Ansys

SCADE

嵌入式系统 | 细数Ansys SCADE的前世今生

作者：沈轶烨

安全关键领域的系统中，软件的占比正稳步提高。而软件开发的成本既包括设计和编码的成本，也包括验证的成本。传统方式设计的商用软件中每百万行代码中约有100个缺陷，通常缺陷中20%是严重等级的，1%是灾难等级的。相对于商用软件，安全关键系统的软件缺陷可能会少一个数量级，即每百万行代码中约有20个，因此，在安全关键系统的软件中，每百万行代码中平均5个缺陷中就至少有1个是严重的。

由于安全关键系统的全生命周期会长达几十年，再考虑到后期的维护、改造和升级费用，传统方式设计的软件成本几乎不可控制。有什么工具和技术，能在现代安全关键系统的软件规模急剧增长的前提下，既能保证系统的安全可靠，又能适当降低设计的复杂度，还能在开发阶段的前期检测出缺陷，以减少开发成本呢?这就不得不提到Ansys SCADE系列产品。

SCADE诞生于上世纪80年代的法国，从欧洲的航空与核能领域的工程应用起步，经过近40年的发展，现代SCADE已经是融合了Esterel、Lustre、SAO\SAGA、Lucid Synchrone等多个语言和工具的集大成之作。由于SCADE专注于流程规范、标准严苛的安全关键行业，行业的特性使得其多应用于研制具有相当密级的、高难度的重大项目。因此，尽管进入中国市场已有10多年，其知名度却一般。

在逐渐成为航空航天、国防军工、轨道交通、核能重工、汽车电子等安全关键行业里具有广泛的商业应用的同时，SCADE也是产学研相结合的一个典范，SCADE研发团队中脱颖而出的法国科学院院士和图灵奖获得者就是对该产品的一种绝佳肯定。本文将主要介绍SCADE的起源，发展，现状和展望，在后续系列文章中也将对国外多家知名企业的SCADE应用情况做详细介绍，希望以此让大家对SCADE产品和行业应用有更好的认识。

01 诞生阶段

在上世纪80年代早期，Jean-Paul Marmorat和Jean-Paul Rigault，这两位在巴黎高等矿业学校(Ecole des Mines de Paris)研究控制理论和计算机科学的学者，**他们在设计机器人汽车的过程中，受困于传统编程语言不太适合写出精确的控制逻辑，因此专门设计了一种新的语言，而这就是SCADE语言的雏形。**

不久，在巴黎索菲亚科技园(Sophia-Antipolis)，由研究所主任Gerard Berry带领的国立巴黎高科矿业学院和法国国立计算机及自动化研究院(INRIA)所组成的联合团队学者也逐渐加入到这个语言的开发中来，并以索菲亚科技园旁风景秀丽的Esterel山命名该语言（Esterel山也与以电影节闻名于世的法国南部小城戛纳相距不远）。

与此同期，位于格勒诺布尔(Grenoble)的法国国家科学研究院(CNRS)的Paul Caspi和Nicolas Halbwachs两位学者共同发明了Lustre语言，Lustre语言是开源的，而且是一种形式的语言，在学术领域有较多研究。

