电子数据单工具链插件式管理系统设计

何睿[1,2] 张峻巍[1,2] 卢广佑[1,2] 王彧泽[1,2] 吕良庆[1,2]

[1](中国科学院复杂航天系统电子信息技术重点实验室(中国科学院国家空间科学中心) 北京 100190)

[2](中国科学院大学 北京 100049)

摘 要 电子数据单能够解决航天器星载部件的即插即用，其在空间数据系统应用时，需要一系列的软件工具支持电子数据单的管理与编辑操作。同时，随着工具数量的增加，对独立且可扩展的电子数据单工具链架构需求逐渐增高。本文结合电子数据单工具链的需求，设计了与分层架构融合的基于插件化架构的电子数据单工具链管理系统。本研究明确了系统软件架构、应用插件开发方法和接口规范，结合相关技术，提出了基于开放服务网关协议(Open Service Gateway Initiative, OSGi)和 Eclipse RCP 的工具链管理系统技术实现方案。结果证明，基于插件化架构的电子数据单工具链管理系统可以满足系统的实际功能与应用需求，并在工具的组织上满足独立性与可扩展性。

关键词 电子数据单 插件模式 工具链 OSGi 管理系统 RCP

中图分类号 TP315 TP391 V57 **文献标志码** A DOI:

Plug-in architecture design of electronic data sheet tool chain

HE Rui[1,2] ZHANG Junwei[1,2] LU Guangyou[1,2] WANG Yuze[1,2] LYU Liangqing[1,2]

[1](*Key Laboratory of Electronics and Information Technology for Space Systems(National Space Science Center, Chinese Academy of Sciences), Beijing 100190, China*)

[2](*University of Chinese Academy of Sciences, Beijing, 100049*)

Abstract The electronic data sheet can solve the plug and play of spaceborne components of spacecraft. When it is used in the space data system, it needs a series of software tools to support the management and editing of electronic data sheets. At the same time, with the increase of the number of tools, the demand for independent and scalable electronic data sheet tool chain architecture is gradually increasing. Based on the requirement of electronic data single tool chain, this paper designs an electronic data single tool chain management system based on plug-in architecture which integrates with layered architecture. This research defines the development method and interface specification of application plug-in, and proposes a technical implementation scheme of tool chain management system based on Eclipse RCP, combining with related technologies. The results show that the electronic data sheet tool chain management system based on plug-in architecture can meet the actual function and application requirements of the system, and meet the independence and scalability of the tool organization.

Keywords Electronic Data Sheet(EDS); plug-in mode; tool chain; OSGi; management system; RCP

0 引 言

随着航天技术的不断发展，航天器的种类、任务类型、功能等方面呈现出多样性。航天器数量逐步增多的同时，异构系统的需求相应增加，航天器之间、航天器内载荷之间逐渐暴露出接口设计多样化、各个项目应用协议不统一的问题，这需要在系统设计、研制、测试的各个阶段信息交互时，对数据采用统一的标准进行描述[1]。电子数据单（Electronic data sheet,EDS）是某个系统中一个部件的自描述信息载体，能对异构系统间数据传递格式标准化[2]。使用 EDS 作为载体可以实现系统内各部件之间的信息交换，进而实现部件即插即用。因此，上述数据描述需求可以采用电子数据单技术实现。

EDS 在空间数据系统应用时，需要一系列的软件工具，实现 EDS 的创建、管理、自动编辑、格式转换、校验等操作。这些软件工具依照 EDS 不同的标准和数

收稿日期: xxxx-xx-xx。北京市科技计划项目（Z201100003520006）。**何睿**，硕士研究生，主研领域：空间数据系统 。**张峻巍**，博士研究生。**卢广佑**，硕士研究生。**王彧泽**，硕士研究生。**吕良庆**，研究员，博士。

据对象，能基于具体的应用场景，开发整合为若干不同的 EDS 工具链[3]。从软件架构角度而言，为便于数据交互与数据管理，EDS 工具需要规范的接口开发标准以便于开发，且需要可扩展且独立的软件平台对其管理和维护。

1 EDS 工具链原理与特点分析

1.1 以上行遥控数据注入为例的 EDS 工作原理

具体来说，EDS 是指在描述某个系统中的部件过程中应该具备的数据信息，其文件的内容主要是所要描述部件的数据信息集合。该集合可以是各种形式的信息对象，不同的信息对象均需能被本系统内的其他部件或外部系统所识别[4]。目前 EDS 已经在互联网领域和机械制造行业内得到了广泛的应用，并逐渐应用于航天领域[5]，如空间数据系统咨询委员会（Consultative Committee for Space Data Systems，CCSDS）提出了可扩展标记语言遥测遥控数据交换标准（XML Telemetric and Command Exchange，XTCE）和航天器星载接口业务领域电子数据单标准(Spacecraft Onboard Interface Service Electronic Data Sheet,SEDS)等相关 EDS 标准[6]。

