International Conference on Control, Automation and Systems 2007
Oct. 17-20, 2007 in COEX, Seoul, Korea

XML based Robot Description Language

Ji hwan Park[1] Tae Houn Song[2], Soon Mook Jung[3] and Jae Wook Jeon[4]

[1 2 3] School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea
(Tel : +82-31-290-7937; E-mail : {fellens, thsong, kuni80} @ece.skku.ac.kr)
[4] School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea
(Tel : +82-31-290-7129; E-mail : jwjeon@yurim.skku.ac.kr)

Abstract: This paper discusses XML based Robot Description Language which can describe structures and ability of any robot using common documents. XML is the most promising language for storing and delivering information. So structures and abilities is described using XML. A robot's structure and ability is stored using XML and grouping. The robot's structure is consisted of joints, motors, frames. Joints are a ball joint, a hinge and a slider joint. And a developer can create user definition frame. RDL can not only be use to show specifies of any robots but also linked robot simulator. Then the robot simulator can check robot stability.

Keywords: XML, Robot, RDL, Description Language, Language

1. INTRODUCTION

As robots evolve, robots are increasingly being used for a variety of functions. As a result, the robot's structure and ability are becoming more complicated because of the increase in technical requirements. Robots are made by many companies and form a wide range of development tools. So the robot is described documents using each other form. Then, A user can not easy understand many robot' structure and ability. Because a user must know many form which describe structure and ability of any robots.

We need a common language that describes robot structures and abilities. However, up until now, no such language has existed. Languages that describe the robot interface exist [1]. But languages that describe the robot's structures and abilities do not exist.

This paper discusses XML based Robot Description Language which can describe the structures and ability of any robot using common documents. This XML based RDL can be used for various fields of application.

VHDL, which is a common descriptive grammar of hardware and a standard of the digital system can not only describe the functions of hardware, but also uses a digital system design tool. XML based RDL can use a robot design tool and be linked to a robot simulator. So the simulator can check the robots stability.

In this paper, XML is used as a database organized using Robot Description Language. It stores a robot joint with and without a motor, and robot frame. And it is saved for all sensor features of the robot. Sensor features are installed in the robot, and measure data. The motor features of the robot are saved using XML.

2. XML

In the past, any paper which define user interface program using XML is existed. These are RobotML, XUL, and UIML. These define a user interface program that executes on a wide range of platforms and uses a common language.

Extensible Markup Language or XML, is currently the most promising language for storing and delivering information on the World Wide Web. XML has a highly flexible syntax that allows it to be used to describe virtually any kind of information, from a simple recipe to a complex database [2].

2.1 RobotML

Robot Markup Language or Robot ML is an experimental XML based markup language that is used for communication between autonomous mobile robots and robot components.

2.2 XML User Interface Language

XML User interface Language is an XML language based on W3C standard XML 1.0. The XML User interface Language is Mozilla's XML-based user interface language that lets you build feature rich cross-platform applications that can run connected to or disconnected from the Internet.

Most applications need to be developed using features of a specific platform, making building cross-platform software time-consuming and costly. A number of cross-platform solutions have been developed in the past. Java, for example, has portability as a main selling point. XUL is one such language designed specifically for building portable user interfaces. It takes a long time to build an application even for only one platform [3]. It provides the ability to create most elements found in modern graphical interfaces, like buttons, toolbars, pop-up menus, etc.

XUL based on XML can define various interfaces in one language and has the same purpose as XML based RDL.

2.3 User Interface Markup Language

User Interface Markup Language is also an XML language. The User interface can be defined to use this language. It is used to describe the location and design of controls. It also defines actions to take when certain events occur.

978-89-950038-6-2-98560/07/$15 ©ICROS

Table 1 “Hello World” example in UIML

```
<?xml version="1.0"?>
<!DOCTYPE uiml PUBLIC "-//Harmonia//DTD
UIML 3.0 Draft//EN"
"http://uiml.org/dtds/UIML3_0a.dtd">
<uiml>
 <interface>
  <structure>
   <part id="TopHello">
    <part id="hello" class="helloC"/>
   </part>
  </structure>
  <style>
   <property part-name="TopHello"
    name="rendering">Container</property>
   <property part-name="TopHello"
   name="content">Hello</property>
   <property part-class="helloC"
    name="rendering">String</property>
   <property part-name="hello"
    name="content">Hello World!</property>
  </style>
 </interface>
 <peers> ... </peers>
</uiml>
```

Table 1. is the famous “Hello World” example in UIML. A typical UIML document is composed of these two parts. First is a prologue identifying the XML language version encoding, and the location of the UIML document type definition. The second is the root element in the document, which is in the UIML tag [5].