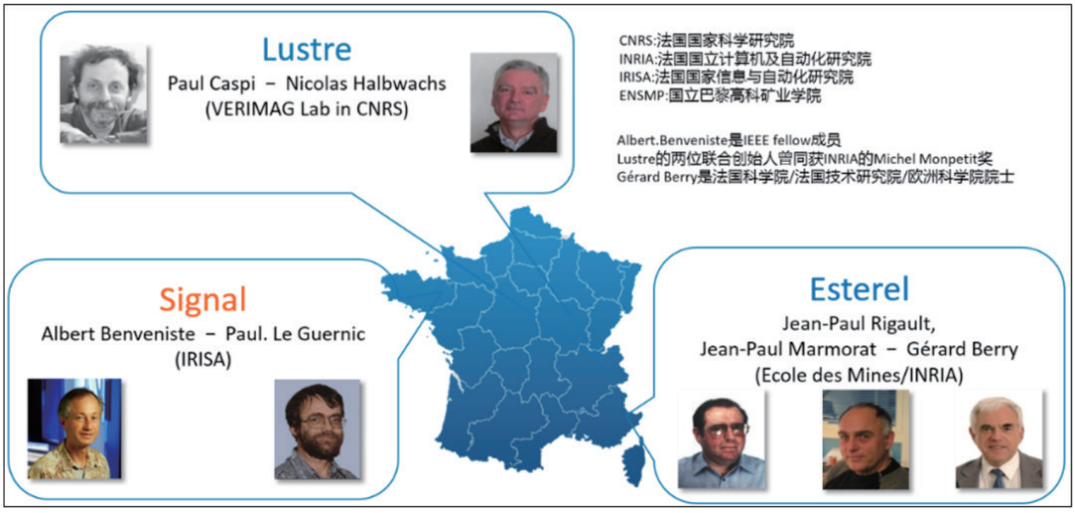
02 早期发展

在上世纪80年代中期，Gerard Berry团队的Laurent Cosserat、Philippe Couronné和Georges Gonthier相继开发出了Esterel版本1/2/3。其中版本2引入了Gerard Berry与贝尔实验室的Ravi Sethi共同设计的算法，能将正则表达式转换为自动机(automata)。版本3开始逐步走向实用化，在Dassault Aviation和Bertin两家公司有了成功实践后，Esterel语言已能在中小规模的项目上应用。

位于法国格勒诺布尔的Merlin-Gerin公司，即后来的施耐德电气公司(Schneider-Electric)，正承担为法国电力公司研制名为SPIN(integrated nuclear protection system)的反应堆保护系统，用于检测和紧急制动法国新型核电站反应堆。由于该系统的软件需要满足安全关键的要求，而市场上没有合适的COTS产品用于开发，Merlin-Gerin管理层不得不决定自己研制一套开发工具。

Merlin-Gerin公司聘请Lustre语言研发团队的两位成员Eric Pilaud和Jean-Louis Bergerand为负责人，开发了名为SAGA (Assisted Specification and Automatic Generation)的工具。**SAGA工具是基于Lustre语言设计的，可以混合图形和文本语法进行编程，并含有一个简单而高效的代码生成器。**不久，SAGA工具大获成功，顺利开发出了SPIN N4系列产品，并相继应用到了法国四个1450 MWe机组(N4 PWR)以及法国原子能委员会的几个研究堆。

在上世纪80年代末期，三大同步语言研究小组—Gerard Berry在索菲亚科技园的Esterel团队，Nicolas Halbwachs和Paul Caspi在格勒诺布尔的Lustre团队，Albert Benveniste和Paul Le Guernic在雷恩的Signal团队，三方通力协作相互借鉴，共同确立了同步语言编程的特点。



Lustre、Signal和Esterel三种同步语言

03 快速发展

到了上世纪90年代早期，数字设备巴黎研究实验室的Jean Vuillemin团队正在开发基于Perle FPGA的设备，以期应用于基于Alpha工作站的快速协处理器。**同大部分硬件设计团队一样，尽管很清楚基于数据流的硬件设计，但苦于无法有效地开发出包含大量门电路和寄存器的控制密集型复杂设计，于是他们寄希望于Esterel语言。于是Gerard Berry以咨询顾问的身份加入Jean Vuillemin团队，并成功设计了Esterel语言到电路的高效直接转换，同时避免了Esterel版本3中对大规模项目的状态空间异常暴涨问题。**这种新方法对Esterel语言的软件应用产生了决定性的影响，确保Estererl语言可推广到更大规模的应用程序开发。经过改进和优化，Esterel版本4也于1992年正式发布。

