SCADE-DISPLAY 调研报告

# 1 SCADE Display的起源

2001年，为了给A380的新型座舱显示系统选用合适的图形开发工具，法国Thales Avionics公司评估了当时业界的几款产品，结果都不甚理想。于是Thales Avionics就与德国Diehl Aerospace公司协作，结合现有产品的通用属性和空客特有的业务需求，联合开发出了一套适用于**航电系统图形设计**的工具集。

该工具集由Thales负责开发名为IMAGE的基于模型的集成开发环境；Diehl负责开发名称为CoCOON的代码生成器，该代码生成器用于将IMAGE模型转换为由Diehl自研的名称为GLI的图形库接口,而GLI图形库可运行于由Thales和Diehl联合研制的具有自主知识产权的目标机平台中(SGL+RTOS+CPU+GPU)，其中的SGL为自研的标准图形库Standard Graphic Library。

2006年，爱斯特尔技术公司收购了这套工具集的平台无关模块，并将其升级打造成了商用货架产品（COTS），以便其更广泛地适用于安全关键行业嵌入式系统的人机交互界面开发。新工具的品牌名定义为SCADE Display，其余的对应关系如图表1。

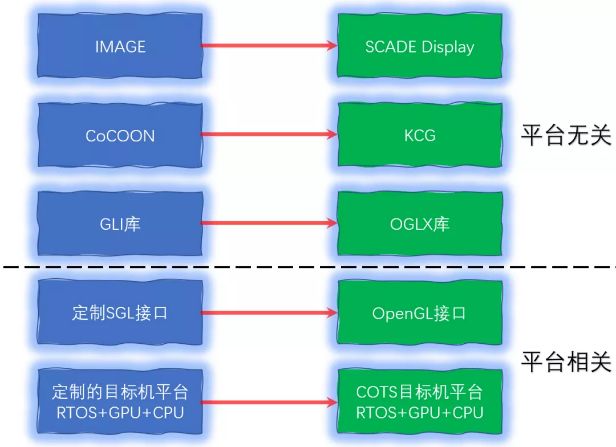


图1  现SCADE Display与原IMAGE的对应关系

# 2 基于SCADE Display的HMI解决方案

经过20多年的演进，当前SCADE Display已是Ansys公司旗下的产品，其最新版本号为R2021。SCADE Display可用于航空航天、国防、轨道交通、核能重工、汽车电子等安全关键领域的嵌入式系统的人机交互界面开发。



图2: SCADE Display的应用场景

常规的基于SCADE Display的人机交互界面开发解决方案如图表3。



图3: 常规的基于SCADE Display的人机交互界面开发解决方案

在画面左侧，可使用SCADE Display的建模器进行“所见即所得”的基于模型方式的人机交互界面设计，建模器提供了包含大量通用控件库和航空相关的专用控件库，同时，还支持用户封装自定义的控件库。

在画面中部，可使用SCADE Display对设计完毕的模型进行简单的验证工作，包括定制规则的模型检查，模拟仿真和桥接其他工具的联合仿真。更专业更复杂的大规模模型级验证工作可由SCADE Test实现(将在后续文章中做详细介绍)。

在画面右侧，1.可使用SCADE Display KCG将设计和验证完毕的模型生成为C代码，结合可定制的OGLX库，这些C代码可在支持开放图形库(OpenGL: Open Graphics Library)标准接口的驱动和不同操作系统(Windows, Linux, VxWorks, Green Hills, PikeOS, Android, iOS等)上运行；2. 可使用SCADE Display将模型自动生成详细设计文档；3.可使用鉴定包数据，通过特定安全关键的行业认证工作。

# 3 SCADE Display 集成开发环境

跨平台的专业图形API首推OpenGL，它是可用于渲染2D、3D图形的跨语言、跨平台的应用程序编程接口。当前OpenGL的最新版本为4.6。OpenGL下又有OpenGL ES和OpenGL SC等子集，分别发展到了3.0和2.0版本。

OpenGL ES (OpenGL for Embedded Systems)是OpenGL的子集之一，主要针对手机、平板电脑和游戏主机等嵌入式设备而设计。OpenGL ES 是从 OpenGL 裁剪定制而来的，去除了包括glBegin/glEnd，四边形(GL\_QUADS)、多边形(GL\_POLYGONS)等复杂图元在内的许多非必需的特性。经过多年发展，现在主要有两个版本，OpenGL ES 1.x 针对固定管线硬件的，OpenGL ES 2.x 针对可编程管线硬件。

