SELECTED ENGINEERING PROBLEMS
NUMBER 4
INSTITUTE OF ENGINEERING PROCESSES AUTOMATION
AND INTEGRATED MANUFACTURING SYSTEMS

Krzysztof FOIT*

Institute of Engineering Processes Automation and Integrated Manufacturing Systems,
Faculty of Mechanical Engineering, Silesian University of Technology, Gliwice, Poland
*krzysztof.foit@polsl.pl

MAPPING OF THE ROBOT SCENE USING EXTENSIBLE MARKUP LANGUAGE

Abstract: The paper shows the example of using Extensible Markup Language (XML) coding for mapping of the industrial robot scene. Due to its popularity, the XML is recognized by many engineering applications and, because it uses a text format, it could be easily interpreted and converted to other format. The human readable form is another advantage of the XML, so in simple cases it is enough to read the listing to understand the substance. The XML documents could be also processed by using so called XML Style Sheets (XLS), what simplifies machine decoding and translating. In this manner, we can map the robot's environment using XML notation in the similar way as the terrain mapping.

1. Introduction

The beginnings of the Extensible Markup Language are dated in the 1960s. The first appearance of the markup language was GenCode in 1967, which was developed for text processing purposes. Its creator W.W. Tunnicliffe has focused on the independence of text processing and the target typesetting devices that often use specific printout control notation. The idea gives a birth to the Standard Generalized Markup Language (SGML in short), which has been standardized by the ISO committee. However some sources point to the other event, which has major influence on the final form of the SGML: the IBM employer and researcher C.F. Goldfarb, along with E. Mosher and R. Lorie, has developed the GML language. Initially it has been invented for company's internal use, but in 1973 it was presented to publicity [1]. The GML acronym officially stands for General Markup Language, but some say that the name originates from the initials of authors' names.

The SGML language is seen as the base for creation two today's most popular markup languages: the HTML (Hypertext Markup Language) and the XML. The first of them has more in common with SGML/GML languages, because it is also dedicated for text processing purposes. It is used mainly for web pages coding with web browser as a target device. The HTML is standardized by the W3C organization and there are strict rules, which should be followed by programmers and the rendering software. Unlike the HTML, the XML has no strict rules, so the user can define its own structures according the general principles of the language – in this way we could present different data structures, like database records, web contents or graphics elements that are defined in one file.

2. Overall structure of the XML document

The XML document consists of markups and content. The markups in the XML terminology are called tags and – together with the content – they form elements. The elements are arranged as a tree; the hierarchy is as follows:

- root element,
- parent element,
- child element,
- sibling element.

The document can have only one root element, but the number of other elements is not limited [2]. Hereby, the hierarchy looks like in Figure 1.

```
<root>
    <parent_1>
        <child_1>
        content
        </child_1>
        <child_2>
        content
        </child_2>
    ..............
        <child_n>
        content
        </child_n>
    </parent_1>
    <parent_2>
        <child_1>
        content
        </child_1>
        <child_2>
        content
        </child_2>
    ..............
        <child_n>
        content
        </child_n>
    </parent_2>
</root>
```

Fig.1. Exemplary structure of a XML document

The siblings elements have common parent element; in the Figure 1, child_1, child_2 etc. are the siblings.

User can freely define tags inside the XML document and use human-recognizable names for them. The new versions of the XML support almost any Unicode character in document

text (names, attributes, comments etc.), so it can be written in any language. However the documents used by various programs usually use English language for tags and processing commands.

The XML structure could be informational (like database) or operational (when describe how to do such thing). Let's consider the example of the rectangle, which is shown in Figure 2.

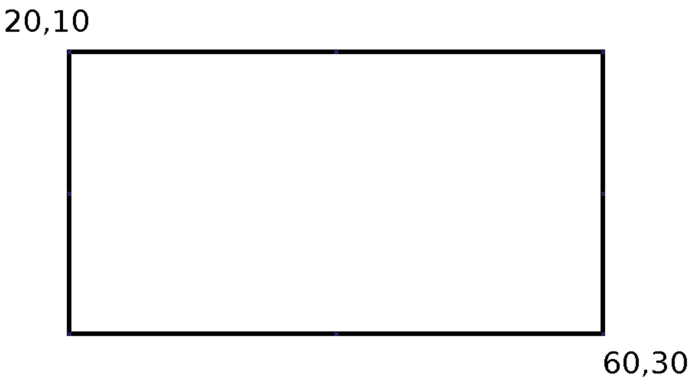

Fig.2. The rectangle considered in the example

We could create informational structure, which describes the rectangle as follows (Figure 3):

```
<rectangle>
    <top_left_corner x="20" y="10" />
    <bottom_right_corner x="60" y="30" />
</rectangle>
```