版本4虽然解决了版本3中因程序规模增大后状态空间暴涨的问题，但是却增加了额外的因果关系方面的约束，限制了用户在循环设计方面的应用习惯。1995年前后，Gerard Berry参考了普林斯顿大学电机工程学院Sharad Malik教授关于循环电路的论文，并沿用了加州理工大学Tom Shiple关于布尔电路的观点，逐渐加深了对因果关系的理解。新增了构建语义 (constructive semantics) ，并最终开发出了Esterel版本5。

位于以色列雷霍沃特(Rehovot)魏兹曼科学院(the Weizmann institute of Science)的David Harel教授在上世纪80年代发表过图形化状态机StateChart的设计，但是StateChart不太适用于涉及通信、并发、抢占功能的反应式系统。在StateChart基础上，CNRS成员兼Nice-Sophia Antipolis大学I3S实验室教授Charles André发表了符合同步假设的图形化状态机SyncChart。1997年，Esterel团队用Ocaml语言开发了scg2strl工具，支持将SyncChart转换为Esterel语言，在进行模块化改进后形成了Esterel版本6。版本6在达索航空，泰雷兹，Thomson CSF公司等多个项目上都有成功的应用案例。

90年代末期，Gerard Berry与Intel公司的Michael Kishinevsky先生开始合作扩展Esterel语言，使其可以定义任何类型的真实同步电路。**这需要语言定义的深度扩展及修改新的编译功能，以便能够支持任意的数据路径和位操作结构等内容，这些工作为后来Esterel版本7的发布奠定了坚实的基础。**

同期，Merlin-Gerin公司逐渐感到疲于长期投入资源来研制并维护SAGA工具，就与位于图卢兹的Verilog软件公司协作，进行SAGA工具的商业化开发。有趣的是，当时同样位于图卢兹的Aerospatiale公司，现在属于Airbus集团，在开发空客A320机型的线传飞控系统(fly-by-wire flight control)时，遇到与Merlin-Gerin公司相同的安全关键方面的问题，并因此也设计了一个内部工具SAO(Computer Assisted Specification)，**SAO也具备自动代码生成功能。**

在获悉相关情况后，Verilog公司牵头联系了Aerospatiale公司，并协商成立了由Aerospatiale公司，Merlin-Gerin公司和Verilog公司三方合作的共同体，并最终研制出了综合SAGA和SAO功能的新工具——SCADE。

1993年，Verilog公司与CNRS的Lustre团队合作创立了VERIMAG联合实验室，Verilog公司聘请VERIMAG实验室的Daniel Pilaud负责领导SCADE团队。不久SCADE工具成功地应用到了欧直的EC135/155，空客的A340-600机型，Thales Rail Signaling System的香港地铁等项目。后来Daniel Pilaud根据在Aerospatiale公司合作的成果，与INRIA的Alain Deutsch博士创立了PolySpace Technologies公司，2007年PolySpace Technologies公司被Mathworks公司并购。而VERIMAG联合实验室的创始人Joseph Sifakis博士由于“在将Model Checking发展为被硬件和软件业中所广泛采纳的高效验证技术上的卓越贡献”获得了2007年的图灵奖。

04 融合发展

在90年代中后期，瑞典的Prover-Technology与SCADE团队合作，通过采用同步观察(Synchronous Observers)技术来定义属性(Properties)和假设(Assumptions)等约束，**将其基于可满足性问题(SAT-based)设计的工具Prover嵌入到SCADE产品中，极大地提升了SCADE产品的验证能力，从此SCADE具有了在模型级进行形式化验证的功能。**90年代末期Verilog公司先被法国CS集团并购，又被瑞典Telelogic公司并购。

成立于1984年的Simulog公司是从INRIA分拆出来的，包括Gerard Berry在内的许多INRIA学者都是Simulog的联合创始人，Simulog公司是INRIA科研成果成功转化的案例之一。Simulog公司开始负责将Esterel产品商业化。