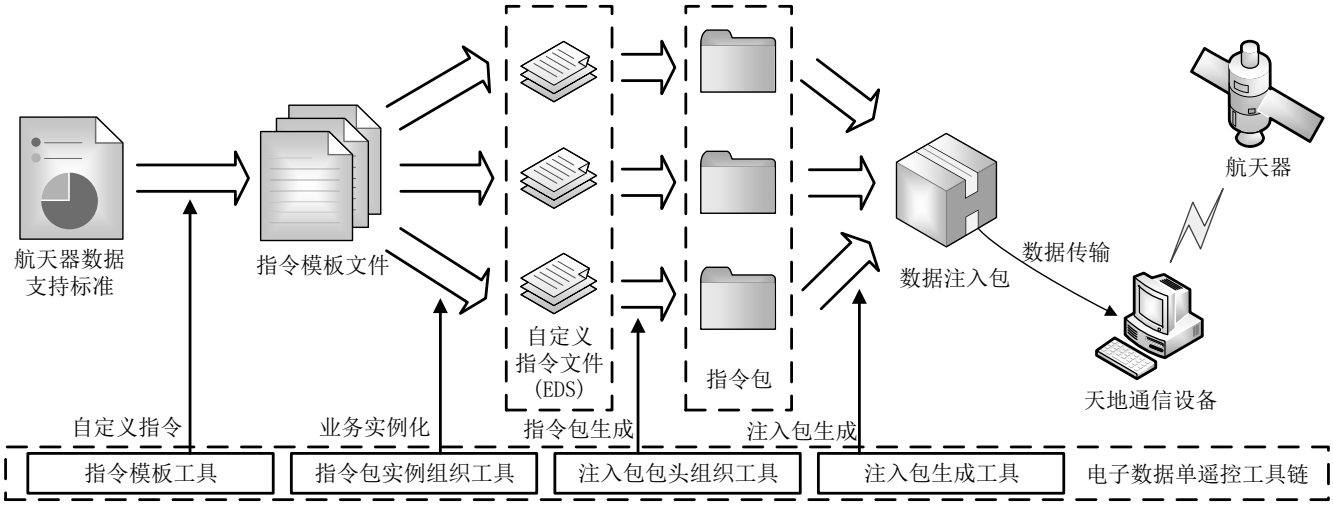

图 1 EDS 工具链工作原理图

以上行遥控数据注入为例，上行注入指令的格式是以相应的空间包标准为基础的。用户需要在地面端基于包标准设计相应的指令包和注入包，之后通过天地通信设备将数据包上行至航天器。在上述过程中，EDS 的使用可以促进航天器之间和航天器载荷之间的数据规范化。通常，设计者在航天器数据系统的地面设计阶段与测试阶段应用 EDS，基于相应的航天器数据支持标准和包标准，设计相应的指令格式，生成对应的 EDS 模板文件。之后，对应的 EDS 模板文件经过实例化、包头组织与注入包生成等过程，可以生成对应的注入包，用于数据上行。相应的工作原理图如图 1 所示。

若基于传统的 EDS 处理过程，上述相应操作过程均为人工进行，EDS 文件需要用户自行填写。这种方法随着 EDS 数量的增多，会对用户的使用造成困难[7]。因此，上述过程中从数据标准到 EDS 模板文件的转换、业务实例化、指令包和注入包的生成都需要参照数据化研发思想，使用特定的工具来辅助完成相应的操作，每一步操作均有特定的工具来对应相应的数据转换与操作[8]。为了便于工具间的管理，将这些工具进行集成，即可得到 EDS 工具链。工具链能充分简化用户对 EDS 的编写工作，提高用户的工作效率，使航天器数据管理规范化、统一化[9]。

1.2 EDS 工具链集成需求

针对 EDS 工具链的设计，国内外已有相应的一系列研究，例如基于 XTCE 的遥测数据处理软件[10]、SEDS 的设备解析工具[11]、综合电子空间数据系统等[12]。目前的工具链设计方法，核心在于软件的功能实现与数据的传递、处理，可以满足用户的基本功能需求，但当用户需求逐渐增多时，工具链数量不断扩增，功能不断扩展，此时就需要一定的软件架构辅助 EDS 工具链集成，以便在一个“工具箱”中对各个工具链进行统一的组织与管理。

EDS 工具链集成的架构需求如下。

第一，从单一工具的功能角度而言，工具仍然需要实现各类型 EDS 的编辑、生成、转化、管理等相关功能。

第二，从单一工具的开发角度而言，工具链集成

后应当可扩展，即 EDS 工具应当具有统一且通用的数据接口，以便支持集成工具链的功能扩展。

第三，从工具之间的架构组织角度而言，工具链集成后，各个工具之间仍需保持独立性。一方面，每一个工具具备在操作系统环境下独立运行的条件。另一方面，若其中一个工具无法正常使用时，不应影响其余工具的运行以及软件的整体运行。

第四，从各个工具的整体管理角度而言，整个工具链集成后，工具在架构中应实现“即插即用”，在统一的架构中增删、使用、更新与维护。

总之，整个工具链的集成除需要满足必要的功能需求外，工具之间、工具链之间还需具备较强的独立性，保持低耦合。集成后的“工具箱”需要保持整个工具包必要的功能模块同时，还要保证工具之间的独立性，即保持“通用—专用”架构，以便满足独立和可扩展的需求。

基于上述需求，本研究提出了一种基于插件化方法的 EDS 工具链管理系统软件架构。

2 基于插件的 EDS 工具链管理系统整体架构

2.1 传统插件化架构及优化方法

插件是一种软件组件，主要用来在原有系统上添加一些扩展功能，使得系统具有定制化的能力。用户通常希望能够在不影响原有系统的条件下完成新功能的添加，实现这一目标的方法称为插件化（Plug-in）[13]。插件化有利于降低模块间的耦合度，有利于各模块和项目的维护更新以及快速整合交互[14]。

传统插件化架构由两大组件构成——系统内核与插件组件。其中，插件组件负责实现各种具体的业务逻辑，可以根据业务功能不断扩展[15]。系统内核除用于管理各种插件外，还用于实现与具体业务无关的公共功能。所有的插件组件均在同一层次下，且由系统内核负责调用。根据实际需求，插件化开发中常采用接口方式定义可被扩展的功能点，插件也能主动向外暴露外部接口用于外部调用[16]。

为了使插件化架构适用于 EDS 工具链，并满足独立性与可扩展性的要求，本研究对传统插件化架构进行了部分改造。

首先，为了保证插件之间的独立性，将插件中使用的基础服务与插件的业务逻辑分离，本设计把传统插件化架构中层级相同的插件划分为通用服务插件和应用业务插件。通用服务插件负责提供基础性功能、可复用功能与底层调用，应用业务插件负责实际的业务逻辑问题。其中，一个通用服务插件可被多个应用业务插件调用。

其次，为了保证平台的可扩展性，一方面，在插件的基础之上设置统一的外部接口层，用于插件与软件平台的通信和插件与软件外通信。另一方面，将传统架构中的系统内核功能进行简化，仅在内核中实现插件管理中心模块，并在管理中心实现接口管理和插件管理的功能，其余内核功能从插件框架中分离。最后，将所有的插件和接口纳入到可扩展框架中，由插件管理中心对框架中的插件与接口进行加载、管理和维护。

基于上述优化前后的插件化架构示意图如图 2 所示。

在图 2 中，各个工具可以解耦运行，工具之间的运行互不干扰。同时，可扩展框架提供了一种功能扩展的方式。本研究基于上述优化后的插件化架构，进一步完善并设计了 EDS 工具链管理系统的整体架构。

