

THE DESIGN OF TANGO BASED CENTRALIZED MANAGEMENT PLATFORM FOR SOFTWARE DEVICES

Z. G. Ni[†], J. Luo, J. Liu, X. W. Zhou, Institute of Computer Application, China Academy of Engineering Physics, Mianyang City, China

Abstract

Tango provides the Tango device server object model(TDSOM[1]), whose basic idea is to treat each device as an object. The TDSOM can be divided into 4 basic elements, including the device, the server, the database, the application programmers interface. On the basis of the TDSOM, we design a centralized platform for software device management, named VisualDM, providing standard servers and client management software. Thus the functionality of VisualDM are multi-folds: 1) dynamically defining or configuring the composition of a device container at run-time; 2) visualization of remote device management based on system scheduling model; 3) remote deployment and update of software devices; 4) registering, logging, starting and stopping devices. In this paper, platform compositions, module functionalities, the design concepts are discussed. The platform is applied in computer integrated control systems of SG facilities.

INTRODUCTION

The large-scale scientific experimental facility consists of a number of systems, units and components, For exam-

ple: The National Ignition Facility (NIF)'s ICCS[2][3] is a layered architecture of 300 front-end processors attached to nearly 60,000 control points and coordinated by supervisor subsystems in the main control room. Each component is basically a control unit with independent control algorithms, such as valves, motors, CCD, etc. These associated control units are combined to a sub-system. These control objects in the software embodied as one of the software device, the relationship between software devices includes combinations and dependencies. After a large number of cases, the controlled device management becomes a problem, how to effectively deploy, monitor, upgrade and visualize a huge number of devices is a topic worthy of study. Centralized Management Platform named VisualDM is designed to address this issue.

DEVICE MONITOR BASED ON JIVE OR ASTOR

Jive[4] is a standalone JAVA application designed to browse and edit the static TANGO database. Jive can manage and create devices, properties and classes in Fig. 1. Jive also offers advanced search/selection features.

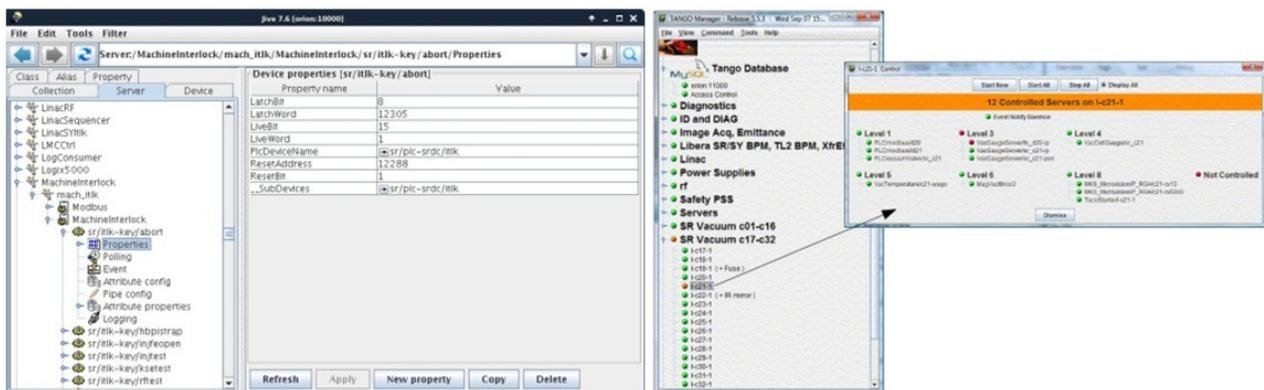


Figure 1: Jive and Astor running interface.

Astor[5] is a Java program using Swing classes. Astor displays a tree where nodes could be a family of hosts, and leaves are hosts where a Starter (a program belonging to tango) device server is registered in database, as provided in Fig. 1.

These two programs have a common feature: the managed device is equal, is flat, determined by the tango framework. This flat management of devices is suitable for the debugging phase, not suitable for the running phase. During the running phase the operator is facing the view of system objects, is a hierarchical view.

[†] email address: drops.ni@caep.cn

CONTROL SYSTEM SOFTWARE STRUCTURE

A human-built system with complex behaviour is often organized as a hierarchy[6]. For example, in Fig. 2 a command hierarchy has among its notable features the organizational chart of superiors, subordinates, and lines of organizational communication. Hierarchical control systems are organized similarly to divide the decision making responsibility. Each element of the hierarchy is a linked node in the tree. Commands, tasks and goals to be achieved flow down the tree from superior nodes to sub-

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ordinate nodes, whereas sensations and command results flow up the tree from subordinate to superior nodes. Nodes may also exchange messages with their siblings.