1999年，Simulog公司分拆出Esterel语言相关产品，成立了爱斯特尔技术公司 (Esterel Technologies) 。Gerard Berry教授作为爱斯特尔技术公司首席科学家(Chief Scientist Officer)于2002年荣膺法国科学院院士，之后相继获得法国技术研究院院士，欧洲科学院院士。

2001年，爱斯特尔技术公司并购了Telelogic公司的SCADE业务，开始着手融合Esterel语言和基于Lustre语言的SCADE产品。由于Lustre是声明式的，侧重描述数据流，但不支持状态机；Esterel语言是命令式的，侧重描述控制流，但支持的状态机SyncState较复杂，不宜通过认证。于是爱斯特尔技术公司决定通过几个同步语言概念的借鉴与融合，形成新的SCADE语言。

这项研发的学术方面工作由时任巴黎第十一大学的Marc Pouzet教授，与爱斯特尔技术公司的Jean-Louis Calaco和Bruno Pagano共同完成。Marc Pouzet教授使用其设计的融合Lustre语言和ML语言(函数式编程语言)两者特性的Lucid Synchrone语言扩展了SCADE语言，其中的ReLuC编译器是新版SCADE语言对应的代码生成器(KCG)的原型，该编译器也是用Ocaml语言编写。

另外，在新版SCADE语言中除了添加状态机功能之外，爱斯特尔技术公司还新增了勒诺布尔大学Florence Maraninchi教授设计的迭代器(用于循环设计)等一系列高阶运算功能，新增了由UPMC，INRIA，ENS Paris三个机构联合成立的PARKAS小组设计的Zelus语言中关于常微分方程扩展功能等。值得一提的是，**Marc Pouzet教授获得了2016年度INRIA创新奖，表彰他在同步语言领域的精深造诣，以及为推动SCADE语言发展做出的卓越贡献。**

05 品牌壮大

2005年爱斯特尔技术公司扩展了SCADE品牌，**SCADE旨在成为面向安全关键嵌入式领域的、基于模型的、覆盖全生命周期应用的工具。原SCADE产品更名为SCADE Suite，适用于控制软件的逻辑建模。**

2006年爱斯特尔技术公司收购了Thales Avionics和Diehl Aerospace联合研制的**嵌入式图形显示设计工具IMAGEIMAGE，并重新定义品牌名为SCADE Display。**原IMAGE工具曾经在空客的A380和A400M，达索航空的阵风战机，苏霍伊的支线客机等项目上有成功应用。

2009年爱斯特尔技术公司与CEA LIST研究所组成了名为LISTEREL Critical Software Lab联合研发实验室。不久该实验室推出了基于SysML语言的架构设计工具，并定义品牌名为SCADE System，后又更名为SCADE Architect。区别于传统**基于SysML的架构设计工具，SCADE Architect支持在SysML的基础进行封装定制，扩展出了符合ARP 4754A流程的航空嵌入式系统设计解决方案 (支持导出符合ARINC 429,ARINC 664，ARINC 653等协议的ICD；支持基于AADL 2.2版本对航电非功能属性进行建模、实现虚拟系统的集成；支持FACE: Future Airborne Capability Environment架构(最高3.0版本))、扩展出了符合ISO 26262流程的汽车嵌入式系统设计解决方案(支持AUTOSAR标准)。**

同年，为了专注于安全关键系统领域的业务，爱斯特尔技术公司将研制多年的旨在简化电子系统级(ESL)设计和系统级芯片(SoC)设计的EDA工具Esterel Studio 出售给了Synfora公司。而2010年，Synopsys公司又收购了Synfora公司。尽管如此，SCADE依然可以通过定制与部分硬件设计语言进行桥接或转换。

2011年爱斯特尔技术公司推出SCADE Lifecycle产品，用于帮助系统和软件开发人员进行产品的全生命周期管理。

2012年爱斯特尔技术公司推出SCADE ARINC 661解决方案，可用于符合ARINC 661标准的交互式座舱显示系统的设计。SCADE ARINC 661是业内唯一的、以基于模型的方式完全实现ARINC 661标准版本4和版本5中定义的所有控件 (Widget)的解决方案。当前兼容支持的最高版本为ARINC 661版本6(含93个Widget，15个Extension) 。