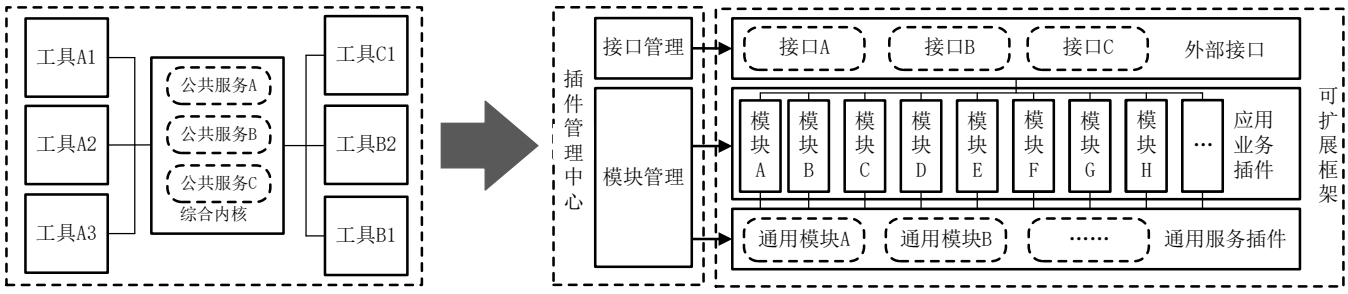

图 2 优化前后的插件化架构示意图

2.2 基于优化后插件化架构的系统概要设计

在本研究中，工具链管理系统采用传统的软件分层架构模式[17]，并将插件化架构融入软件其中，以便保证软件模块间的低耦合与独立性。首先，系统基于原先的可扩展框架增加通用插件工具，作为其他工具的基础支持，将原先的插件可扩展框架自上而下分为视图层、应用业务层和插件服务层。其次，考虑到系统的内核运行以及底层数据与运行环境支持，将系统的插件服务层之下分为通用服务层、平台驱动层、数据管理层与运行环境层。

同时，纵向来看，架构中的模块根据功能划分为

可扩展插件模块、插件管理中心和平台扩展服务。可扩展插件模块是插件的重要组成部分，其中通用功能插件为应用功能插件提供接口，应用功能插件主要实现各个工具链的主要功能。插件管理中心是插件化架构的管理模块，用于对各种插件工具进行管理。平台扩展服务是整个平台运行的基本环境和支撑模块，为整个系统提供基础功能和服务。

架构中，每个层次分别承担不同的职责，自下而上业务逐渐具体化和实例化，一般来说，下层为上层提供接口服务和基础运行条件环境等，基于上述七层结构的电子数据单工具链管理系统整体架构图如图 3 所示。下文简述各层主要功能及层次之间的关系。

运行环境层主要指平台的基础运行环境，该环境通常为工具链管理系统的生产环境，保证整个工具链管理系统的基本运行。

数据管理层包括分布式数据库系统。数据库系统包含软件的关系型数据与每个插件的数据管理。不同的插件可以创建不同的数据库便于管理和即插即用。同时，操作系统层的文件也会在数据库系统中建立索引与操作日志。

平台驱动层主要包括软件框架运行环境。该层次主要作用是保证数据库、操作系统与上层软件架构的通信，并基于相关软件框架，实现上层工具链管理系统的构造。

通用服务层是整个软件通用的相关预置功能，主要分为平台工具服务、数据驱动服务和运行支持服务。平台工具服务主要包括整个平台运行时的部分通用工具。数据驱动服务指数据在软件层面的驱动与读写模块。运行支持服务包括软件正常运行必要引入的部分支持模块，相关模块均来自软件外部。

插件服务层和应用业务层二者均为插件所在层，其分别对应插件化架构中的应用业务插件和通用服务插件。通用服务层和应用业务层由插件管理中心负责的模块管理功能对插件进行统一维护。其中插件服务层和通用服务层的区别在于通用服务层的相关服务可以供平台扩展服务、插件管理和可扩展框架调用，但不支持即插即用。而插件服务层相关服务只能支持可扩展框架调用，且支持即插即用。

通用服务层用于提供工具链的基础服务和开发工具包等通用工具，例如文件解析（XML，XSD 文件解析等）、文件校验（XML、XSD、二进制文件校验）、包生成、包格式转换等。对于不同的标准，核心的处理方法是通用的，因此通用服务层为应用业务层提供了通用解决方案，且该层次通过通用工具的高内聚保证系统的独立与可扩展。

