在 1995 年 12 月 4 日《朝日新闻》(夕刊)上介绍 XYZ 系统时说:"虽然这个系统(指 XYZ 系统)所采用的基础数学理论来源于西方,但构造此系统的基本思想却来自于孔子的中庸哲学与佛教禅宗的认识论哲学.这也许可以说是东方文明对于新的 21 世纪计算机技术发展的一大贡献吧".

参考文献

1　唐稚松等.时序逻辑程序设计与软件工程(上册).北京:科学出版社,1999
(Tang Zhi-song *et al*. Temporal Logic Programming and Software Engineering (Vol.I). Beijing: Science Press, 1999)
2　Pnueli A, Lin H ed. Logic and Software Engineering. Singapore: World Scientific, 1996

The Goal, Meaning, Effect and Application of the XYZ System

TANG Zhi-song

(*Institute of Software The Chinese Academy of Sciences Beijing* 100080)

Abstract　XYZ system is a software engineering system based on linear time temporal logic designed and implemented by the Institute of Software, The Chinese Academy of Sciences for one and a half decades. It has reached its last stage of development in application to solve some high technical engineering problems. In this paper, the goal, meaning, effect and applications of this system are introduced.

Key words　Temporal logic language, state transition, dynamic semantics, static semantics, specification language, Von Neumann model, stepwise refinement, component, software architecture, consistency checking.