OpenGL SC(OpenGL - Safety Critical Profile) 也是OpenGL的子集之一，主要针对安全关键行业(例如：航空航天、国防军工、轨道交通、核能重工、汽车电子等)中的认证服务和业务应用。它简化了认证工作，保证了安全关键行业要求苛刻的实时系统的可靠性，并便于安全关键图形相关应用程序的移植。当前主要有两个版本，2009年发布的OpenGL SC 1.0.1定义了OpenGL ES 1.0固定功能图形管线的高安全度子集。2016年发布的OpenGL SC 2.0 是OpenGL ES 2.0的子集，使用基于GLSL的可编程渲染器以实现更强的图形功能，包括提高性能和降低能耗。

实践中，如图4所示，安全关键行业的长周期项目会在开发的不同阶段采用不同平台组合来设计和验证HMI应用，这样代码之间的差异可能就会很大，将极大地影响项目的研制进度。

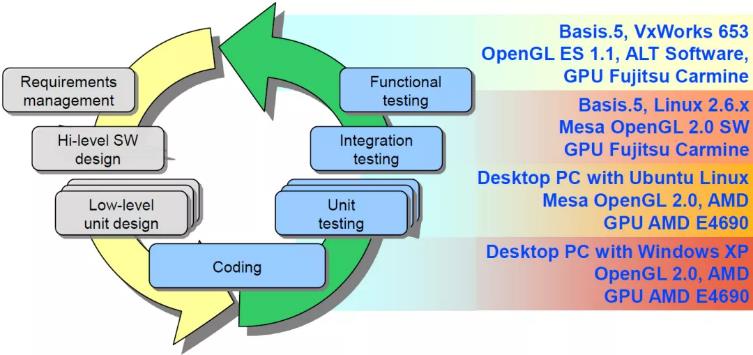


图4: 长周期项目在不同开发阶段采用不同版本的平台

随着项目的进展，由于驱动或硬件升级等原因，修改OpenGL源代码的代价就会很高。而有了SCADE Display这样的基于模型的图形界面设计工具，就可自动适配对应版本的OpenGL标准，从繁琐易错的手写编码方式中解脱出来，SCADE Display是以“所见即所得”的方式精准地设计人机交互界面，便于工程师之间的交流和改进。

SCADE Display支持的OpenGL特性包括基本图元绘制、遮罩设置、纹理贴图、α透明通道设置、光晕设置及反锯齿处理等；支持矢量和点阵字体、支持包括中文在内的多国语言开发。支持在模型级或代码级与视频、地图(2D或3D)的叠加集成。

复杂的人机交互界面设计需要有复杂的后台逻辑控制，**同类竞争对手的产品在为图形添加控制逻辑这方面的建模能力较弱**。而SCADE Display集成开发环境支持在图形模型中添加Assign内置控件，SCADE Suite模型和C代码共三种方式来实现的行为功能，实现控制逻辑和显示画面的完美融合。

# 4 SCADE Display KCG

SCADE Display KCG用于将设计完毕的模型和资源生成C代码，并生成该过程对应的日志文件和映射文件，供配置管理和认证时的评审。资源文件包括颜色表、线宽表、线型表、纹理表、字体表和渐变表等。

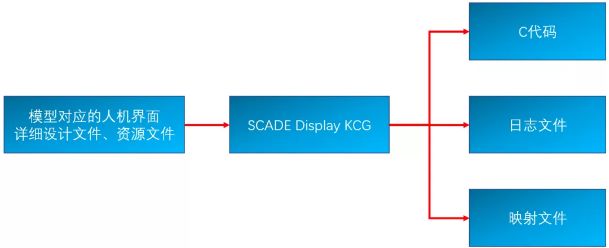


图5: SCADE Display KCG的输入和输出

SCADE Display KCG的认证包可提供满足相关安全关键行业标准(DO-178C/DO-330 TQL-1、IEC 61508-1/3:2010 SIL3、EN 50128 SIL3/4、ISO 26262-8:2011 ASIL D的认证材料。

下图是常规的基于SCADE Display生成代码的集成方案。如果不做额外的定制，用户只需要重点关注最左侧main函数等平台相关的手写代码即可，最大程度地减少了软件集成的工作量。值得一提的是，SCADE Display KCG的C代码并不包含对OpenGL API的直接调用，该任务由OGLX库来实现。