同年，Ansys收购爱斯特尔技术公司，并将其归于Ansys的系统事业部(System Business Unit)。

2015年Ansys发布了SCADE R16版本，从这时开始SCADE品牌产品的版本号同Ansys所有产品的版本号保持一致。

2016年Ansys推出了SCADE Test产品，用于实现基于模型的软件验证。SCADE Test中包含可通过DO-178C/DO-331 TQL-5级工具鉴定的自动化的认证测试环境（QTE: Qualified Test Environment）。支持在开发平台上对SCADE Suite模型和SCADE Display进行功能测试(含图形的像素比对功能)和覆盖分析(含模型级和代码级覆盖分析结果)；支持与主流目标机测试平台的桥接，重用模型级的测试用例。

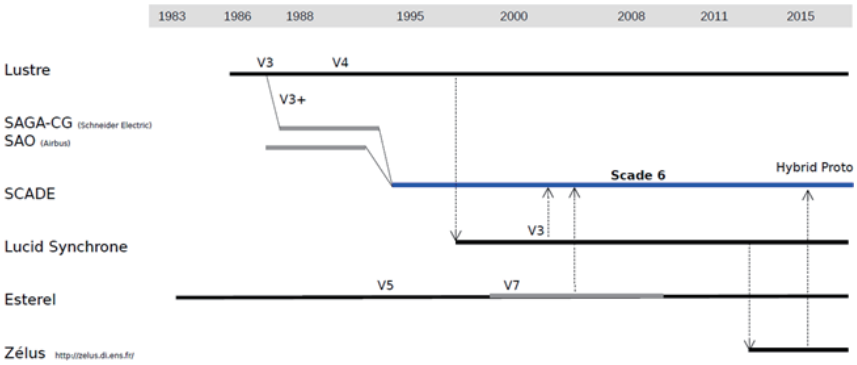
2019年Ansys推出了品牌名为SCADE Vision的基于AI感知的系统测试产品，该产品是Ansys与美国卡内基梅隆大学联合开发的AI智能感知算法验证套件。SCADE Vision是基于网络前端Web图形界面和云端部署的后台搜索引擎的，允许用户配置输入视频/图像存储库、感知算法/待测卷积神经网络 (CNN: Convolutional Neural Network) 等，**可自动识别包括自动驾驶车辆在内的各类感知系统中的潜在漏洞，帮助用户查找和分类这些系统中可能出现不安全行为的边界状况，有效降低AI智能感知算法的验证效率并快速定位AI智能感知算法的潜在缺陷。**

2020年Ansys发布了SCADE 2020版本。

年份	版本号
2010	6.1
2011	6.2
2012	6.3
2013	6.4
2014	R15
2015	R16
2016	R17
2017	R18
2018	R19
2019	R2019
2020	R2020

SCADE版本
与发布年份的关系

通过以上发展历史回顾可以看出，SCADE系列产品是伴随着众多理论研究和工程领域的实践应用而逐步发展起来的。现代SCADE语言至少综合了图表2所示6种语言的特性，这也奠定了其坚实的理论基础。



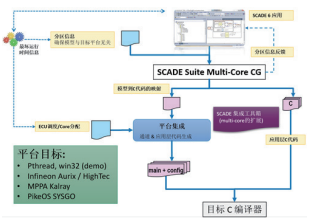
SCADE语言的渊源

Lustre语言及其派生的语言和工具

注：Lustre语言是开源的，在学术领域有较多研究。SCADE语言可认为是由Lustre语言发展而来的商业化版本之一，但SCADE语言包含更多的特性，可无缝连接Ansys的相关产品，不论在商业领域还是军事领域，都有更多成功的大型工程应用。