Fig.3. Description of the rectangle – informational structure

On the other hand we can define the rectangle by instruct someone how to draw it (Figure 4).

```
<rectangle>
    <corner x="20" y="10" />
    <draw direction="right" length="40" />
    <draw direction="down" length="20" />
    <draw direction="left" length="40" />
    <draw direction="up" length="40" />
</rectangle>
```

Fig.4. Description of the rectangle – operational structure

The first approach is useful when collecting information about objects. The structure carries sufficient information to define the rectangle, which has horizontal and vertical pair of edges. The second approach presents more developed structure: this way is better for developing programs and describing algorithms, but sometimes unclear to understand by

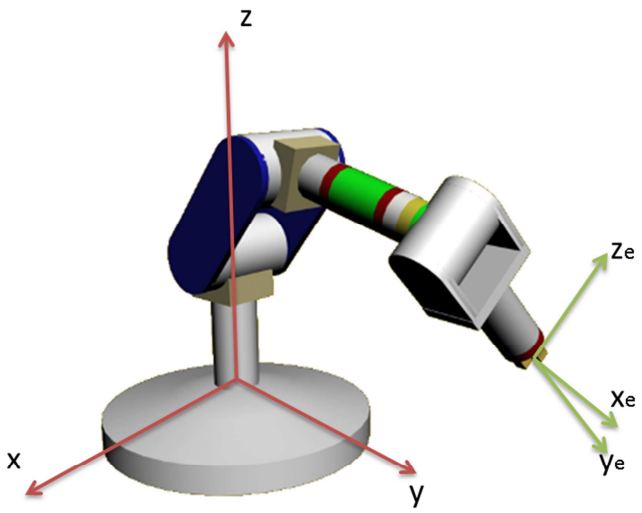

Fig.5. Global and local coordinate system of manipulator

```
<object>
    <type>type_of_object</type>
    <coordinates>
        <x_coord>x</x_coord>
        <y_coord>y</y_coord>
        <z_coord>z</z_coord>
    </coordinates>
    <angles>
        <a_angle>alpha</a_angle>
        <b_angle>beta</b_angle>
        <g_angle>gamma</g_angle>
    </angles>
</object>
```

Fig.6. The rectangle considered in the example

4. Conclusion

XML language is widely used in modern information technology issues. Due to its flexibility, the documents written in this language are suitable for storing information (like database) or describing algorithms (operations). The most important advantage of the XML document is its syntax and text form, which allows machine processing, but is readable and understandable by human.

In this paper, I have discussed the possibility of using XML to store configuration of the manipulator workspace. Presented structure of the XML elements refers only to the Cartesian coordinate system and does not cover all the possibilities of storing the data. It should be noted that the process of creating the structure of the XML elements will vary depending on the type

of the robot, used coordinate system or the amount of information about the object. Such document serves as a knowledge base for programming of a robot, but it is possible to expand the XML definitions by adding the ability of description of the manipulator's movements (operational structure).

Using effector's coordinate system as a coordinates source is not only effective way of identifying the position of the objects in the robot's workspace, but this information is also valuable for automated reasoning to automatically create the robot's program.

An XML document is also a way of code standardization, allowing its subsequent processing by other computer software, what increases the portability of information.

References

1. Goldfarb C. F.: The Roots of SGML -- A Personal Recollection, [online] Retrieved from: http://www.sgmlsource.com/history/roots.htm [Accessed: 25 Jun 2013].
2. XML - Managing Data Exchange, Wikibooks, The Free Textbook Project, 2012, [online] Retrieved June 25, 2013 from http://en.wikibooks.org/w/index.php?title=XML_-_Managing_Data_Exchange.
3. Makatchev M., Tso S. K.: Human-robot interface using agents communicating in an XML-based markup language, Robot and Human Interactive Communication, 2000, RO-MAN 2000. Proceedings of the 9th IEEE International Workshop on Robot and Human Interactive Communication, Osaka, Japan, 2000, 270-275.
4. Zhang Y., Golovinsky A., Yim M., Eldershaw C.: An XML-based scripting language for chain-type modular robotic systems, 8th Conference on Intelligent Autonomous Systems (IAS), 2004.
5. Wang Q., Liu S., Wang Z.: A New Internet Architecture for Robot Remote Control, Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, 2006, 4989-4993.
6. de Rivera G. G., Ribalda R., Colas J., Garrido J.: A generic software platform for controlling collaborative robotic system using XML-RPC, Advanced Intelligent Mechatronics, Proceedings, 2005 IEEE/ASME International Conference on, 2005, 1336-1341.
7. Christanto N., Luo E., Lim, L.: Chalk Robot, Final Report for ECE 445, Univ. of Illinois, 2013.