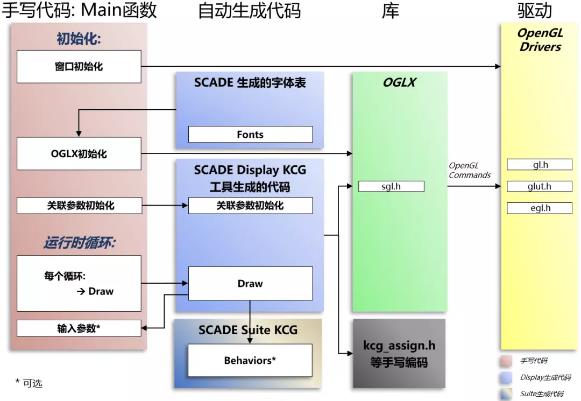


图6: SCADE Display KCG代码的集成方案

# 5. SCADE Display OGLX

SCADE Display相对于同类竞争对手产品的杀手锏，就是其**超强的可移植性**，而这主要就是归功于SCADE Display OGLX库的独到设计。OGLX是**O**pen**GL** e**X**tension to SCADE Display KCG中蓝色字母缩写，OGLX作为高层的图形库，负责将SCADE Display KCG生成的C代码中的SGL命令调用转换为底层的符合特定OpenGL标准的命令调用。当前OGLX对下列3类集合中5个版本的OpenGL是完全支持，对其他版本和子集的OpenGL是兼容支持。

如下图所示，SCADE Display模型生成C代码后，再配合OGLX库的不同编译选项配置，可轻松将模型生成的代码联编为适用于不同版本OpenGL驱动的目标码，运行在基于特定图形帧缓存的GPU上。OGLX在其中起了桥梁的作用，在增强了OpenGL标准中诸如向量计算，顶点绘制，复杂的遮罩设置，纹理贴图，透明处理，光晕反锯齿处理，显示列表处理，帧缓存处理等底层操作的基础上，使得用户可以在模型级轻松设计，同一模型不做任何改动，就可移植到不同的平台，用户只需要专注于业务逻辑本身即可。

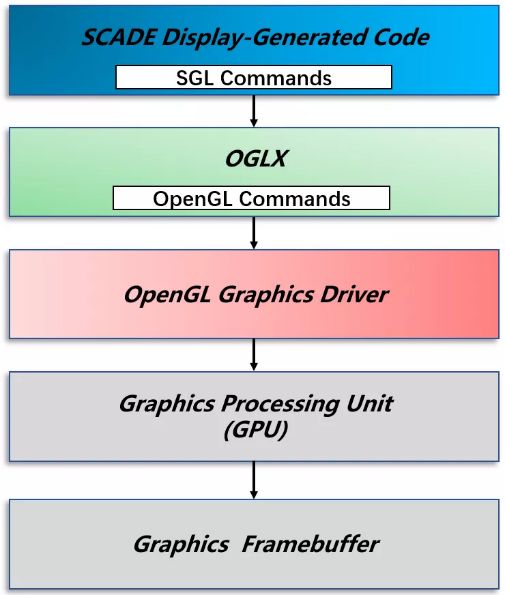
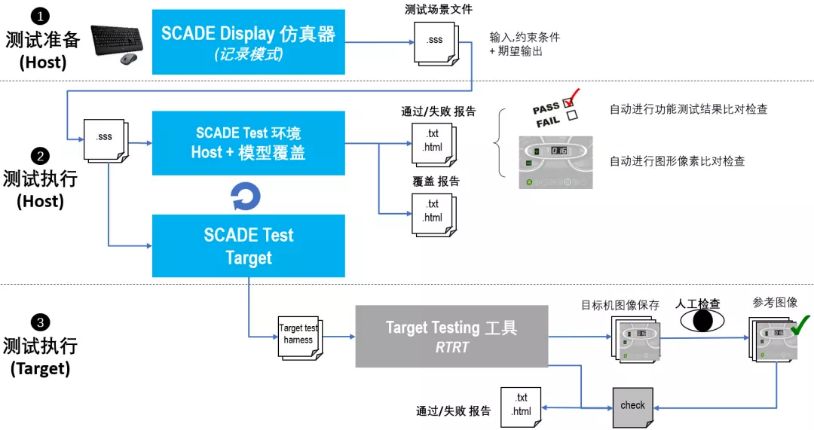


图7: SCADE Display KCG & OGLX的调用路径图

# 6. SCADE Display基于模型的研制

复杂的大规模的SCADE Display模型级的验证工作需要依赖SCADE Test工具。用户可以在SCADE Test中进行模型的功能测试，覆盖分析和像素比对。这些工作既可以在Host机(与开发平台相同的机器)上执行，也可将在Host机上积累的测试用例转换后，在目标机上执行相关验证。详细内容会在后续SCADE Test专题文章中介绍。**同类竞争对手的产品在基于HMI模型的认证级验证方面的能力较弱。**