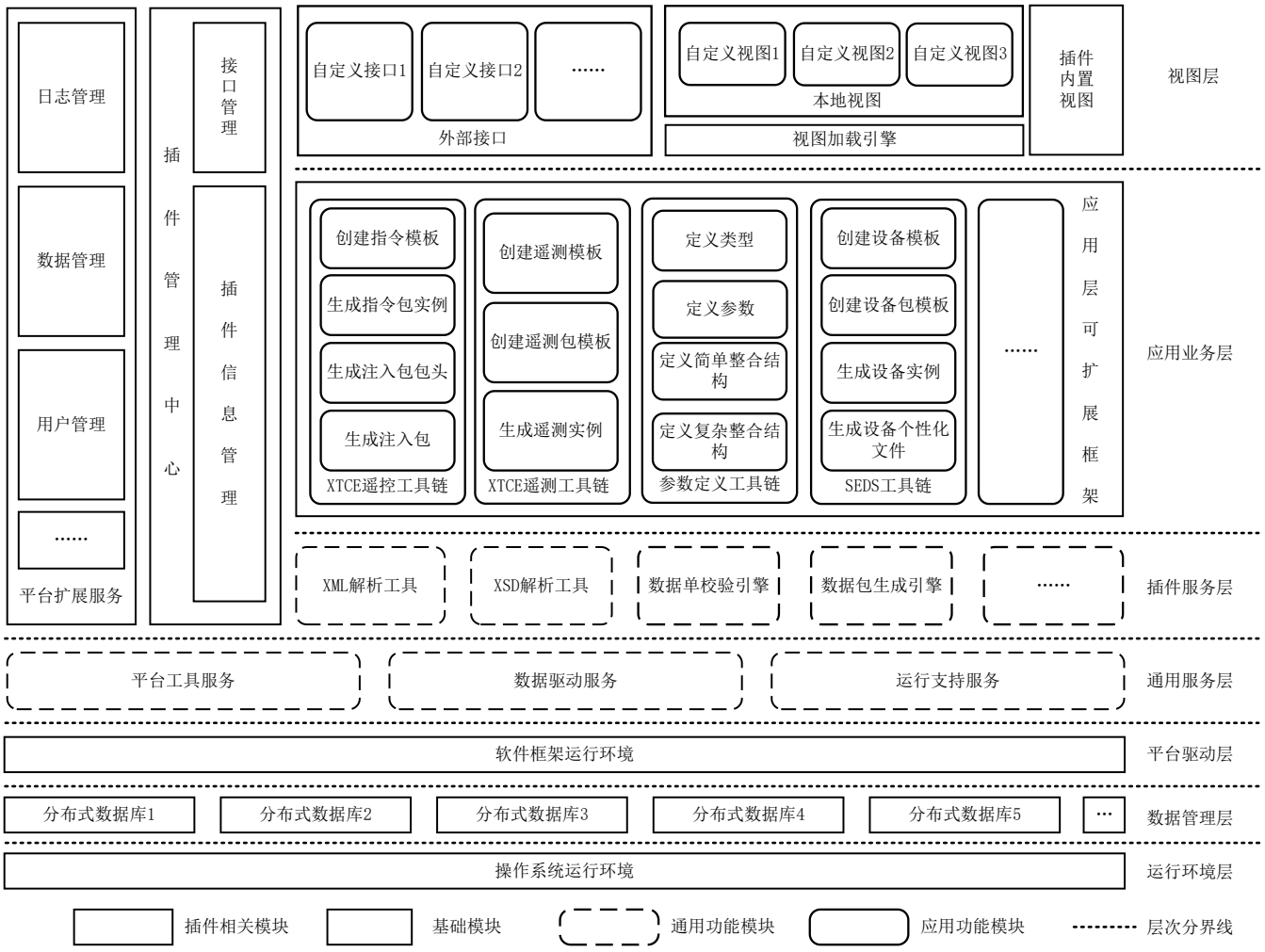

图 3 插件式整体架构图

应用业务层从架构上看，通过横向与纵向关系保证工具链的独立性和可扩展性。从横向而言，应用业务层由若干 EDS 工具链组成，同时基于可扩展框架，可以继续对工具链进行扩展，以便丰富整个工具链管理系统的功能，保证不同信息对象工具链的可扩展性。纵向而言，每个工具链中由针对同一个信息对象的处理工具组成。在功能上，各个工具之间互相独立，保证支持一个信息对象的工具链可配置性。

视图层主要由视图加载引擎、本地视图和外部接口组成。视图加载引擎负责将插件的内部逻辑翻译为可视化界面，供用户操作与反馈。本地视图模块依照视图加载引擎的反馈与通信，选择相应的自定义视图模板加载出相应的可视化界面。数据接口部分为整个工具链管理系统预留部分，用户可以不通过可视化窗口，而是调用数据接口使用工具。视图层主要解决了整个系统的功能可视化与应用问题，将工具的使用方式暴露于外部。其基于应用业务层实现了模块与用户的动态对接。

贯穿多层次的模块有插件管理中心与平台扩展服务。插件管理中心分为接口管理和模块管理两部分，负责对所有的插件进行管理。同时，作为整个插件管理的内核，负责对底层数据库进行操作和各个层次的模块间通信。平台扩展服务提供了整个管理系统的通用功能，用于支持软件运行，例如用户管理、日志管理、数据管理等。该接口与插件管理中心层次相同，级别相同，为软件提供运行的基本服务。

整个工具链管理系统的多层架构保证了软件的低耦合与可扩展，对于 EDS 工具链而言，工具之间的分类可以更为清晰，相应的工具管理方法也有所优化。

3 应用业务插件面向对象的设计开发方法

3.1 应用业务插件架构与接口设计

应用业务插件对应 EDS 相关的操作工具，这些操作工具在整个平台整合之前，是零散、无规律的状态。在平台统一架构之后，相关的工具需要统一接口，以便在一个完整的架构中实现工具的插件化，保证各个

工具的独立性和整个架构的可扩展性。

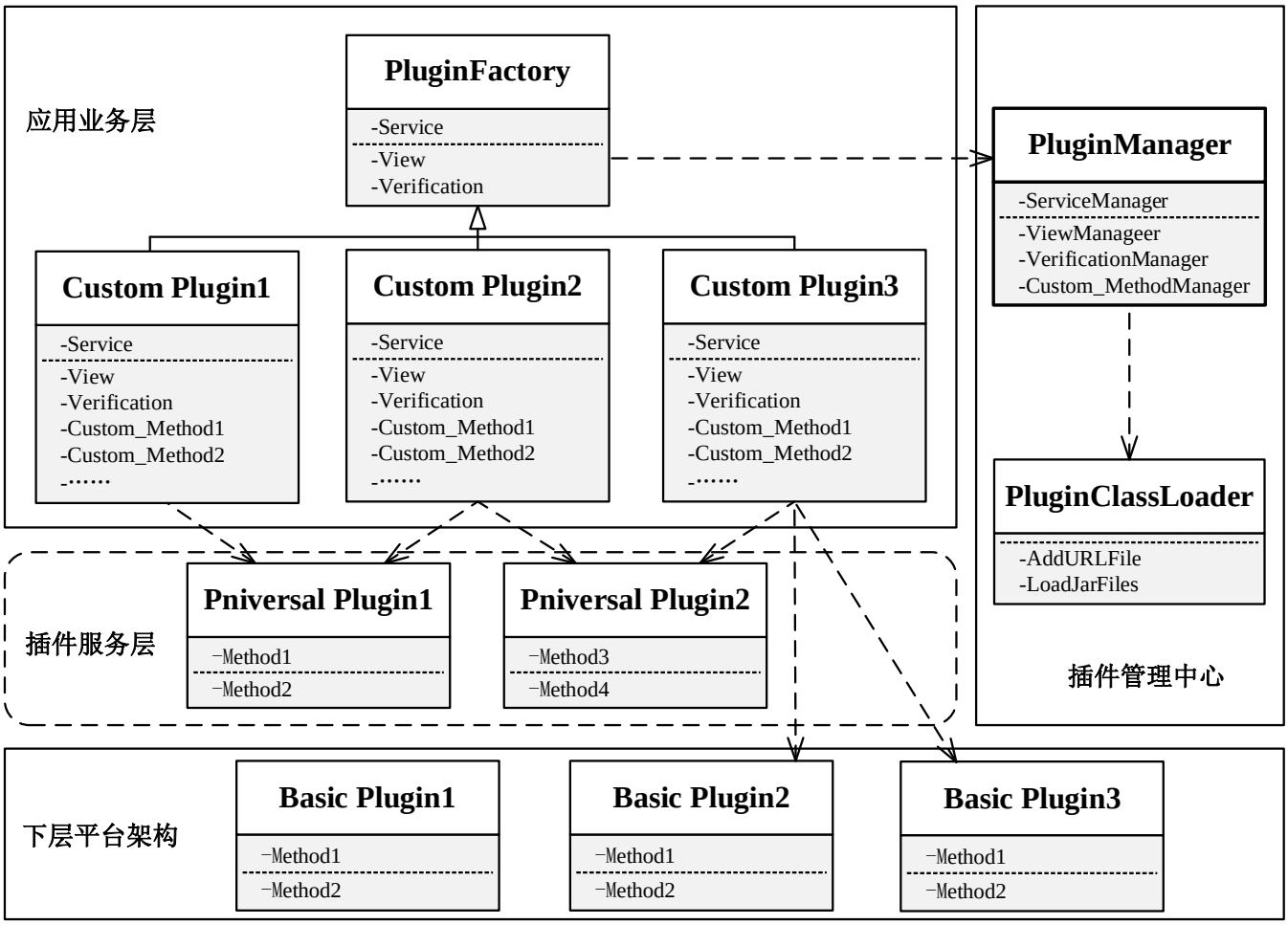

图 4 平台核心架构的 UML 类图及其层次示意

为统一各个工具的实际调用接口，本研究基于工厂模式实现插件的构造。在应用业务层，所有的工具需要基于“插件工厂”（Plugin Factory）完成核心接口的实现，主要包括 Service、View 和 Verification 三个核心接口。具体接口的输入输出及其去向如表 1 所示。