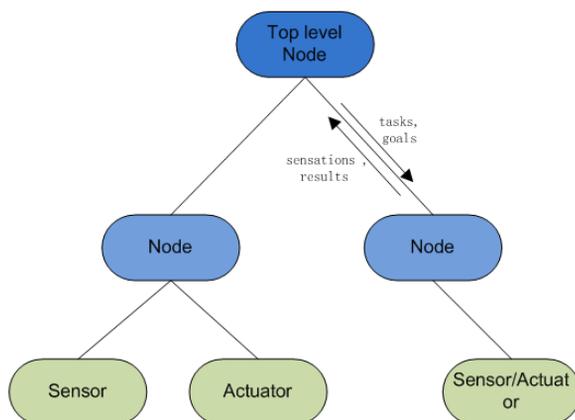


Figure 2: Control system software structure.

The Integrated Computer Control System of inertial confinement laser facility is a hierarchical control system. The control system is divided into three layers: integrated control layer, system service layer, device service layer [7][8], as provided in Fig. 2. The device service layer is responsible for the control integration of devices with independent logic. The system service layer is responsible for the coordination of all related devices within the system level. The integrated control layer is responsible for organizing the scheduling relationship between the systems to achieve system-wide sequential control scheduling.

IMPLEMENTATION

VisualDM implements soft device lifecycle management including software device deployment, upgrade, registration/logout, start/stop, status monitoring, fault alarm, etc. The platform consists of three programs: DLoader, Starter++, VisualDM in Fig. 3, where Starter++ is a collection of DLoader and libStarter.so.

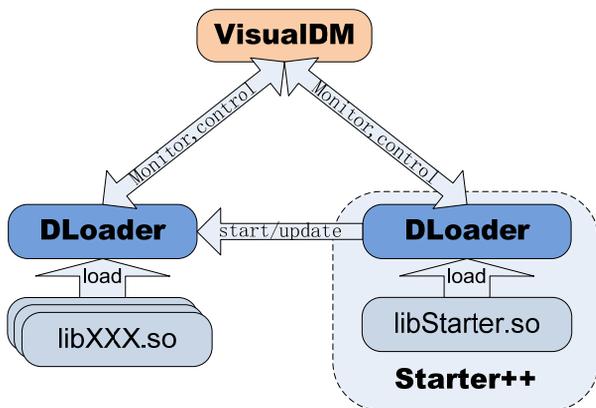
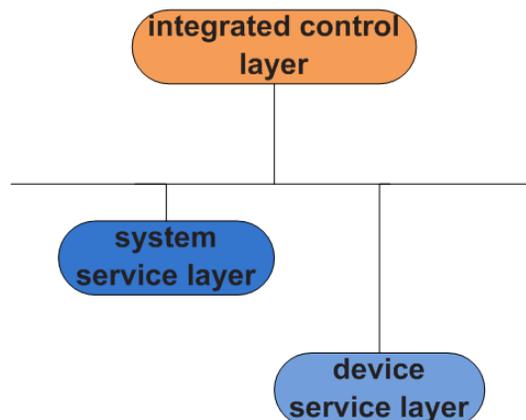


Figure 3: VisualDM platform composition diagram.



DLOADER

DLoader is a server program, similar to the ordinary tango server. The difference is that device instance is dynamically loaded by DLoader, depending on the registration information as provided in Fig. 4. As the registration information changes, the DLoader loads a different device instance when it runs again. Each device class generates a dynamic library that exists as a binary file. To increase or change the device class is very convenient, only need to update their corresponding dynamic library. Add two functions named RestartClass and KillClass to the DServer class, which makes it possible to dynamically upgrade a device class.

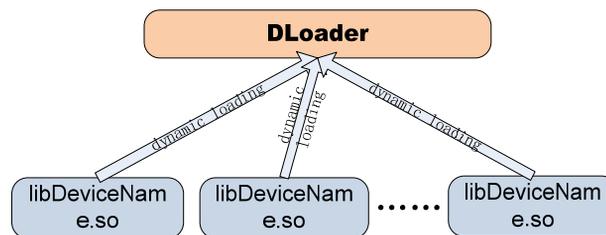


Figure 4: Dloader loading the dynamic library at run-time.

Starter++

Starter++ is similar to the Starter feature in the Tango toolset, which adds deployment and upgrades to the original functionality. Starter ++ is a standard software that must be installed when the embedded front controller is online. After that, the remote client (VisualDM) completes the deployment and registration of the software device through the deployment function; in the software device maintenance process, if you need to upgrade the device, through the remote client can achieve the device dynamic library upgrade and restart function.

Device Monitor Model (DMM)

As provided in Fig. 5, the rules of Device Monitor Model are as follows:

1. Each tango device is a node that can be monitored.
2. There is a dependency between nodes, which can be expressed by acyclic directed graphs.
3. The combination of nodes is a collection node.
4. Give role permissions to manage nodes.
5. The actor belongs to a specific role and can belong to multiple roles.

