

电子战三维视景软件适应性架构设计

李晓燕, 王立冬, 郭金良, 张阳
(中国人民解放军63880部队, 河南 洛阳 471003)

摘要: 针对电子战三维视景软件的需求变化特征, 提出了其适应性软件架构的设计。研究了领域需求的变化, 提出了需求的抽象和分析方法, 进行了电子战三维视景软件的适应性架构设计, 具体说明了软件的需求处理、需求与架构的对应关系, 及其逻辑架构和运行架构, 给出了领域需求变化时通过架构调整进行系统应用扩展的方法, 表明适应性架构可快速响应变化需求, 缩短开发周期, 减少开发成本。

关键词: 视景软件; 电子战; 适应性架构; 逻辑架构; 运行架构

中图分类号: V271.4; TP311 **文献标志码:** A **文章编号:** 1671-637X(2014)04-0086-05

Adaptive Architecture Design for 3D Scene Simulation Software of Electronic Warfare

LI Xiao-yan, WANG Li-dong, GUO Jin-liang, ZHANG Yang
(No. 63880 Unit of PLA, Luoyang 471003, China)

Abstract: Considering the different requirements to 3D visual software of electronic warfare, the design of an adaptive software architecture was put forward. After studying the change of domain requirements, we proposed a method for demand abstraction and analysis, and designed an adaptive software architecture of 3D visual software for electronic warfare. The requirement processing, corresponding relation of requirement with architecture, and logical architecture were described concretely. In addition, the system expanded application method was given through the structure adjustment when the demand changes, which shows that the adaptive architecture can response to the changing requirements quickly, shorten the development cycle and reduce development costs.

Key words: scene simulation software; electronic warfare; adaptive software architecture; logical architecture; run architecture

0 引言

电子战是使用电磁能和定向能控制电磁频谱或攻击敌军的军事行动^[1], 其态势具有复杂、抽象等特征, 使决策者难以准确、快速地做出判断。电子战三维视景采用计算机图形图像技术, 构造仿真对象三维模型, 以三维模型和特效的实时动画表现真实的场景, 使电子战试验、作战态势、进程及对抗效果看得见、听得到, 为决策人员提供一个共享的视听环境, 具有重要的军事应用价值^[2]。

传统的电子战视景软件只注重功能的实现, 面对不同的作战想定和战场态势表现需求, 软件经常需重新进行设计或大面积适应性改造。这种自顶向下的软

件设计方法对需求变化的适应能力差, 且往往不适用于高层设计^[3-4], 代码的移植性和软件系统的可扩展性差。为做到快速响应需求、缩短开发周期和成本, 如何构造面向需求变化特性的适应性软件架构就成为开发人员关心的重要话题。

软件架构是部件和子系统相互作用来构成系统的组织形式及在系统层面进行最好设计和分析的系统属性^[5-7], 它是软件的核心元素, 决定了系统的主体结构、宏观特性和基本功能。从高层抽象和系统抽象角度, 提供了控制软件复杂性和质量、支持软件开发和复用的途径。面对电子战三维视景软件代码重用、灵活扩展、质量控制、不断升级的需求, 本文通过分析需求和软件架构之间的关系, 归纳领域模型, 建立了可适应需求变化的软件架构, 提高了软件的开发效率。

1 需求变化特性的处理方法

需求的不准确、不完整、不稳定是其变化特性的根

源,几乎所有软件在其生命周期中都会发生需求变化^[8]。如图1所示,传统的软件开发存在从用户需求直接到软件实现的情况,经常是定制的,灵活性差。而基于软件架构的设计方法体现了架构在软件生命周期中的特殊性和关键作用,即它是需求与实现之间的桥梁。

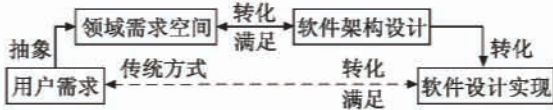


图1 两种需求处理方法的对比

Fig. 1 Comparison of two requirement handling methods

需求变化特性的处理是软件适应性架构开发的基础,在面向领域应用的软件架构开发方面,要定义需求域,对需求进行功能的抽象分类,并区分稳定需求和变化需求以提高架构的适应能力,累积的大量领域需求也有助于开发方提醒客户,在开发初期减少需求的不确定性,故须加强需求提取和问题分解,此为对策应对的基础。