图表8: SCADE Display的基于模型的验证工作流

# 7. SCADE相关开源软件情况收集

为了满足国内某安全攸关领域的需求,由清华大学主创的 L2C编译器的开发始于2010年9月, 其目标是设计实现一个经过形式化验证的可信编译器（已有开源版本）, 其源语言与scade一致，都是面向领域的同步数据流语言Lustre, 目标语言是C, 最终可用作相关领域数字化仪控系统的安全级代码生成器.据了解，迪捷软件的modelcoder也是基于开源版L2C在做此类工具开发。

目前国内做得比较好的全数字仿真公司是北京迪捷数原科技有限公司，他们公司的主营产品是SkyEye。天目系统的前身来源于清华大学陈渝老师的在研开源项目，迪捷数原公司拿过来之后进行了市场化的更新升级之后，市场前景还不错。

开源工程地址：https://sourceforge.net/projects/skyeye/files/skyeye/

# 8 SCADE Display基础培训教程（学习、了解功能）

<https://www.zhihu.com/zvideo/1398314069397549057>

<https://download.csdn.net/download/qq_43312879/15876358?utm_source=bbsseo> //教程文档及翻译

# 9. 参考文献

[1] Bonnet, P. and A. Coupier. 3.5.2 Integrated Modular Avionics: a challenge in Tools & processes. [J] (2004).

[2] Pavel Fedorov. Primary Flight Display Prototype with Combined Vision System designed using SCADE Suite & Display [EB/OL] SUGC 2010 (内部文档)

[3]. Frank Köster. Vehicle automation and advanced driving assistance systems are being streamlined using Ansys SCADE capabilities. [EB/OL] Ansys ADVANTAGE Volume IX Issue 2 2015 (内部文档)

[3] Leanna Rierson.安全关键软件开发与审定——DO-178C标准实践指南[M]. (崔晓峰). 北京：电子工业出版社，2015年6月  
[4] Todorov V, Boulanger F, Taha S. Formal verification of automotive embedded software[C]//Proceedings of the 6th Conference on Formal Methods in Software Engineering. ACM, 2018: 84-87.  
[5] Xavier Leroy Trust in compilers, code generators, and software verification tools [EB/OL].  
[6] 宋俊. LTLNFBA: LTL 公式到 Büchi 自动机的转换[D]. 西安电子科技大学, 2014.  
[7] 王瑞. 基于 SAT 的符号化模型检验技术研究[D]. 国防科学技术大学, 2014.  
[8] Sun Y. Strengthening fonctional validation of critical system by using Model Checking: Application to Instrumentation and Control systems in nuclear power plants[D]. 2017.  
[9] Li W, Kan S, Huang Z. A Better Translation from LTL to Transition-Based Generalized Büchi Automata[J]. IEEE Access, 2017, 5: 27081-27090.  
[10] 刘志锋. 模型检测中关键技术的研究及其应用[D]. 南京大学, 2011.  
[11] 解定宝. 混成系统有界模型检验优化技术研究[D]. 南京大学, 2016.  
[12] Miller S P. Bridging the gap between model-based development and model checking[C]//International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, Berlin, Heidelberg, 2009: 443-453.  
[13] SCSDV\_2019R1\_M01\_Tool. Ansys SCADE Suite Design Verifier and Formal Verification [EB/OL]. Ansys SBU. 2019（内部文档）  
[https://xavierleroy.org/talks/ERTS2018.pdf](https://link.zhihu.com/?target=https%3A//xavierleroy.org/talks/ERTS2018.pdf). Embedded Real Time Software and Systems, 2018-02-02  
[14] Bhatt D, Madl G, Oglesby D, et al. Towards scalable verification of commercial avionics software[M]//AIAA Infotech@ Aerospace 2010. 2010: 3452.  
[15] Wang J, Zhan NJ, Feng XY, Liu ZM. Overview of Formal Methods. Journal of Software, 2019, 30(1): 33-61(in Chinese). [http://www.jos.org.cn/1000-9825/5652.htm](https://link.zhihu.com/?target=http%3A//www.jos.org.cn/1000-9825/5652.htm)  
[16] 形式化方法导论[M]. 清华大学出版社 , 张广泉, 2015