3. ROBOT DESCRIPTION LANGUAGE

Robot’s structure and ability is stored using XML and grouping. The robot’s structure is consisted of joints, motors, frames. Joints are a ball joint, a hinge and a slider joint. And a developer can create user definition frame.

3.1 Architecture

The robot’s structure consists of joints, motors, and frames. Joints are a ball joint, a hinge and a slider joint. And a developer can create a user definition frame.

Fig 1 shows that XML based Robot Description Language can define any robot. XML based Robot Description Language is based on XML syntax. This consists of three parts. The first part is the unit of the robot, such as joint, motor, frame, and sensor. The second part is the XML based Robot Database. The XML based Robot Database is consists of the units. To use any units, these are stored in XML based Robot Database. The third part is the robot as defined in the XML based Robot Database. A robot can be defined to use the XML based Robot Database.

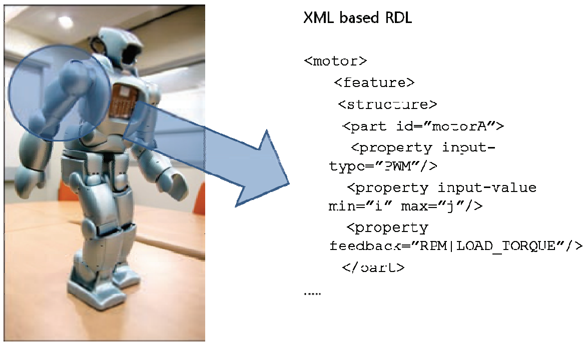

Fig. 1 Overview XML based Robot Descripstion Language

3.2 Joint

The joints are divided into two parts. These are joints with a motor, and joints without a motor. The joints with a motor can move actively. But the joints without a motor cannot move actively. It can move by outer force. It is a passive. In this paper, joint is divided into four parts. It is showed in Fig. 2.

Fig. 2-(a) is indicated cylinder. Hydraulic cylinder and Air pressure cylinder are a kind of this cylinder. Fig. 2-(b), Fig. 2-(C) and Fig. 2-(d) are joints of rotation. The majority of joints are contained in these major joints.