1.1 需求抽象

需求的抽象是从需求关注点和软件架构设计的视角出发定义需求。通常用户的需求可能是不完整、不准确和不系统的,需要从设计者而非用户的角度认识那些在软件架构设计中须解决的关键问题,并将其进行领域抽象,是一个将用户需求转化为领域设计问题集合的过程。该过程的特点是填补了传统用户需求到软件设计间的鸿沟,实现了需求到设计的过渡媒介。该集合使得领域软件架构在设计初期就具备一定的完备性考虑,较少地依赖设计者的经验和技能。

1.2 需求分解

分解是应对设计复杂性的基本手段。需求分解是通过分层、分类的方法构建领域需求空间,形成体系化的需求结构。首先是依据对系统具有关键作用的高层和全局性需求,为软件架构划定范围;然后从低一层次确定顶层问题域内的重要需求,进而确定低一层次的软件架构问题范围;如此下去即可建立层次化的需求结构,以此作为架构设计的重要功能性要求。在此基础上,依据需求结构抽象出非功能性需求,作为整个需求结构的重要必须补充。通常非功能性需求是不易被发现和表达的,如设计约束、质量属性、可维护性、配置、界面要求等。

从上述分析可知,抽象所得需求的独立性和关联性确定了需求的可复用性,领域需求空间的完备性确定其应对领域开发的适应能力,影响领域软件架构的适应性设计。

2 电子战三维视景软件适应性架构设计

开发电子战三维视景软件的目标是建立一个包含作战想定编辑、场景制作、实时浏览及控制的开放基础平台,为不同类别、不同层次的电子战视景开发提供软件平台支撑,其架构设计的科学性直接影响到软件的二次开发效率。

2.1 软件的领域需求分析

电子战三维视景软件的领域需求包括功能需求部分和非功能需求部分,功能需求部分自上向下可分为集成、业务和数据访问层需求。通常情况下,集成层的需求是根据仿真目的编辑作战想定,完成各类数据、对抗场景的可视化、界面要求及战情的编辑输入等,业务层集中处理各种算法、响应各种命令等,用户需求通常不会细致到服务层和数据访问层。非功能性的需求经常是更加模糊,往往需要从顶层对需求进行抽象、分解和补充,而后形成较完整的需求结构。软件架构在需求结构上能够较为清楚地定义需要解决的问题和解决方法,能够在需求变化时及时确定架构的调整方案和影响范围。依据上述思想将电子战三维视景软件的需求进行分解,如表1所示。

表1 领域需求的分解

Table 1 Decomposition of domain requirement

功能需求 M	需求分解		
	P	A	
战情编辑	战术、战役级想定	进攻装备要素	
	防御装备要素	对抗装备要素	
	侦察装备要素	通信装备要素	
界面布局	菜单	场景内直接显示	
	工具栏	属性编译栏	
集成层	显示栏	多个视窗	
	航迹数据	装备属性	
	视点位置	实时帧率	
	仿真时间	仿真命令	
场景显示	关注点的地理坐标	两点间距离、方位等	
	平台飞行	雷达探测	
	侦察过程	侦察效果	
	干扰过程	干扰效果	
	指控过程	通信效果	
业务层	视点移动	视点切换	
	编辑、保存、调用	锁定某装备	
	运行方式	在线实时	
	显示模式	线框模式	全屏切换
		视窗的关闭打开	不同视窗的选择
	模型控制	鼠标拾取模型移动	重新编译、保存、调用 模型新位置
模型运动		模型放大缩小	
效果控制	效果及动态表现	效果显示、关闭	
	数据访问层	数据获取	邦员公布的数据 外部设备,如键盘鼠标 数据库 配置文件