[17] Yong.Wang. FBW flight control system Airworthiness&Safety Design[R] CAIF 2019  
[18].飞行二次元 [EB/OL]. 电传飞控，民航客机的“绝顶内功”[http://www.sohu.com/a/247579057\_685248](https://link.zhihu.com/?target=http%3A//www.sohu.com/a/247579057_685248" \t "_blank). 2018年8月  
[19] 军中利刃 吃透电传飞控：国产三代机上天的头号战役[EB/OL] [https://www.jianshu.com/p/3ae6592abd0e](https://link.zhihu.com/?target=https%3A//www.jianshu.com/p/3ae6592abd0e). 2015年11月  
[20] 张建军. 神奇的电传侧杆控制器[J]. 大飞机, 2017 (12): 82-83.  
[21] Goupil P. AIRBUS state of the art and practices on FDI and FTC in flight control system[J]. Control Engineering Practice, 2011, 19(6): 524-539.  
[22] Airbus Customer Services A380-800 Flight Deck and Systems Briefing for Pilots [EB/OL]. [http://www.smartcockpit.com/docs/A380\_Briefing\_For\_Pilots\_Part%202.pdf](https://link.zhihu.com/?target=http%3A//www.smartcockpit.com/docs/A380_Briefing_For_Pilots_Part%25202.pdf). 2006年3月  
[23] Van Den Bossche D. The A380 flight control electrohydrostatic actuators, achievements and lessons learnt[C]//25th international congress of the aeronautical sciences. 2006: 1-8.  
[24] Traverse P, Lacaze I, Souyris J. Airbus fly-by-wire: A total approach to dependability[M]//Building the Information Society. Springer, Boston, MA, 2004: 191-212.  
[9] Jean-Philippe BEAUJARD. Airbus Flight Control Systems and future evolutions[EB/OL]. [https://slideplayer.com/slide/3110094](https://link.zhihu.com/?target=https%3A//slideplayer.com/slide/3110094). 2003年  
[10] Bochot, Thomas, Pierre Virelizier, Hélène Waeselynck and Virginie Wiels. Model checking flight control systems: The Airbus experience[J] 2009 31st International Conference on Software Engineering - Companion Volume (2009): 18-27  
[25] Jean Souyris. Formal Methods at Airbus: Experience Feedback[EB/OL]. [http://projects.laas.fr/IFSE/FMF/J1/P04\_JSouyris.pdf](https://link.zhihu.com/?target=http%3A//projects.laas.fr/IFSE/FMF/J1/P04_JSouyris.pdf). (2012)  
[26] Boulanger, Jean-Louis. Static analysis of software: The abstract interpretation[M]. John Wiley & Sons, 2013.  
[27] Souyris J, Wiels V, Delmas D, et al. Formal verification of avionics software products[C]//International symposium on formal methods. Springer, Berlin, Heidelberg, 2009: 532-546.  
[28] Laurent O, Michel P, Wiels V. Using formal verification techniques to reduce simulation and test effort[C]//International Symposium of Formal Methods Europe. Springer, Berlin, Heidelberg, 2001: 465-477.  
[29] J.SOUYRIS.Industrial Use of CompCert on a Safety-Critical  
Software Product[EB/OL]. [http://projects.laas.fr/IFSE/FMF/J3/slides/P05\_Jean\_Souyiris.pdf](https://link.zhihu.com/?target=http%3A//projects.laas.fr/IFSE/FMF/J3/slides/P05_Jean_Souyiris.pdf) 2014年2月  
[30] Leroy X. Formally verifying a compiler: what does it mean, exactly?[C]//ICALP. 2016: 2:1-2:1.  
[31] Kosmatov, Nikolai and Julien Signoles. Runtime Assertion Checking with Frama-C[R]. (2013).  
[32] 崔少轩. 基于抽象解释理论的程序循环边界分析[D].南京航空航天大学,2018.  
[33] 尚书,甘元科,石刚,王生原,董渊. 可信编译器L2C的核心翻译步骤及其设计与实现[J]. 软件学报,2017,28(05):1233-1246.  
[34] Blazy S, Laporte V, Maroneze A, et al. Formal verification of a C value analysis based on abstract interpretation[C]//International Static Analysis Symposium. Springer, Berlin, Heidelberg, 2013: 324-344.  
[35] Jean-Charles DALBIN. SCADE for AIRBUS critical avionics systems[EB/OL] SUGC 2009 (内部文档)  
[36] Wiels V, Delmas R, Doose D, et al. Formal verification of critical aerospace software[J]. 2012.  
[37] Moy Y, Ledinot E, Delseny H, et al. Testing or formal verification: Do-178c alternatives and industrial experience[J]. IEEE software, 2013, 30(3): 50-57.