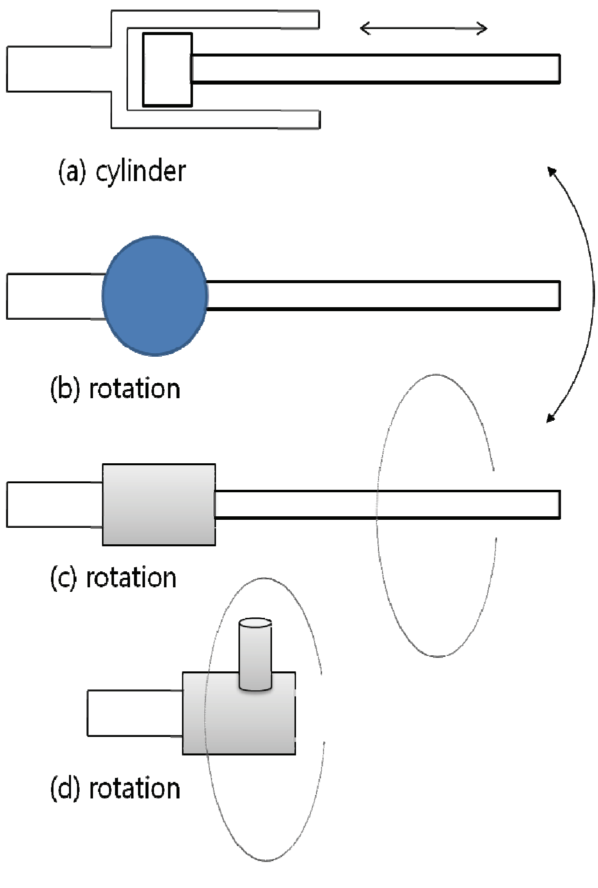

Fig. 2 The joints of a robot

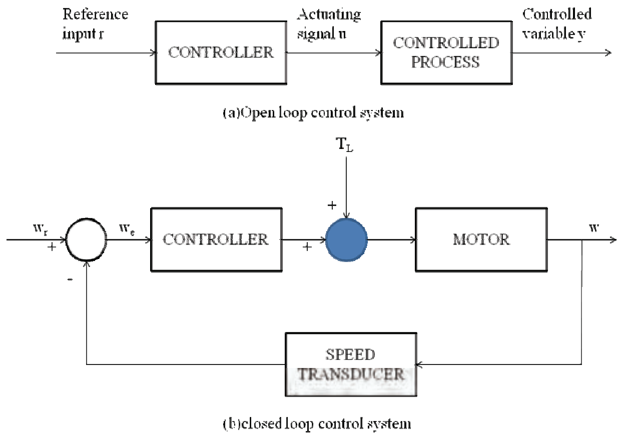

Fig. 3 The block diagrams of control system.

Table 2 A motor of the robot

```
<motor>
 <feature>
  <structure>
   <part id="motorA">
    <property input-type="PWM"/>
    <property input-value min="i" max="j"/>
    <property
feedback="RPM|LOAD_TORQUE"/>
   </part>
   <part id="motorB">
    <property control-type="controlA"/>
    <property volume x="i" y="j" z="k"/>
    <property axis-position x="i" y="j" z="k"
type="cylinder|gear"/>
    <inheritance
part-name="motorA">input-type</inheritance>
    <inheritance
part-name="motorA">input-value</inheritance>
   </part>
  </structure>
  <control>
   <part id="controlA">
    <property type="CLOSED_LOOP" KP="i"
KD="j" KI="k"/>
   </part>
  </control>
 </feature>
 <action> ... </action>
</motor>
```

part: The elements which are defined the name of the robot parts.

property: The elements which are defined the property of the robot parts.

input-type : The elements which are defined the type of the input signal. Normally, the type of the input signal of stop motor is PWM signal.

Input-value: The elements which are defined the maximum and minimum value of the input data

Feedback: The elements which are defined the data type of the feedback data.

Control-type: The elements which are defined the type of controller which operates a motor.

Volume: The elements which are defined the size of the motor In this version, XML based RDL only supports a hexahedron

Axis-position: The elements which are defined the position and type of the motor's axis.

Table 3 A joint of the robot with motor

```
<joint>
<feature>
<structure>
<part id="jointA">
 <property motor-type="motorA"/>
</part>
<part id="jointB">
 <property motor-type="NULL"/>
</part>
</structure>
<link>
 <part id="link_1">
<property type="slider"/>
</link>
</feature>
<etc> ... </etc>
</joint>
```

motor-type: The elements which are defined the motor that it is can link joint. If this element is 'NULL', the joint cannot link every motor.

Link: The elements which are defined the joint

Fig. 3-(a) is rather unsophisticated and is called an open-loop control system. It is not difficult to see that the system as shown would not satisfactorily fulfill critical performance requirements. The elements of an open-loop control system can usually be divided into two parts. These are the controller and the controlled process. An input signal or command r is applied to the controller, whose output acts as the actuating signal u; the actuating signal then controls the controlled process so that the controlled variable y will perform according to certain prescribed standards.

Generally, the system that controls a motor is a closed-loop control system. Closed-loop systems have many advantages over open-loop systems. It is called a feedback control system. Fig3-(b) shows a block diagram of the closed-loop systems [7,8].

Table 4 A frame of the robot

```
<frame>
<feature>
<structure>
<part id="frameA">
  <property frame-type="cylinder"/>
<property frame-volume x="i" y="j" z="k"/>
<property frame-link_position x="i1" y="j1" z="k1" frame-joint_type="jointA" frame-link_type="link_1" motor-type=" motorB"/>
<property frame-link_position x="i2" y="j2" z="k2" frame-joint_type="jointB" frame-link_type="link_1"/>
</part>
</structure>
</feature>
<etc> ... </etc>
</frame>
```

frame-type: The elements which are defined the shape of the robots frame.

frame-value: The elements which are defined the size of the robot's frame.

frame-link_position: The elements which are defined the position which is able to link the another frame or motor.

frame-joint_type: The elements which are defined the type of the joint which is already defined in motor part.

frame-link_type: The elements which are defined the type of the link. which is already defined in motor part.

motor-type : The elements which are defined the type of the motor which is already defined in motor part.