用 R 表示需求域,可划分为多个主题,用 $M_1, M_2,$

..., M_n 表示, $R = \{M_1, M_2, M_3, \dots, M_n\}$ 。 M_i 又由若干个功能 P_1, P_2, \dots, P_m 组成, 每个功能对应系统中的一个软构件, 表示为 $M_i = \{P_1, P_2, P_3, \dots, P_m\}$ 。 同样, P_j 有若干个行为 A_1, A_2, \dots, A_k , 行为对应软构件中的实现方法, $P_j = \{A_1, A_2, A_3, \dots, A_k\}$ 。

从该示例中可以看出, 易变的需求都是位于上层, 有概括模糊和不清晰的特点, 体现在仿真目的的多样化和用户对人机界面、数据可视化等个性化要求上, 其对软件架构的设计影响较小, 经常是可以界定清楚的。 愈向底层, 用户愈是难以界定清楚, 需要设计方辅助决策, 但越底层的需求越具有功能稳定性的特点。 另外, 软件质量、可维护性等非功能需求按照高规格设计后, 其处理方法也是相同和可复用的。

2.2 软件架构方案的形成

软件架构方案的形成是从领域需求空间确定的需求结构导出软件架构的设计问题、再到合成软件架构的过程。 就电子战三维视景软件来说, 软件架构的方案要包括全局的组织和控制结构、设计功能分配、基础信息结构、构件间的通信关系、数据接口协议、非功能设计元素集成等。 当出现集成层功能需求的变化时, 选取新的问题解决方案, 通过构件关系将其融合到软件架构中去, 做到改动小、效果大, 体现架构的适应性。

2.2.1 逻辑架构设计

电子战三维视景软件的逻辑架构如图 2 所示, 包括集成层、业务层和数据访问层。

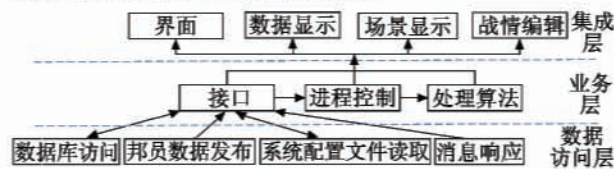


图 2 视景软件逻辑架构分层图

Fig. 2 Logical architecture of scene simulation software

集成层主要是窗口界面、数据及场景态势的展示及战情的录入。 该层需求常常会变动, 设计时应预留变更空间, 可灵活处理更改; 业务层由多个处理算法完成, 包括处理接口与各类数据的交换、对进程的控制及对模型驱动、场景效果实现等算法, 完成功能需求中控制部分的实现。 该层的需求是最稳定的, 用户基本不参与, 许多模块是通用的, 可进行重组使用; 数据访问层读取本电子战三维视景软件所需的各类数据, 主要完成功能需求中通信部分的实现。 该层所涉及的数据库和系统配置文件的读取、消息响应等是相对稳定的, 其变动主要体现在通信方式上, 或是 HLA, 或是 socket 网络通信等。 本层与集成层的不稳定性有所不同, 集成层的改动可能出现在任何时候, 而这一层在最初确定下来之后通常不发生变化。

为保证架构的适应性, 在设计时建立“需求 - 架构构件 - 功能实现”的关联链, 架构构件对应不同的需求, 构件间通过关联接口发生关系, 如图 3 所示。

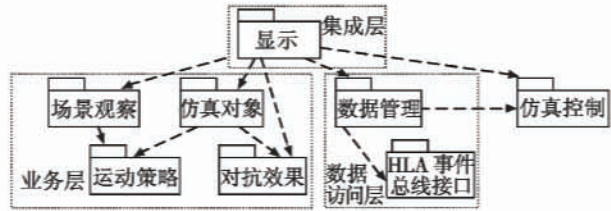


图 3 视景软件构件结构图

Fig. 3 Modules of scene simulation software

关联接口与构件的功能实现分离, 在架构层面对构件增删或更改时, 关联接口保持不变。 构件可以细化为更小的颗粒, 需求变化时, 通过关联链确定影响域, 对关联链条和构件进行较小更改实现新的功能。

