



# CastML – a language for description of casting products and processes

A. Stawowy <sup>a,\*</sup>, R. Wrona <sup>b</sup>, A. Maciol <sup>a</sup>

<sup>a</sup> Faculty of Management, AGH University of Science and Technology, Gramatyka 10, 30-067 Krakow, Poland

<sup>b</sup> Faculty of Foundry Engineering, AGH University of Science and Technology, Reymonta 23, 30-059 Krakow, Poland

\*Corresponding author. E-mail address: astawowy@zarz.agh.edu.pl

Received 17.06.2008; accepted in revised form 18.07.2008

## Abstract

This work presents CastML – an XML dialect for description of casting products and processes. CastML is extension of MatML which is an extensible markup language designed specifically for the exchange of materials information. The set of CastML tags allows to describe materials' information as well as technological processes, engineering drawings, products classifications and products manufacturers. CastML is simple, understandable and flexible language which makes it attractive for the specialist involved in any aspect of casting research, development, production, or design.

The details on key elements of the proposed language are given. Practical application of CastML is illustrated on the example of an iron castings description.

**Keywords:** Application of information technology to the foundry industry, XML

## 1. Introduction

The exchange of information through electronic media is more and more often used not only as a tool to solve the simple problems of commercial cooperation but also as a means to deal with more complex issues of e-collaboration. In the traditional form of commercial relations between manufacturers and buyers/users of products of all possible types (castings included), the key role the technical standards have played. Unfortunately the knowledge comprised in standards does not fit the requirements of contemporary systems for information exchange.

Modern form of cooperation between the economic entities, like e-commerce and e-collaboration, consists in transferring some activities connected with information transfer and decision-making processes onto largely autonomous electronic systems. For more advanced solutions, such as B2B (business-to-business), shopping agents, e-procurement, e-collaboration or Virtual Organizations, there appears the necessity of the cooperation of

many suppliers, buyers and other market participants in the interchange of information of different character and structure. Because of the diverse character of computerized systems used by the interchange participants, the traditional EDI (Electronic Data Interchange) system cannot be considered. It has become necessary to elaborate standards for structured documents and data on the Web [1, 2]. One of the most promising approaches is to use of XML (eXtensible Markup Language) as a Web services standard. A wide range of XML dialects exist: from MathML (for description of mathematical problems), UBL (Universal Business Language) to MatML (describing material properties) or PPS (production planning and scheduling problems).

The growing importance of Internet-based collaboration among researchers, designers, engineers, managers, suppliers, and customers working in casting industry requires to develop a standard language for information exchange. In this paper we propose such a standard.

We have found in literature only one application of XML which captures the information related to cast products. It is a

Casting Data Markup Language (CDML) which has been developed by Akarte [3]. The CDML consists of two parts: CDML tree and data blocks. The CDML tree represents the hierarchical relationship between different types of information essential for collaboration between the product, tooling and foundry engineer, whereas the data blocks are used for storing the actual project data. It should be noted that CDML is product and project-oriented and does not contain many material and business properties which are necessary to build a web-based platform for information exchange. It is also not well-structured.

## 2. XML basics

XML is metalanguage developed by World Wide Web Consortium [4]. It means, it is a language that describes other languages.

XML provides the facility to define tags and the structural relationship between them. As a result, developers can create their own customized tags (the extensible property of XML) in order to define, share, and validate information between computing systems and applications. Since everything that user creates has to meet the XML criteria and standard, it allows customization without many of the usual risks of customization (such as a lack of interoperability).

XML documents are composed of markup and contents. Six kinds of markup can be present in XML documents [4]:

1. document type declarations,
2. elements,
3. comments,
4. entity references,
5. processing instructions,
6. conditional sections.

Each XML document has a root element within which all other elements are nested. Fundamental to the understanding of XML are the rules all elements must follow [4]:

- every start-tag must have a matching end-tag,
- tags cannot overlap; XML elements have to be properly nested,
- XML documents can only have one root element,
- element names must obey the XML naming conventions:
  - names must start with letters or the "\_" character (excluding the letters "xml" in any combination of case),
  - names cannot contain spaces and the ":" character,
  - the element name must come directly after the "<" without any spaces between them,
- XML is case sensitive,
- XML preserves white space within text.
- elements may contain attributes; if an attribute is present, it must have a value, even if it is an empty string "".

If a document follows the above mentioned rules we can say it is well-formed and valid XML document.

Other XML related standards comprise:

- Document Type Definition (DTD) for describing the schema of XML document,
- eXtensible Style Language (XSL) for translating the document style,
- XML Path Language (XPath) for addressing parts of an XML document,

- XML Schema Definition (XSD) used to express a structure and content of XML document in terms of the valid data model.

## 3. CastML core elements

While developing CastML we have assumed to cover all aspects of business and manufacturing activities connected with casting products. To achieve this we propose six objects related each other:

- a) item – represents basic data of casting products, raw materials, tooling, etc.
- b) material – represents detailed material description of items and it is taken from MatML [5] namespace,
- c) party – represents the participant of supply chain (partially taken from UBL [6] namespace),
- d) order – represents document which is a trigger for supply or sell operation,
- e) resource – represents real manufacturing objects as equipment or tooling,
- f) operation – represents technical knowledge of resources utilization for products manufacturing (partially taken from PPS [7] namespace).

