# Research of Data Processing Difference Component and Automatic Deployment Package

Xiaoyang Liu[1, a], Jianping Zhao[1, b], Junlei Bao[1, c], XiaoChun Liu [1, d], Liya Xu [1, e]

[1]Chinese Satellite Marine Tracking and Control Department, Jiangyin, 214431, China

[a]email:liuxiaoyang120@163.com, [b]email:niuniu-jiangying@163.com, [c]email:suray@126.com, [d]email:lxc439.student@sina.com, [e]email:xlych2003@163.com

**Keywords:** Automated Deployment; Data Process; Framework Design; Difference Component

**Abstract.** It put forward the embed data process technology based on difference rule and Frames mould process environment construct technology, constructed command display data difference process component. It accomplished the process and display for disparity data form information for the experiment information surveillance display software to construct the data frames process environment for rocket and satellite by way of Frames mould, compile and execute difference rule collection in real time and dynamically with the configuration of difference rule collection between the general and special trajectory and remote measure data frames for rocket and satellite, the customization expectation process consequence Frames mould for remote measure and trajectory data information of rocket and satellite. At the same time, it increased the efficiency of command display data process maintenance through the application of customization technology based on symbol and the automatic deployment of component.

## Introduction

It designed and actualized data difference process component automatic deploy subsystem. For multi-satellite type task, as the dissimilarity of satellite type, the remote measure data and trajectory data form of rocket and satellite transmitted from information center to headquarters would generate variation. For BeiDou-2 engineering task with its high emission frequency, multi-satellite type and multi-type orbit, how to inherit generality, share preparation achievement and validly increase the task preparation efficiency is an urgent problem to accomplish the task. It put forward the embed data process technology based on difference rule and Frames mould process environment construct technology, constructed command display data difference process component. It accomplished the process and display for disparity data form information for the experiment information surveillance display software to construct the data frames process environment for rocket and satellite by way of Frames mould, compile and execute difference rule collection in real time and dynamically with the configuration of difference rule collection between the general and special trajectory and remote measure data frames for rocket and satellite, the customization expectation process consequence Frames mould for remote measure and trajectory data information of rocket and satellite. At the same time, it increased the efficiency of command display data process maintenance through the application of customization technology based on symbol, the transparent construction of deploy package and the automatic deployment of component[1].

## Process Display Mechanism for Experiment Information Surveillance Display Software

In Experiment Information Mornitoring Display Software, the data package of Package Data eXchange Protocol dispatched from the center was explained firstly for the field by data process software according to the form information configured in database which changed the quantitative data into semanteme data, then it was sent to display software in the light of the interface agreed before. After the data was received by the display software it triggered each display component which binds the data to accomplish display. For example, it refurbished the text of textbox or drew one point in curve， which is shown in figure 1.

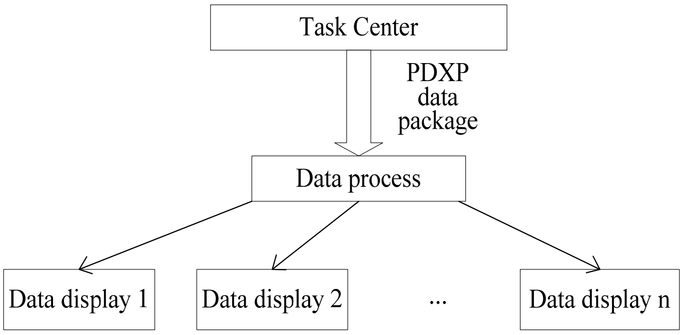


Fig.1. Centralized process and customized display mechanism

From the function, the data process software conducted process, while the data display software conducted UI user display. But the process method for data process software was configured in database corresponding with certain task symbol, and in multi-goal task the calculate parameter of data form for different task symbol is different while data process software has no better means to process. Next, base on the definition of join test outline it exists requirement of advanced semanteme judgement for the data calculated by certain data process software, for instance, to get rid of invalid value or to compute for the second time. These problems were all much troublesome toward data process software. In the preparation course of previous task, on condition of encountering similar problem, somewhat would need to deploy more data process computers to process each different task symbol data, somewhat would need to conduct code maintenance for the data process software, also somewhat were unwieldiness as a result of the restriction for present code framework.

As a matter of fact, regardless of data process software or data display software, as a whole, EIMDS's function was to accomplish the calculation and display for data. The question was nothing but how to discriminate the two terms of function for calculate and display more soundly. Therefore, it could add a data process component in data display to partake a portion data process work, especially for data that could not be processed inconveniently by the process software. Compared to data process software, the data process component of display end should have three characteristics:

Firstly, construction light quantization, as its actualization expense was less and the function was more concentrate.