此外, 考虑到集成层的易变性, 在实现时要注意构件设计的高内聚和松耦合特性, 高内聚提升构件的功能针对性, 松耦合提升构件的独立性。 这样避免集成层需求的调整带来大范围的更改, 从而预留出较大的操作空间。 同时, 还可以从架构上将集成层要处理的数据、战情设置等分离出来, 以配置文件的形式保存而不是直接封装到软件程序中, 实现调用算法和参数数据的分离, 修改时可修改分离出的文件或配置项, 达到提升变更空间的目的。

每个构件又由多个类组成, 以电子对抗效果类组成为例说明, 如图 4 所示。

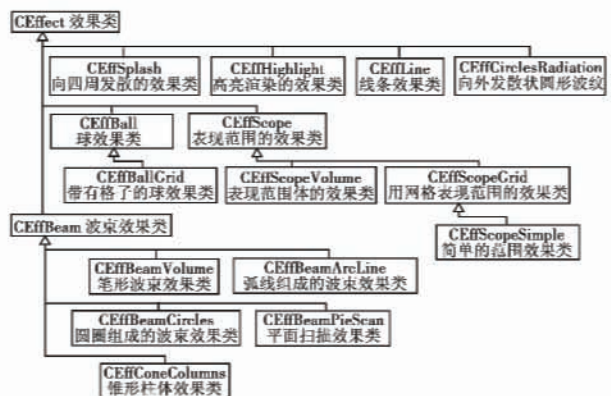


图 4 对抗效果构件类图

Fig. 4 Class diagram of countermeasure effect component

CEffect 为效果基类, 通过与仿真对象模块的 CObjSource 对象进行交互, 可以在符合条件(条件编程可控)的源和目标之间表现各种效果。 CEffect 下包括许多子类, 某些子类又作为父类包含多个子类, 在 CEffect 中封装了基本的操作函数, 其实现者只需关注怎样创建动态效果, 而无需关注如何把效果放置于场景中, 何时显示效果, 或者以怎样的频率更新效果等基础操作, 实现功能的高内聚和对外松耦合。

2.2.2 运行架构

电子战三维视景软件有着复杂的业务逻辑和互操作逻辑,为了优化关键资源的使用效率,采用了多条控制流并行或并发执行的运行架构。主线程为 Visual C++6.0创建的窗口系统,主线程不断执行,处理用户输入指令和显示输出数据信息,并负责建立、关闭和协调各线程的工作。子线程处理通信和场景显示,为并行工作模式。每个子线程执行过程中不断监听消息指令,一旦接到指令就立即处理各自负责的事务,处理完毕后将结果交由主线程处理。

软件架构运行如图5所示。

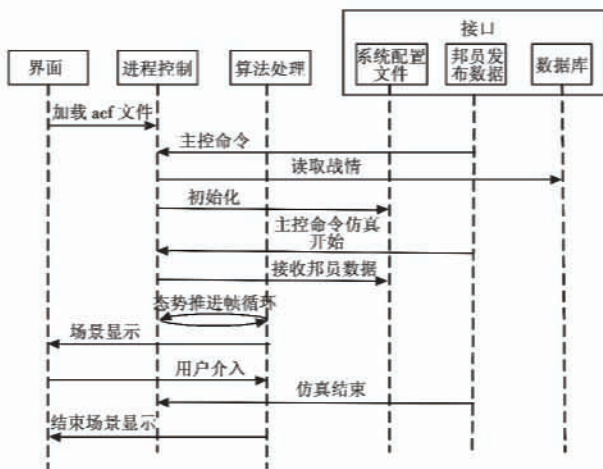


图5 视景软件架构运行图

Fig.5 Running architecture of scene simulation software

仿真开始前,从界面加载 acf 文件为程序配置参数,当收到主控邦员发出的初始化命令时从数据库中读取战情文件对装备模型布站并进行初始化。仿真开始后,接受邦员发送的数据进行帧循环,同时显示态势的推进过程,期间用户可介入参与,程序接收并响应用户命令,主控发出仿真结束命令后,视景软件结束仿真。