表 1 插件接口信息表

序号	接口名称	接口输入	接口输入来源	接口输出	接口输出去向
①	核心业务(Service)	业务输入数据	应用插件	业务输出数据	对应业务入口
②	视图(View)	视图文件	应用插件	视图	视图加载引擎
③	验证(Verification)	插件验证信息	应用插件	插件验证结果	插件管理中心

Service 作为最主要的接口，用于实现工具的核心业务和功能。通过相应格式的输入与输出，将插件本身的核心业务功能同其余非核心业务操作剥离开来，使插件本身“小而精”，专注于其业务的实现，也有利于简化插件开发。View 用于传递插件相关视图，将插件的相关视图映射到视图加载引擎。Verification 用于在加载插件前验证插件的完整性和可用性。除此之外，插件本身也可以构造自己的相关业务方法，用于内部的调用处理、数据库操作等。每一个插件都可以选择性地依赖若干插件服务层提供的通用插件或者通用服务，每一个通用插件和通用服务也可以被若干应用业务插件依赖。

本研究中，平台架构的核心 UML 类图如图 4 所示。对于插件管理中心而言，若所有的插件接口统一后，则对统一的接口进行统一的维护，即可对插件进行统一管理。为实现插件的即插即用，插件本身是可以独立运行的。因此如需对插件业务调用解码，需要插件管理中心依赖基于底层虚拟机驱动改造的 ClassLoader，对插件本身调用之后，这种调用和接口的实现统一由插件管理中心的管理模块实现。对于平台而言，该架构只需要对插件管理中心的管理模块外部接口进行调用和实现，即可完成对所有插件的维护

和控制。

平台在运行时，为了插件与架构之间的解耦合，将插件分为不同的状态类型。不同状态类型的插件，相关属性等有所区别。根据架构的相关设计，平台插件的状态可以分为插件卸载状态、插件加载状态、插件活动状态和插件停用状态。每个插件无论处于何种位置，均为其中上述四种状态其中之一。插件默认为卸载状态，当插件装入平台架构时，插件状态变更为加载状态。之后当插件调用时，插件会在加载状态和活动状态之间切换。插件可以在架构中随时停用，并在停用状态时选择恢复使用或者卸载。平台插件状态转移图如图 5 所示。

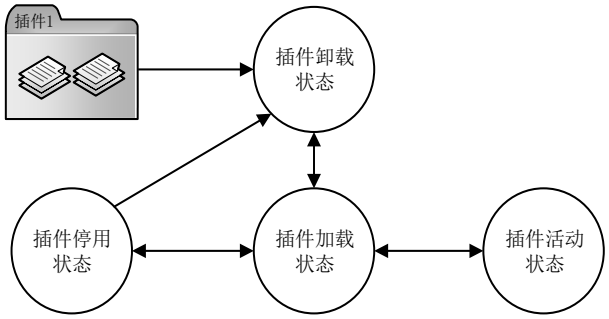

图 5 平台插件状态转移图

3.2 工具链具体业务在架构中的使用场景和应用方法

对架构构建完成后，需要将相关的工具基于工厂模式单独开发，并实现即插即用过程。因此，对于工具链中的工具，需要明确通用服务层、插件服务层和应用业务层每个工具的具体名称和功能。

通用服务层主要包括文件流的生成输出和编辑器管理相关工具，用于平台本身的运行和提高用户体验。对于部分通用操作，例如数据库操作等，可以供应用插件调用使用。

插件服务层主要对部分 EDS 常见的数据文件格式开展数据生成、校验和解析，这样，每个工具无需单独对与业务无关的外部操作过分关注，只需要调用相关通用插件服务即可。同时，插件服务层提供应用业务自定义数据库的统一操作接口，也可以供插件服务层直接调用。

通用服务层和插件服务层的通用插件或服务基本列表如表 2。

表 2 插件或服务名称列表

序号	所属层次	插件或服务名称	插件或服务输入	插件或服务输出
①	插件服务层	XML 标准文件生成工具	节点、值、节点位置	XML 文件流
②		XSD 标准文件生成工具	节点、节点位置、节点数据类型和约束	XSD 文件流
③		INI 标准文件生成工具	标签、值	INI 文件流
④		XML 标准文件校验工具	XML 文件流	文件格式校验结果
⑤		XSD 标准文件校验工具	XSD 文件流	文件格式校验结果
⑥		INI 标准文件校验工具	INI 文件流	文件格式校验结果
⑦		XML 标准文件解析工具	XML 文件流	节点、值、节点位置
⑧		XSD 标准文件解析工具	XSD 文件流	节点、节点位置、节点数据类型和约束
⑨		INI 标准文件解析工具	INI 文件流	标签、值
⑩		自定义数据库管理工具	数据库操作符、相关数据	操作结果、查询结果
⑪	通用服务层	文件流输出工具	文件数据	文件
⑫		插件数据库操作工具	数据库操作符、相关数据	操作结果、查询结果

⑬		XML 编辑器管理工具	文件流	文件解析结果
⑭		XSD 编辑器管理工具	文件流	文件解析结果

对于应用业务层的即插即用工具，更多是对相关工具链具体工具的业务实现。同一个工具链中的工具可以连续使用或者具有相关关系，工具和工具之间也均可以独立运行或独立在插件架构内运行。因此，不仅需要关注每个工具的服务调用情况，也需要结合实际适用场景确定工具及其输入输出。

对每一个工具，工具本身会依据其输入和输出，调用通用插件实现对输入数据的预处理和输出结果的处理，从而得到该工具的最终输出结果。

前期，工具链中的主要应用业务包括了基于 XTCE 的遥控工具链、基于 XTCE 的遥测工具链、参数定义工具链和 SEDS 工具链。未来，工具链和工具可以支持横向及纵向扩展，增加相应的其他工具和工具链。

遥控工具链适用于数据上注阶段指令、指令包和注入包的生成，遥测工具链实现了地面端对非既定遥测格式信息的自动识别及标准化映射，进而支持卫星遥测数据的按需组织下传及接收识别。