Secondly, it must be inner embedded in data display software, nonuse computer or port singlely, moreover it could be configured to use or nonuse basis task requirement.

Thirdly, the process module should be used convenience, which had good flexibility, and was capable of fit the requirement for specific data calculation, and didn't need to alter maintenance frequently on account of the data form variation.

## Rule Process Technology for Microsoft WorkFlow

According to the former program exploit experience, the similar problem encountered in task preparation also existed in MicroSoft WF actualization technology, and WF gave a suit of unique solve clue--rule. Therefore, first of all it studied intensively dynamic rule technology of MicroSoft WF.

**Microsoft WF summary.** The full name of WF was Workflow. MicroSoft integrated the WF function actualization in NetFramework 4.5. Other than many previous else work flow actualization technology, MicroSoft WF completely based on Object-Oriented model design exploit. No matter it was work definition, event transfer, flow control, condition preservation or routine administration, that whole was pure Object-Oriented with its extraordinary mightiness description ability. It also offered WF visualization designer in Microsoft Visual Studio 2015 integration exploit software while user could accomplish work flow design with dragging component and setting up attribute[2]. It was extraordinary convenience and high effect.

**The characteristic of WF's PolicyActivity.** Microsoft WF work flow falled into two categories, Sequential and State, which corresponded order model and condition machine model as their minimal compose unit was activity, for instance the Activity of order model was orderly executed.

In such Activity, there was a sort of PolicyActivity whose execution was very peculiar. Its execution result was determined by Policy, namely user configure strategy[3]. By the way of configuring different strategy, i.e., rule collection, user could agilely adjust and control the flow and outcome of work flow including PolicyActivity.

**RuleEngine model.** PolicyActivity could actualize its agile control function while its foundation was RuleEngine. It was one core component of Microsoft WF. And some other important flow control also all depent on RuleEngine. Although RuleEngine had powerful function and important action, it was independent in entire WF design framework[4]. So it became an extraordinary favorable condition to desquamate RuleEngine from WF, and also became one key point for entire research work.

## Design of Subsystem Framework

**Chief peculiarity.** The chief peculiarity of embedded-data process technology based on difference rule included three aspects:

Firstly, it had intact process ability. Not only it could conduct quadratic process for the consequence processed by data process software, but also according to information exchange prescript it could immediately calculate original code data sent from center.

Secondly, embed operation pattern. By the form of PnP plug-in components it operated in display program while it was the actualization of expand function for data display program. It didn't need to occupy single task computer and port. If certain task had no special form data and the data process software could whole process, then user might not use the plug-in components.

Thirdly, configure agile. By means of rule form, i.e., character string text, the process method was configured in display attribute as a part of display data package to advocate leading-in, leading-out and automatic down load and update.

**Deploy pattern.** As a selectable plug-in components, the embed data process component based on difference rule completely followed the distributing deploy method of present display software. Under the situation embed data process naturally advocated customization process[5]. On the one hand, different display computer might get different type consequence for the same data according to different process pattern; on the other hand, different display computer might seperately process different data to in effect reduce load and increase process efficiency, as shown in figure 2.

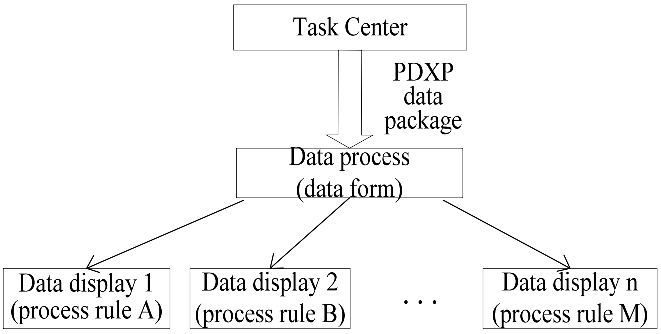
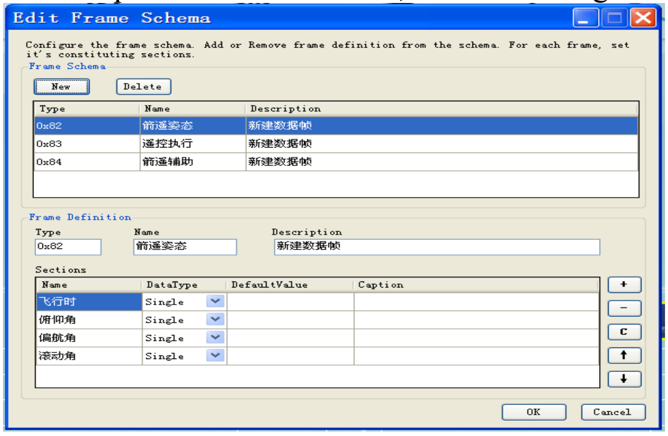