软件架构设计完成后,还需要通过软件架构来评测软件能达到的质量,文献[9]提供了8种架构分析方法的描述,可根据具体开发情况进行选择使用,主要是从功能性、可靠性、可用性、效率、可维护性和可移植性等几个方面进行架构质量的衡量。

3 适应性架构的扩展应用

电子战三维视景软件适应性架构的建立为其扩展和调整提供了极大便利,调整时可根据增加的功能需求确定影响的软件模块、再对模块的功能和行为做最小调整,并明确调整后的模块与其他模块的关系,从而从架构上确定影响域和开发代价。

某预警机雷达对抗视景软件的开发过程中,用户

对界面、通信方式及战情提出了新的需求,且预警机的运动方式在原视景软件中无相应的运动模块支撑。通过软件架构分析,改动将影响图3中集成层的窗口界面和战情编辑,业务层处理算法中的运动策略及接口模块中的 HLA 事件总线接口。依据影响域对框架进行修改设计。对界面控件进行调整不影响其他模块;新的战情只需按设定的标准格式重新输入,战情参数的调用不需更改;设计导弹算法,开发相应的子类加载到运动策略模块中去,主程序中对其调用方法保持不变;更改图3中 HLA 事件总线接口为 socket 网络通信,数据接收后在主程序的调用保持不变。

该软件表现了红蓝双方预警机雷达的对抗态势,即对抗过程中,红方预警机协同地面侦察装备发现蓝方预警机,指挥所接到报告后命令雷达及侦察单元对其跟踪定位并实施压制干扰和假目标干扰,同时命令航空兵突防攻击。蓝方预警机受到干扰后,侦测距离缩小,并发现假目标。在干扰过程中,红方航空兵突防到蓝方预警机附近,并实施导弹攻击击毁预警机。仿真中预警机侦测范围根据干扰功率的大小实时调整,飞机运动和干扰范围、功率等由相应的软件单元实时计算并通过网络传输至视景软件,视景软件实时表现飞行、干扰、突防情况和攻击弹道等过程和效果。图6是预警机雷达的侦测范围和通信效果图。软件更改效率评估表明,基于适应性架构设计的领域软件在面对新需求更改时,适应能力强,开发代价小。

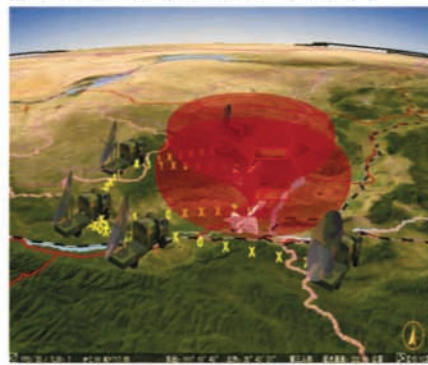


图6 预警机雷达对抗视景效果图

Fig.6 Demonstration of AWACS radar countermeasures

4 结论

相对于传统的软件设计方法,面向需求变化特性的适应性软件架构能够把握需求特点,由需求确定架构,在满足不同需求时再次开发的代价小,适应能力强,系统维护、升级方便。本文从软件适应性架构开发需求入手,分析了需求变化特性和处理方法,对电子战三维视景软件架构进行分析设计,给出开发实例,为基于适应性软件架构的领域软件设计开发提供了有益