Table 2 is XML code that defines a motor. The motor feature consists of 3 parts. There is a structure, a control, and an action. The structure defines input value, input type, output value, and output type. And it also defines external features. The code of table 2 defines motorB. Its input type is a PWM signal. The maximum value of the duty rate of the PWM signal is j, and the minimum value of the rate of the signal is i. motorA inherits two properties of motorB. These are input-type field, and input-value field. motorA uses controlA which is the control system that it is defined in the control field. And motorA has the volume field and axis-position. The volume field expresses the size of motorA. And the axis-position field indicates the position of the motor's axis and the type of the axis. A control field defines the control system. The code of table 2 explains controlA which is a control system. It is a closed-loop control system and has 3 variables.

The action field defines the concrete motion of the motor and events. It contains any data that includes the direction of rotation, motor's feedbacks, bounds of rotation, power, and etc..

Table 5 A range sensor feature of the robot

```
<sensor>
<feature>
<structure>
<part id="rangeA">
  <property sensor-type="sonar"/>
<property sensor-volume x="i" y="j" z="k"/>
<property sensor-position x="i" y="j" z="k"/>
<property sensor-vector x="i" y="j" z="k"/>
<property sampling="100ms"/>
<property value-type="8-digital"/>
</part>
</structure>
</feature>
<etc> ... </etc>
</sensor>
```

sensor-type : The elements which is defined the type of the sensor. These are the range or value which can be measure

sensor-volume: The elements which is defined the size of the sensor.

sensor-position: The elements which is defined the position of the sensor.

sensor-vector : The elements which is defined the vector that the sensor look at position.

sampling : The elements which is defined the time of the sampling time.

value-type : The elements which are defined the type of the feedback data or input data.

Table 3 is a data of joints of robot with motor. It defines jointA and jointB. jointA can connect a motor of motorA type and has the property of a motor. But jointB cannot connect all motors. Because the type that the joint can connect is NULL. This kind of joint must connect other frames. A link field describes the features of the joint. The features are operating state, way, and joint type that the joint can connect. Data of the features is information inherited to define the frame.

3.3 Frame

Frames decide a shape and a structure. The developer can make a developer definition frame which is made of union frames. A joint type that the frame can connect to is inherited from the joint data field. And a motor type that the frame can install is inherited to the motor data field.

Table 4 shows the definition of a frame. It describes frameA. Its shape is a cylinder. And a position that can connect motors or other frames is defined as link_position field. This frame has two copulas. One copula can connect another frame, but cannot connect a motor. Another copula can connect a motor where the property of the motor has motorB.

Table 6 A camera feature of the robot

```
<sensor>
  <feature>
    <structure>
      <part id="camera_main">
        <property frame-type="camera"/>
        <property sensor-volume x="i"
        y="j" z="k"/>
        <property sensor-position x="i"
        y="j" z="k"/>
        <property sensor-vector x="i"
        y="j" z="k"/>
        <property sampling="100ms"/>
        <property value-type="image:bmp"/>
        <property image-type="640;480;24"/>
      </part>
    </structure>
  </feature>
  <etc> ... </etc>
</sensor>
```

3.4 Sensor

The kinds of sensors are a very wide range. In this paper, XML based RDL defines only the range sensor and camera module. The basic features are the position of the sensor, size of sensor, vector of sensor's direction, and sampling rate.

Table 5 defines range sensor. This sensor's size is described as sensor volume. And the position of the sensor is defined as a sensor-position field. This sensor looks at the sensor-vector direction. This sensor's sampling rate is 100ms, and the return value is an 8bit digital value.

Table 6 shows a camera module. The most common property is the sensor properties. These properties are sensor-volume, sensor-position, sensor-vector, and sampling rate. The value type of the camera module is an image. And the format is bmp. This resolution of the format is 640x480, and the color depth is 24bit.