The control functions of the node monitoring device include startup / restart / stop, registration / logout, deployment / upgrade, and so on. The status displaying function of the node monitoring device includes device running status, device state and status, front node status, and so on. If it is a collection node, the total status is displayed.

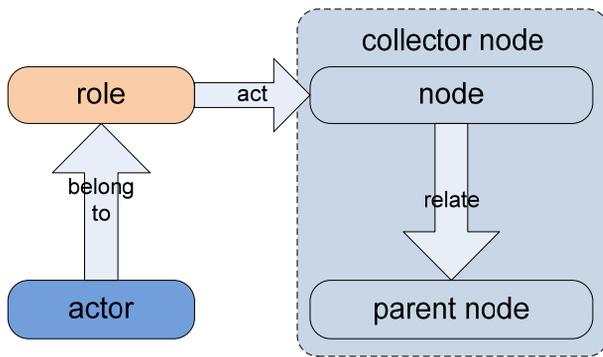


Figure 5: Device monitor model diagram.

Implementation of Device Management Based on DMM

Based on the device monitor model, VisualDM implements the full lifecycle management of the software device, including: device class management, device deployment, monitoring visualization (editing and running), user management and so on. Fig. 6 is the monitoring interface of VisualDM.

1. User management: different roles see different pages after the user logging in. The roles are admin, manager, operator, monitor, etc.
2. Import the dynamic library of the device class: Open the platform software named VisualDM, Go to the menu to import the dynamic library, import new device or new version of the device to the device library, you need to describe in detail the operation of such device and special requirements.
3. New node monitoring device: According to the device class in the device library, generate new devices, configure the host and server to which the device belongs, deploy the device to the corresponding host, register the device into the database, run and test it.
4. Design monitoring screen: From the list of devices on the left, drag the node monitoring device or collector node to the right of the monitoring editor page, connect nodes according to dependencies. The collection

of all newly generated nodes in the right monitoring editor page is saved as a new collector node. This collector node can be embedded as a node in other monitoring screens.

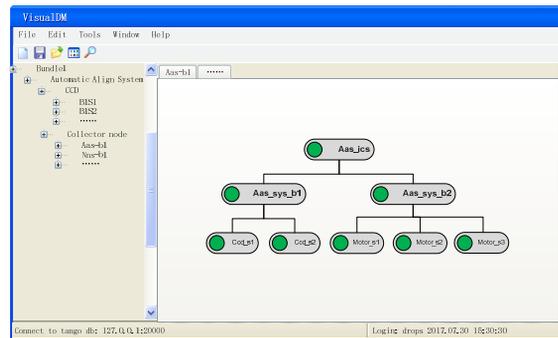


Figure 6: VisualDM running the monitoring interface.

5. Run the monitoring program: In the monitoring process, the monitoring screen is hierarchical, from macro to micro, layer by layer refinement. As shown in the figure, the background color of each node represents the overall state of its parent node; the green circle on the left side of the node indicates its own state. Users can control according to their own authority. In Table 1, We defined the color monitoring node state.

Table 1: State of the Monitoring Node and Associated Color

State	Associated Color
NOT STARTED	black
UNKNOWN	grey
INIT	beige
FAULT	red
DISABLE	magenta
OFF	white
STANDBY	yellow
MOVING	Light blue
...	...

CONCLUSION

Design and implementation of VisualDM meet the diverse needs of different users monitoring, solving a problem with a single monitor screen is difficult to meet all user needs in the past. The platform allows users to independently define a variety of monitoring interfaces and dependencies, and provides a production process of software device from scratch, while real-time monitoring of device running status. With the continuous operation of facility, when we encounter new needs, we will continue to optimize the platform software to protect the long life cycle of the control system. In the future we hope to achieve adaptive deployment, load detection and other functions.

REFERENCES

- [1] The TANGO device server model, <https://tango-controls.readthedocs.io/en/latest/development/device-api/device-server-model.html>
- [2] Robert W. Carey *et al.*, “Large-scale CORBA distributed software framework for NIF controls”, ICALEPCS 2001.
- [3] L. J. Lagin *et al.*, “The Overview of the National Ignition Facility Distributed Computer Control System,” ICALEPCS 2001, San Jose, CA, USA.
- [4] TANGO Jive, <https://tango-controls.readthedocs.io/en/latest/tools-and-extensions/jive/index.html>
- [5] TANGO Astor, <https://tango-controls.readthedocs.io/en/latest/tools-and-extensions/astor/introduction.html>
- [6] Hierarchical control system, https://en.wikipedia.org/wiki/Hierarchical_control_system
- [7] 姚导箭, 刘金, 魏仲平, 等. 高功率激光装置集中控制系统软件架构研究. 计算机工程与设计, vol. 28, pp. 1737-1740, 2007.
- [8] 刘金, 姚导箭, 等. 大型激光装置集中控制软件框架及组件设计. 计算机工程与设计, vol. 32, pp. 3534-3538, 2011.