参考。

参 考 文 献

- [1] ADAMY D L. EW102: 电子战进阶[M]. 朱松, 王燕, 译. 北京: 电子工业出版社, 2009.
ADAMY D L. EW102: A second course in electronic warfare [M]. Translated by ZHU S, WANG Y. Beijing: Electronics Industry Press, 2009.
- [2] 汪连栋, 张德锋, 聂孝亮, 等. 电子战视景仿真技术及应用[M]. 北京: 国防工业出版社, 2007.
WANG L D, ZHANG D F, NIE X L, et al. Electronic warfare scene simulation technology and application[M]. Beijing: National Defence Industrial Press, 2007.
- [3] 崔晓峰, 孙艳春, 梅宏. 以决策为中心的软件体系结构设计方法[J]. 软件学报, 2010, 21(6): 1196-1207.
CUI X F, SUN Y C, MEI H. Decision-centric software architecture design method[J]. Journal of Software, 2010, 21(6): 1196-1207.
- [4] 黄双喜, 范玉顺, 赵彧. 一类通用的适应性软件体系结构风格研究[J]. 软件学报, 2006, 17(6): 1338-1348.
HUANG S X, FAN Y S, ZHAO Y. Research on generic adaptive software architecture style[J]. Journal of Software, 2006, 17(6): 1338-1348.
- [5] KRUCHTEN P, OBBINK H, STAFFORD J. The past, present, and future for software architecture[J]. IEEE Software, 2006, 23(2): 22-30.
- [6] 温昱. 一线架构师实践指南[M]. 北京: 电子工业出版社, 2009.
WEN Y. Practice guidelines for front-line architect [M]. Beijing: Electronics Industry Press, 2009.
- [7] JACOBSON I, BOOCH G, RUMBAUGH J. 统一软件开发过程[M]. 周伯生, 冯学民, 樊东平, 译. 北京: 机械工业出版社, 2002.
JACOBSON I, BOOCH G, RUMBAUGH J. The unified process development process[M]. Translated by ZHOU B S, FENG X M, FAN D P. Beijing: China Machine Press, 2002.
- [8] 付赞, 李敏强, 陈富赞. 面向不确定需求的适应性软件体系结构设计[J]. 计算机科学, 2010, 37(12): 99-105.
FU Y, LI M Q, CHEN F Z. Adaptive software architecture design oriented to requirements uncertainty[J]. Computer Science, 2010, 37(12): 99-105.
- [9] DOBRICA L, NIEMELA E. A survey on software architecture analysis methods[J]. IEEE Transactions on Software Engineering, 2002, 28(7): 638-653.
-
- (上接第 72 页)
- [3] GOODSELL S J, GENG D, FEDRIGO E, et al. FPGA developments for the SPARTA project; Part 2[J]. proceedings of SPIE, 6272, Advances in Adaptive Optics II, 6272411-12, 2006, doi:10.1117/12.671646.
- [4] GOODSELL S J, GENG D, YOUNGER E J, et al. FPGA developments for the SPARTA project; Part 3[J]. Proceedings of SPIE, Astronomical Adaptive Optics Systems and Applications III, 6691031-12, 2007, doi:10.1117/12.735585.
- [5] 贾建禄, 王建立, 赵金宇, 等. 基于 FPGA 的自适应光学系统波前处理机 [J]. 光学精密工程, 2011, 19(8): 1716-1722.
JIA J L, WANG J L, ZHAO J Y, et al. Technology for adaptive optical wave-front processor based on FPGA[J]. Optics and Precision Engineering, 2011, 19(8): 1716-1722.
- [6] BASDEN A, GENG D L, MYERS R, et al. Durham adaptive optics real-time controller [J]. Applied Optics, 2010, 49(32): 6354-6363.
- [7] WANG C X, LI M, ZHOU L C, et al. Real-time implementing wavefront reconstruction for adaptive optics[J]. Proceeding SPIE, Adaptive Optical System Technology III, 5639199-204, 2004, doi:10.1117/12.580438.
- [8] ABREU R, CHADWICK D, AMICO ET AL R D. Modeling of the SAAO adaptive optics system [J]. Proceeding of SPIE, 3931272-284, 2000, doi:10.1117/12.384291.
- [9] PENG X F, LI M, RAO C H. A kind of FPGA-based correlating shack-hartmann wave-front processor [J]. Proceeding of SPIE, 7130, Fourth International Symposium on Precision Mechanical Measurements, 2008, doi: 10.1117/12.819702.
- [10] DUNCAN T S, VOAS J K, EAGER R J, et al. Low-latency adaptive optical system processing electronics [J]. Proceedings of SPIE, 4839, Adaptive Optical System Technology II, 2003, doi:10.1117/12.458955.