目前，应用插件的输入输出及其依赖对照表如表 3。

表 3 工具链、工具等应用插件输入输出和依赖对照表

序号	工具链	工具名称	工具输入	工具输出	调用通用服务层的服务	调用插件服务层的插件
1	XTCE 遥控 工具链	创建指令模板	模板名称、参数名称、参数格式、参数数量	指令模板数据	⑪⑫⑭	②⑤⑧⑩
2		生成指令包实例	指令模板数据、参数数量、参数值	指令包数据	⑪⑫⑬	①④⑦⑩
3		生成注入包包头	包类型、包长度、设备访问 ID、包序号、其余包数据	注入包包头	⑬	①④⑦
4		生成注入包	指令包数据、注入包包头	注入包数据	⑪⑫⑬	①④⑦⑩
5	XTCE 遥测 工具链	创建遥测模板	参数类型、参数、简单整合结构，复杂整合结构	遥测数据模板	⑪⑫⑭	②⑤⑧⑩
6		创建遥测包模板	主导头、副导头	遥测包模板	⑪⑫⑬	②⑤⑧
7		生成遥测实例	原始遥测数据，遥测模板	序列化遥测数据	⑪⑫⑬	①④⑦⑩
8	参数 定义 工具链	类型定义工具	包域类型代码、包域格式代码	XTCE 类型模板数据	⑪⑫⑭	②⑤⑧
9		参数定义工具	参数名、参数解释、参数类型	XTCE 参数信息模板数据	⑪⑫⑭	②⑤⑧
10		简单整合结构定义	简单整合结构名、已定义参数	简单整合结构模板数据	⑪⑫⑬	①④⑦⑩
11		复杂整合结构定义	复杂整合结构名、已定义简单整合结构	XTCE 复杂整合结构模板数据	⑪⑫⑬	①④⑦⑩
12	SEDS 工具链	创建设备模板	模板名称、参数名称、类型、引用、格式、描述、排列、接口、组件	SEDS 模板数据	⑪⑫⑭	②⑤⑧⑩
13		创建设备包模板	包类型、包长度、设备访问 ID、包序号、其余包数据	设备包数据	⑪⑫⑬	②⑤⑧
14		生成设备实例	实例名称、ID 引用、描述、接口、组件	SEDS 设备实例数据	⑪⑫⑬	①④⑦⑩
15		生成设备个性化文件	SEDS 设备实例数据	SEDS 设备个性化数据	⑪⑫	③⑥⑨⑩
……	……	……	……	……	……	……

4 系统综合实现与验证

4.1 系统开发技术

目前，基于插件化架构的相关技术规范有很多，目前较为常见的有 Java 的开放服务网关协议(Open Service Gateway Initiative,OSGi)技术规范。开放服务网关协议是基于 Java 语言的软件动态模块化技术，相关的技术规范 OSGi 联盟共同制定。采用 OSGi 技术实现的软件是由一定数量的被称作 Bundle 的代码单元构成，OSGi 技术框架主要提供了插件的执行环境、生命周期管理和依赖关系管理等功能，同时提供了如事件管理、日志管理服务等等，用于满足多样开发场景的需要[18]。

本研究以前文所述架构作为软件框架运行环境，基于 Eclipse 富客户端平台(Eclipse Rich Client Platform,RCP)技术，完成整个工具链管理系统的开发与实现。 Eclipse RCP 是基于 Eclipse 平台技术的一个富客户端程序开发平台，该技术将多余模块从传统的 Eclipse 平台中剥离，仅保留了作为核心的运行时框架和 UI 框架两大部分，搭建了一个通用的程序开发平台[19]。

在实现方面，本系统在宏观层面采用独立的富客户端架构实现，将视图与本地数据存储链接。本地关系型数据库采用 SQLite 实现。 SQLite 是一个进程内的轻量级嵌入式数据库，它的数据库就是一个文件，实现了自给自足、无服务器、零配置的、事务性的 SQL 数据库引擎。该数据库能有效满足富客户端的数据存储需求。同时，平台基于多个数据库实现数据库分布式扩展，有助于每个应用插件独立管理其数据[20]。

对于插件的实现，每个插件基于一个 Java 存档（Java Archive,JAR）单独实现，插件本身需基于工厂设计模式，并按照接口规范实现接口后，即可加入到平台内。插件本身可以单独构建可视化窗口，也可单独运行或者暴露接口，实现插件的独立性和整个系统的可扩展。

4.2 功能结构设计

电子数据单工具链管理系统是以插件化进行构造，这就要求整个管理系统的功能模块也应当以插件为依据划分。管理系统的功能结构图如图 5 所示。整个工具链管理系统分为平台扩展服务、插件服务和平台工具服务三部分，与整体架构横向对应。平台基本服务包括用户管理、文件日志管理、标准支持等，保证整个软件的基本运行。插件服务用于实现插件的加载与管理。平台工具服务指可扩展框架，包括了通用插件和应用插件，即各个工具链。其中，整个软件的功能可以以通用插件或者工具链为单位扩展。

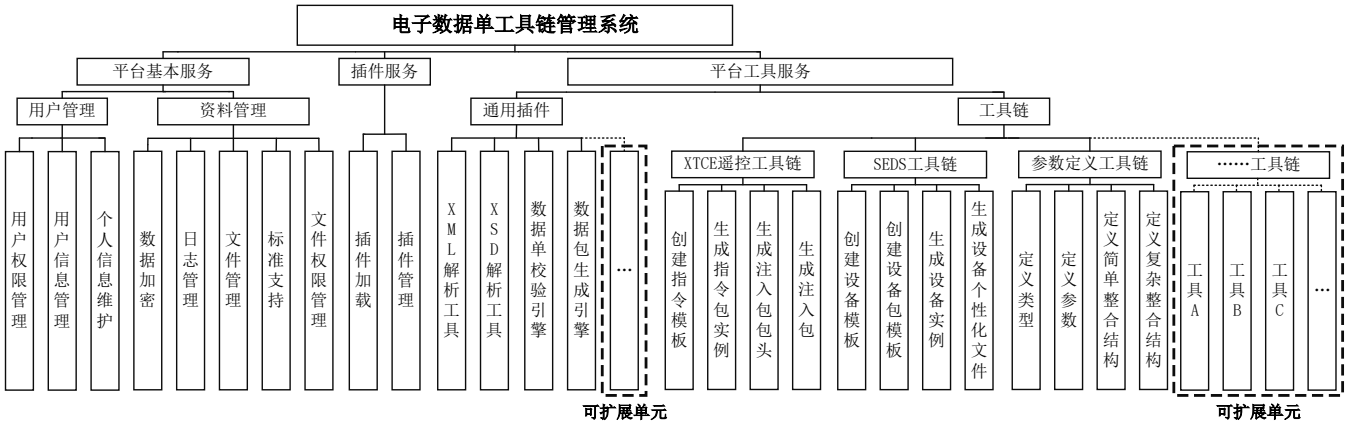

图 6 工具链管理系统功能结构图

4.3 界面渲染引擎设计方案

传统和常见的工具软件，软件界面使用固定模式开发，开发完成之后即不再修改。这种界面开发方式若应用于插件平台中，会造成开发标准不统一、难以重用、界面视觉效果不统一、不利于插件开发本身专注于业务等问题。基于此，本研究针对管理系统，设计了动态视图渲染方案。该方案通过“文件定义界面”的方式，以文件内容作为界面的生成依据，使开发者专注于业务本身，脱离繁杂界面的开发工作。