3.5 XML based Robot Database

In the previous section, we know that joints, motors, frames, and sensors are can be stored to use XML syntax. XML based Robot Database is made these XML data. Then a robot can be defined by the database. The frames can make a robot shape. And the motion of the robot can be described by a motor data field. Fig. 4 showed the architecture of the XML based Robot Database. A developer can define a robot to use this database.

3.6 XML based RDL

In previous section, a method to store database, joints, motors, frames, and sensors in XML syntax is presented. A developer can define a robot to use this database.

Fig. 5 is shows the architecture of XML based Robot Description Language. To make a robot, a developer must register units. The units are joints, motors frames and sensor data field.

4. CONCLUSION

This paper says XML based Robot Description Language. This method is a language which describes robot's structures and ability. XML based Robot Description Language is able to define any robot. When XML based RDL is used, a user can understand robot's structures and ability. XML based Robot Description Language can define any robot. XML based Robot Description Language is based on XML syntax. This consists of three parts. The first part is the unit of the robot, such as joint, motor, frame, and sensor. The second part is the XML based Robot Database. The XML based Robot Database is consists of the units. To use any units, these are stored in XML based Robot Database. The third part is the robot as defined in the XML based Robot Database. A robot can be defined to use the XML based Robot Database.

And when RDL is used, many robots can be described common documents. Simply, it can use to show specifics of robots. And it can be linked robot simulator. Then, the robot simulator can check robot stability in development step.

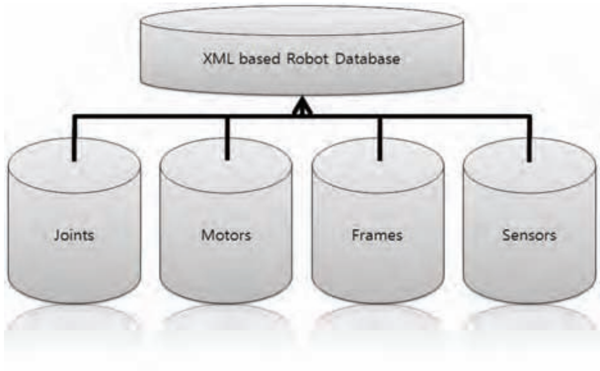

Fig. 4 A architecture XML based Robot Database.

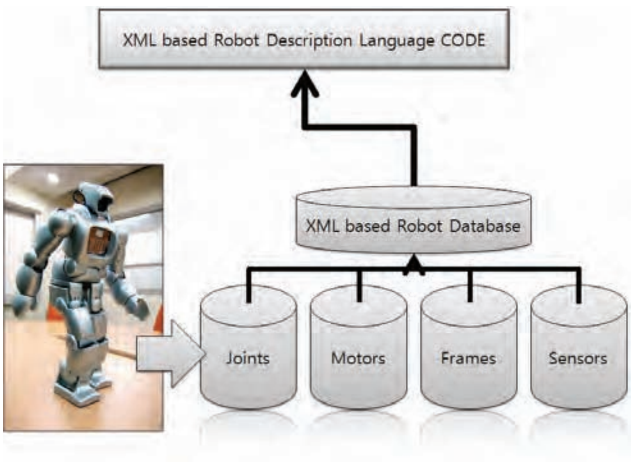

Fig. 5 A architecture XML based RDL

REFERENCES

[1] A. Mueller, T. Mundt, W. Lindner, "Using XML to semi-automatically derive user interfaces," *User Interfaces to Data Intensive Systems 2001*, Proceedings. Second International Workshop on 31 May-1 June, pp. 91-95, 2001

[2] Michael J. Young., "Step by Step XML," *Microsoft,*

[3] M. Makatchev, S. K. Tso, "Human-robot interface using agents communication in an XML-based markup language, Robot and Human Interactive Communication, RO-MAN 2000, pp. 270-275, 2000

[4] http://www.uiml.org

[5] http://www.oasis-open.org/committees/download.php/5937/uiml-core-3.1-draft-01-20040311.pdf

[6] Ogawa Koichi, Katoh Ryozo, The basic robot technology

[7] Benjamin C. Kuo, Farid Golnaraghi, Automatic Control System, WILEY

[8] www.drrobot.com, WirobotSDK Application Program Interface [API] Reference Manual