Task Center
PDXP data package
Data process (data form)
Data display 1 (process rule A)
Data display 2 (process rule B)
. . .
Data display n (process rule M)

Fig.2. The construction of customization process and customization display

**Process engine.** The designed embed data process engine DataProcessEngine included three parts: RuleEngine, RuleSet and ProcessToolKit. RuleEngine stemed from MicroSoft WF RuleEngine,which accomplished dynamic parsed and executed for rule collection; RuleSet was character string text configured by user described data process course; process ToolKit constructed data process environment and offered common process method such as transition and data temporary deposit. These methods constituted basic compose element for rule.

**Framework designer.** As process engine was embeded in data display software, its process consequence should adapt present EIMDS framework peculiarity, and preserve conformity of data interface for present data display software and data process software. Only in this way process consequence of process engine could directly apply to binded display of each component for display software. Therefore, it designed a framework design utensil to offer a User Interface which was used to design the consequence form definition, i.e., interior consequence, processed from process engine[6]. User only needed simply knock several buttons such as New, Accretion and Deletion, and could conveniently define the process resultant form, as shown in figure 3.


Edit Frame Schema

Configure the frame schema. Add or Remove frame definition from the schema. For each frame, set it's constituting sections.

Frame Schema

| Type | Name | Description |
|---|---|---|
| 0x82 | 箭遥姿态 | 新建数据帧 |
| 0x83 | 遥控执行 | 新建数据帧 |
| 0x84 | 箭遥辅助 | 新建数据帧 |

Frame Definition

Type: 0x82
Name: 箭遥姿态
Description: 新建数据帧

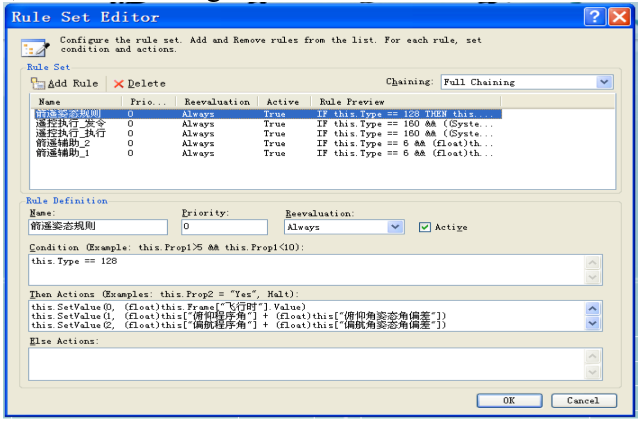
Sections

| Name | DataType | DefaultValue | Caption |
|---|---|---|---|
| 飞行时 | Single | | |
| 俯仰角 | Single | | |
| 偏航角 | Single | | |
| 滚动角 | Single | | |

Fig.3. User surface for frame schema designer

To the process consequence information designed by user, framework design utensil would automatic preserve it as xml form document. It was completely conformance with interface form for data process software and data display software, and every data consequence was all correspondence with a sort of Frame.

**Rule designer.** The chief function of rule design utensil was to offer a convenient UI surface for rule text input, edit and inspection. With it user could input concrete process logic needed finish including transfer all sorts of process method for process tool, accomplish field parse, dimension computation, data type transition, field operation, invalid value inspection and so on. Toward the process rule input by user, rule design utensil finally preserved it as xml document in term of normal rule collection form, as shown in figure 4.


Rule Set Editor

Configure the rule set. Add and Remove rules from the list. For each rule, set condition and actions.