整个具体的渲染过程如图 6 所示。

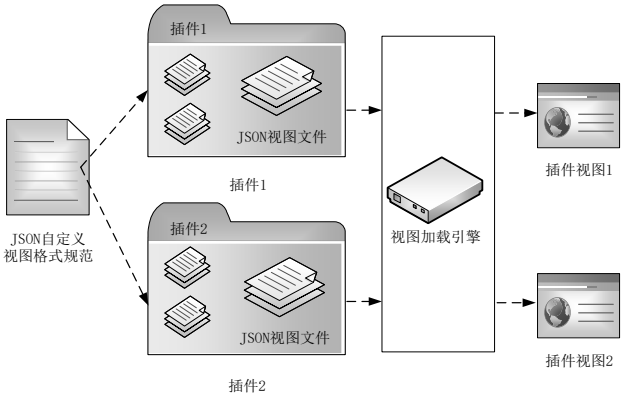

图 7 视图渲染引擎功能示意图

整个渲染过程大致如下：首先对于每一个插件，均需要包含一个基于 JavaScript 对象表示法(JavaScript Object Notation,JSON)格式的视图文件。该文件的组织以事先约定好的 JSON 自定义视图格式规范为依据。其次当插件加入管理系统时，管理系统中的视图加载引擎读取视图文件。最后，视图加载引擎将视图文件基于格式规范，渲染成为界面视图。

对于插件而言，每一个插件的视图是由若干个控件组构成，每一个控件组又由若干控件构成。JSON 文件通过分层形式，对控件组和控件进行基于数组的定义，最小单位为每一个控件。因此，在实际的 JSON 格式规范中的规则描述里，主要是对相关的控件键值进行描述。基于实际的用户需求，控件规则描述与匹配键值表如表 4。

对于控件取值，为了增加描述的方法，对其进行了更加详细的描述，匹配键值表如表 5。

表 4 控件规则描述与匹配键值表

编号	字段名称	关键字	值类型	必填	说明
1	控件名称	Name	String	是	控件的名称。
2	控件标识	Label	String	是	在渲染器显示的标签名称。
3	控件描述	Description	String	否	在渲染器显示的标签描述。
4	控件类型	Type	String	是	在渲染器显示的控件类型（如下拉菜单、单选等）。
5	控件内容	Content	String	否	控件默认数据。
6	控件取值	Field	Map	否	控件值的取值限制。
7	控件输入数据类型	Datatype	String	否	控件输入内容的数据格式（Int，Double）。

表 5 控件取值规则描述与匹配键值表

编号	字段名称	关键字	值类型	必填	说明
1	最大值	MaxValue	Double	否	控件输入值的最大值。
2	最小值	MinValue	Double	否	控件输入值的最大值。
3	小数位数限制	DoubleLength	Int	否	控件输入值的小数最大位数。
4	字符长度限制	StringLength	Int	否	控件输入值的字符最大长度。

基于表 5 和表 4 相关规则，可以构建标准 JSON 文件，之后，对此文件开展解析，即可获得相应视图。

4.4 系统实现与应用

基于上述架构和技术实现的界面示意图如图 6 所示。

整个系统主界面可以分为五大部分。最上面为菜单栏，用于基本操作和工具快捷方式。左侧为数据库系统，用于插件的管理、选择、数据的链接等。下侧为日志系统，用于集中打印所有的日志信息。右侧为插件操作台，视图加载引擎就在这里开展加载工作。中间内容为编辑器，对相关的电子数据单亦可以直接编辑修改相关内容。

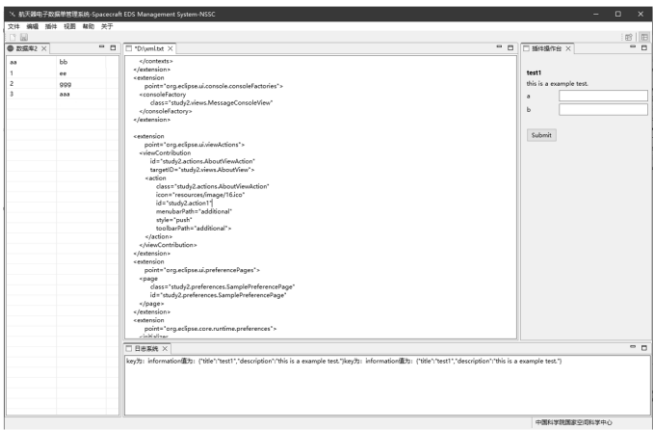

图 8 系统主界面示意图

本系统的架构及其技术实现的管理系统，通常用于对常见的工具链进行替代。将依照本架构实现的软件与其他工具链使用和管理方法进行对比，对比结果如表 6。

表 6 工具链使用方式对比

工具链应用方法	简介	EDS 创建方法	工具组织管理方法	数据管理方法	工具增删灵活性	工具开发便捷性和最小开发单位	视图灵活性
人工数据输入	人工创建编辑和管理 EDS	手工	手工	手工	——	差	差
传统工具链软件	以工具为单位的小型软件	利用工具	无	手工	强	较差，整个工具链	差
非插件化架构工具管理系统	将工具集成的综合性软件	利用工具	无	可借助数据库管理	很弱	差，整个软件	差
传统插件化架构管理系统	依照传统架构的工具集成软件	利用工具	核心组件-插件	可借助数据库管理	强	较优，整个插件	差
本文改进后的插件化架构管理系统	依照本文改进架构的工具集成软件	利用工具	核心组件-通用插件-应用插件	可借助数据库管理	强	优，接口实现	优

由于工具链管理系统面向的使用对象是 EDS 的编写开发人员，因此，在 EDS 创建和工具的管理过程中，用户的应用就较为重要。通过对本方法和人工方法、传统工具与工具链、非插件化的工具管理系统、传统架构系统的对比，可以发现，本架构的系统在可以实现基本工具的相应功能，同时能够基于插件特性，满足运行的独立性和可扩展性。同时，与非插件化架构和传统插件架构对比，本架构能够增加工具管理的灵活性，同时简化工具的开发流程和方式，有利于开发人员进一步自定义和扩展。

5 结 语

本文讨论了电子数据单的应用场景，通过分析电子数据单工具链的功能需求与独立、可扩展的非功能需求，提出了基于插件化架构的电子数据单管理系统。该设计方案重点解决了系统的独立性和可扩展性，将整个系统层次分为六层，同时，本文提出了基于 Java OSGi 技术实现工具平台的方法，并应用到电子数据单的实际应用场景当中。未来，本研究相关的工具链会进一步扩展，进一步丰富工具箱的可靠性、易变性和实用性。

参 考 文 献

[1] 吕良庆. 航天器智能软件体系架构设计与应用研究[D]. 中国科学院大学(中国科学院国家空间科学中心),2018.