06 展望未来

相对于单核处理器的局限性，多核(Multi-Core)处理器由于其更高的能效逐渐被考虑用于安全关键领域。**SCADE多核相关的研发计划是：生成可在多核目标上高效执行的代码，并依然保留原有的精确、无歧义的、确定性语义和平台无关性。**当前主要与Infineon Aurix (Hightec)，Kalray MPPA (Kalray)和SYSGO PikeOS (S3P)几款产品合作，已根据部分客户的需求有了方案应用，未来将在工业实践中使该功能商业化。



SCADE多核代码生成器工作流

在形式化方法领域，主要的开发目标是将Prover Technology与RATP两家公司创立的HLL语言融入新版本产品的形式化流程中。HLL语言有助于在安全关键系统领域中使用模型检查技术进行形式化验证，当前主要有Prover Technology，RATP，Systerel，SafeRiver，Ansys五家公司参与到HLL语言相关的工作中。

在人工智能领域，主要的开发目标是将传统的基于模型的开发验证流程与新型AI研发流程相结合。当前的主要合作伙伴是空客，将在未来欧洲设想的未来空中作战系统(FCAS: Future Combat Air System)的高级无人机飞行控制系统中率先应用该项创新。

参考文献

[1] Halbwachs N. A synchronous language at work: the story of Lustre[C]//Proceedings. Second ACM and IEEE International Conference on Formal Methods and Models for Co-Design, 2005. MEMOCODE'05. IEEE, 2005: 3-11.
[2] Berry G. The constructive semantics of pure Esterel[J]. 1999.
[3] Halbwachs N, Caspi P, Raymond P, et al. The synchronous data flow programming language LUSTRE[J]. Proceedings of the IEEE, 2002, 79(9):1305-1320.
[4] Benveniste A, Caspi P, Edwards S A, et al. The synchronous languages 12 years later[J]. Proceedings of the IEEE, 2003, 91(1):64-83.
[5] Bergerand J L, Pilaud E. SAGA: A Software Development Environment for Dependability Automatic Controls[J]. Ifac Proceedings Volumes, 1988, 21(18):11-16.
[6] Harel D. Statecharts: a visual formalism for complex systems[J]. Science of Computer Programming, 1987, 8(3):231-274.
[7] Charles A. Representation and analysis of reactive behaviors: A synchronous approach[C]//Computational Engineering in Systems Applications, CESA. 1996, 96: 19-29.
[8] Pilaud D. Example of technologies push in the embedded market[J]. 1999.
[9] Colaço J L, Pagano B, Pouzet M. A conservative extension of synchronous data-flow with state machines[C]//Proceedings of the 5th ACM international conference on Embedded software. ACM, 2005: 173-182..
[10] Maraninchi F, Morel L. Arrays and contracts for the specification and analysis of regular systems[C]//Proceedings. Fourth International Conference on Application of Concurrency to System Design, 2004. ACSD 2004. IEEE, 2004: 57-66.
[11] Verimag Laboratory. Lustre, a Synchronous Dataflow Language. [EB/OL].http://www-verimag.imag.fr/DIST-TOOLS/SYNCHRONE/lustre-v6/doc/lustre.pdf
[12] Colaço J L, Pagano B, Pasteur C, et al. Scade 6: from a Kahn Semantics to a Kahn Implementation for Multicore[C]//2018 Forum on Specification & Design Languages (FDL). IEEE, 2018: 5-16.
[13] Benoît Dupont de Dinechin. MPPA Manycore Processor:At the heart of Intelligent Systems. [EB/OL]. https://www.european-processor-initiative.eu/wp-content/uploads/2019/10/Kalray-NEWCAS-2019.pdf. Jul,2019

Ansys 中国 | www.ansys.com.cn

咨 询 电 话：400 819 8999 **邮箱：**info-china@ansys.com **中国分公司：**北京 上海 成都 深圳

官方微博 官方微信