Rule Set

Chaining: Full Chaining

| Name | Prio... | Reevaluation | Active | Rule Preview |
|---|---|---|---|---|
| 箭遥姿态规则 | 0 | Always | True | IF this.Type == 128 THEN this.... |
| 遥控执行_发令 | 0 | Always | True | IF this.Type == 160 && ((Syste... |
| 遥控执行_执行 | 0 | Always | True | IF this.Type == 160 && ((Syste... |
| 箭遥辅助_2 | 0 | Always | True | IF this.Type == 6 && (float)th... |
| 箭遥辅助_1 | 0 | Always | True | IF this.Type == 6 && (float)th... |

Rule Definition

Name: 箭遥姿态规则
Priority: 0
Reevaluation: Always
☑ Active

Condition (Example: this.Prop1>5 && this.Prop1<10):

```
this.Type == 128
```

Then Actions (Examples: this.Prop2 = "Yes", Halt):

```
this.SetValue(0, (float)this.Frame["飞行时"].Value)
this.SetValue(1, (float)this["俯仰程序角"] + (float)this["俯仰角姿态角偏差"])
this.SetValue(2, (float)this["偏航程序角"] + (float)this["偏航角姿态角偏差"])
```

Else Actions:

Fig.4. User surface for rule designer

**Process flow.** In initial course, process engine would first inspect two documents: framework document and rule document .The framework document defined what the process result was, and the rule document defined how to get it. If user didn't configure rule and process framework, it wouldn't generate the two documents and difference data process flow wouldn't start-up.

After successfully loaded framework document and rule document, process engine would enter the status of cycle wait. When the module of data display network accepting received result frames which were send out by Data Process Software, it immediately constructed process environment in use of result frames, then invoke Rule Engine to execute rule, at last returned with extract results from process environment, as shown in figure 5.

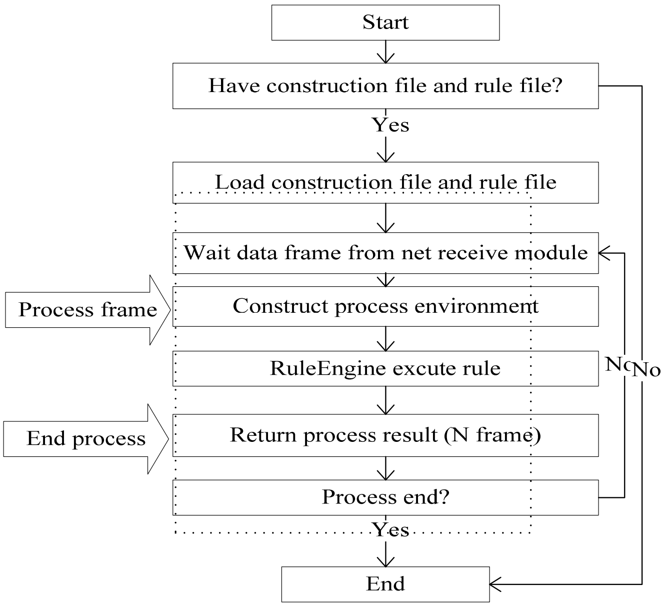

Start
Have construction file and rule file?
Yes
Load construction file and rule file
Wait data frame from net receive module
Process frame
Construct process environment
RuleEngine excute rule
No
No
End process
Return process result (N frame)
Process end?
Yes
End

Fig.5. Rule process flow

## Conclusion

According above framework design, on the foundation of present EIMDS, the difference data process module based on rules was added in EIMDS, and the difference data component was developed. A user only needed to configure rule text, then he would actualize the customization processing in the display end to the data that data process software not convenient for process, get requisite consequence frame, and then directly bind it to each kind of display component to accomplish display.

Through the application and validate for dozens of tasks, it certificated embed that data process technology based on difference rule was completely feasible. It was extraordinary in effect for the data with disparity process requirement in task. Especially to Beidou engineering for the multi-satellite type, multi-orbit and emission frequency, it was high effect and credibility for EIMDS data process software maintenance. Similarly, construct difference data process component technology also suited other tasks.

## References

[1] Chen Zhuo. Component-based Formal Development for Reuse [D]. Henan University of Science and Technology, 2010.

[2] Jiang Haile. Layered Component Based Embedded Real-time Software Framework Design [D]. Shanghai Jiao Tong University, 2013.

[3] Zhang Zhifei. Data Object Access [D]. FUDAN University. 2013.

[4] Tian Jin, Zhan guangxin, Hou Dibo, Zhou Zekui. Development of Water Quality Early-warning Information System Based on Dynamic Compilation Technology [J]. Information of China Construction. 2010:11 (8):1-4.

[5] Feng Xinyang , Fan Ying , Shen Jianjing. Constructing Invocation Model of Web Services with Service Locator Pattern [J]. Computer & Digital Engineering. 2009: 9 (239):32-34.

[6] Liu Yuxin, Yao Kaixue, Xu Daoyun. Automatic Code Generation Based on Template on.Net Framework of Three Layers Architecture [J]. Computer Technology and Development. 2012: 8 (22):13-16.