[2] 吕良庆,黄永辉,安军社.基于CCSDS-SOIS的航天器数据管理系统体系架构的探讨[J].南京大学学报(自然科学),2018,54(03):506-514.DOI:10.13232/j.cnki.jnju.2018.03.003.

[3] 周勇吉,吕良庆,白云飞.空间数据系统电子数据单工具链设计[J].航天器工程,2021,30(04):91-98.

[4] 吕良庆,张峻巍,何睿,王彧泽.航天器电子数据单应用方法分析[J].航天器工程,2022,31(02):126-131.

[5] 黄秀丽,陈志.基于JSON的异构Web平台的设计与实现[J].计算机技术与发展,2021,31(03):120-125.

[6] 何熊文,朱剑冰,程博文,顾明,阎冬.星载标准接口业务在航天器中的应用方法[J].航天器工程,2015,24(06):52-58.

[7] 李宏亮,贾茹,张悦,潘顺良.一种航天器测试注入数据序列自动生成方法[J].航天器工程,2020,29(01):93-99.

[8] 吕良庆.航天器软件的研发路线[J].国防科技大学学报,2021,43(02):61-65.

[9] 何熊文,徐明伟.航天器接口业务标准化和软件架构现状与发展展望[J].中国航天,2020(09):29-35.

[10] 刘洋,李宗德,丁雪静,戴媛媛,何晓苑.基于XTCE的卫星遥测数据处理方法[J].遥测遥控,2017,38(02):27-31.DOI:10.13435/j.cnki.ttc.002833.

[11] 张煦冬,吕良庆,安军社.基于SOIS EDS的实例化设计[J].计算机工程与设

计 ,2020,41(09):2670-2677.DOI:10.16208/j.issn1000-7024.2020.09.040.

[12] 李姗,骆培,安军社.航天器综合电子系统通用测试系统设计[J].哈尔滨工业大学学报,2014,46(09):92-99.

[13] 赵才文.插件化软件开发方法初探[J].电脑与电信 ,2017(11):76-79.DOI:10.15966/j.cnki.dnydx.2017.11.022.

[14] 曲明成,崔乃刚,吴翔虎,陶永超.嵌入式软件虚拟化测试技术标准框架研究[J].哈尔滨工业大学学报,2017,49(05):49-55+121.

[15] 李天琦. 基于业务插件化的电商大数据采集系统[D].浙江工业大学,2019.

[16] 余晟,向永清,宋宏江.基于插件的航天器测试总控软件系统设计[J].航天器工程,2015,24(05):119-125.

[17] 马晨溪,李彦平.软件工厂模式的软件产品快速构建技术[J].计算机与网络,2021,47(15):61-65.

[18] 宋文婷,赵建新,艾冰,高腾飞.基于OSGi的军用指挥软件插件机制研究[J].火力与指挥控制,2019,44(05):172-176.

[19] 李艳艳,徐京.SVG及Eclipse RCP技术在地面站设备监控软件中的应用[J].航天器工程,2010,19(04):74-79.

[20] 李国伟,寇娟,王录锋.实验室间协同试验数据处理软件设计[J].计算机应用与软件,2022,39(02):11-15+54.

提供联系方式

何睿（1999-），女，硕士研究生，主要从事空间数据系统，软件工程等方面的研究。
身份证号：41010119990622502X
单位：中国科学院国家空间科学中心
通信地址：北京市怀柔区京密北二条 1 号国家空间科学中心
手机号：15638147863
E-mail: 1050637025@qq.com

张峻巍（1998-），男，博士研究生，主要从事空间数据系统，软件工程等方面的研究。
单位：中国科学院国家空间科学中心
通信地址：北京市怀柔区京密北二条 1 号国家空间科学中心
手机号：18845128716
E-mail: 1174681078@qq.com

卢广佑（1998-），男，硕士研究生，主要从事空间数据系统，软件工程等方面的研究。
单位：中国科学院国家空间科学中心
通信地址：北京市怀柔区京密北二条 1 号国家空间科学中心
手机号：13667830733
E-mail: 642322360@qq.com

王彧泽（1999-），女，硕士研究生，主要从事空间数据系统，软件工程等方面的研究。
单位：中国科学院国家空间科学中心
通信地址：北京市怀柔区京密北二条 1 号国家空间科学中心
手机号：18518071911
E-mail: zzz1565789907@163.com

吕良庆（1969-），男，博士，研究员，主要从事空间数据系统与体系架构，航天器软件工程，航天器智能能力构建等方面的研究。
单位：中国科学院国家空间科学中心
通信地址：北京市 8701 信箱（100190）
E-mail: lvliangqing@nssc.ac.cn

【初审修改说明】

（1）文中插图不清楚,请直接作图或使用 Visio 作图，使文字可编辑，而不要转换成 JPG 格式的图片贴上去。同时，请去除图的底色、底纹或填充色；其中：图 2-6 请修改；

插图已更改为 vsdx 文件，双击可编辑。图 2-6 已去除底色。

（2）本刊为黑白印刷，请将文中所有的图、表由彩色改成黑白灰度，若文中有对图、表颜色的描述，请作相应修改；

已调整彩色图片为黑白灰度。

（3）摘要过于冗长，请修改中英文摘要；

已修改中英文摘要，其中中文摘要字数小于 300 字。

（4）本刊为双栏格式排版，建议图/表在保持清晰的前提下改成一栏的宽度，如必须跨栏，请在文末说明；

已将图 5、7、8 修改为一栏宽度。图 1、2、3、4、6 由于内容较多，需要跨栏排版。

（5）近期本刊来稿较多，发表周期较长，请谨慎投稿，如对发表周期有要求，请另投他刊；

已知悉，期待回执。

（6）请重新按序编排文中图、表序号（图 1、图 2…；表 1、表 2…）；

已重新按序编排文中图、表序号。

【初审结果】请按要求对稿件修改补充后进入本刊网站“在线投稿”-“修改稿上传”，通过输入第一作者姓名和稿件编号提交“修改稿”（请在论文结尾附上【初审修改说明】），并等待再次初审（超过一月未收到修改稿，将作退稿处理）（稿件状态=等待预审）。

《计算机应用与软件》编辑部 2023-1-19 10:44:48。

已在论文结尾附上【初审修改说明】