The information of particular objects can be stored in separate files or in one large file, depending on user's application needs. The object description consists of several data sections varying in size and structure.

There are a total of over 70 data tags (not including MatML, UBL, and PPS ones) which form data sections describing various properties of objects. Each data section comprises several pairs of field names and their values; some fields are common to all objects (e.g. *id*, *name*, *type*, *unit*, *count*, *value*, *min*, *max*).

For each field in the data sections, starting and ending tags have been defined, and the corresponding value is stored between the two tags. Attributes (if any) are attached to the starting tag. All tags' names are started with capital letters. The following example illustrates two fields in the *Drawings* data section (from *Item* object).

```
<Drawings count="2">
  <Drawing format="VRML">
    http://agh.edu.pl/3D_solid_model.v
  </Drawing>
  <Drawing format="TIFF">
    http://agh.edu.pl/image_1.tif
  </Drawing>
</Drawings>
```

Like the other XML dialect [5] we want CastML to have the following features:

**Simplicity:** the tags are as simple as possible and are self-explaining.

**Understandability:** the tag names are conventional, standard casting terms, not cryptic, indistinct, or in discriminate symbols, so it is easy to recognize and interpret every markup.

**Flexibility:** CastML is able to handle very complex text and numeric data.

**Extensibility:** new tags can be easily added; CastML also has the flexibility to add other very detailed information through the comments and namespaces.

**Business tool:** CastML can perform as the data exchange protocol for all participants of supply chains, from small businesses to e-market firms.

## 4. CastML example documents

Figures 1–4 provide example instances of CastML documents with special attention to product description. The necessary explaining comments are given via standard XML notation (`<!-- comment -->`).

```
<?xml version="1.0" encoding="UTF-8"?>
<Parties xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-1.0" xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-1.0">
  <Party id="PSSA" type="customer">
    <cbc:Name>Polskie Samochody SA</cbc:Name>
    <cac:Address>
      <cbc:StreetName>Samochodowa</cbc:StreetName>
      <cbc:BuildingNumber>11</cbc:BuildingNumber>
      <cbc:CityName>Tychy</cbc:CityName>
      <cbc:PostalZone>43-500</cbc:PostalZone>
      <cac:CountrySubentityCode>PL</cac:CountrySubentityCode>
    </cac:Address>
    <cac:Contact>
      <cbc:Name>Jerzy Motor</cbc:Name>
    </cac:Contact>
  </Party>
  <Party id="POSA" type="maker">
    <cbc:Name>Polskie Odlewnie SA</cbc:Name>
    <cac:Address>
      <cbc:StreetName>Krakowska</cbc:StreetName>
      <cbc:BuildingNumber>54</cbc:BuildingNumber>
      <cbc:CityName>Radom</cbc:CityName>
      <cbc:PostalZone>13-500</cbc:PostalZone>
      <cac:CountrySubentityCode>PL</cac:CountrySubentityCode>
    </cac:Address>
    <cac:Contact>
      <cbc:Name>Jan Odlewnik</cbc:Name>
    </cac:Contact>
  </Party>
  <Party id="FaMa" type="supplier">
    <cbc:Name>Fabryka Maszyn</cbc:Name>
    <cac:Address>
      <cbc:StreetName>Radomska</cbc:StreetName>
      <cbc:BuildingNumber>211</cbc:BuildingNumber>
      <cbc:CityName>Krakow</cbc:CityName>
      <cbc:PostalZone>30-050</cbc:PostalZone>
      <cac:CountrySubentityCode>PL</cac:CountrySubentityCode>
    </cac:Address>
    <cac:Contact>
      <cbc:Name>Ewa Nowak</cbc:Name>
    </cac:Contact>
  </Party>
</Parties>
```

Fig. 1. Party example instance

```
<?xml version="1.0" encoding="UTF-8"?>
<Orders>
  <Order id="1">
    <Symbol>270/2008</Symbol>
    <Party>Polskie Samochody SA</Party>
    <Date>2008-03-27</Date>
    <Item>cast_1</Item>
```

```
<Qty unit="piece">790</Qty>
<Price unit="EUR">45.43</Price>
<DueDate>2008-07-12</DueDate>
<Priority>5</Priority>
</Order>
<Order id="2">
  <Symbol>275/2008</Symbol>
  <Party>Polskie Samochody SA</Party>
  <Date>2008-04-02</Date>
  <Item>cast_147</Item>
  <Qty unit="piece">680</Qty>
  <Price unit="EUR">21.13</Price>
  <DueDate>2008-07-12</DueDate>
  <Priority>8</Priority>
</Order>
</Orders>
```

Fig. 2. Customer order example instance

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Items>
  <Item id="1" type="product">
    <Name>cast_1</Name>
    <Drawings count="1">
      <Drawing format="JPEG">
        http://agh.edu.pl/image_1.jpg
      </Drawing>
    </Drawings>
    <Dimension>
      <Height>76</Height>
      <Width>110</Width>
      <Length>115</Length>
    <!-- the next 3 fields according to Pacyna classification -->
    <Oblongness>147</Oblongness>
    <Thinness unit="mm">6.7</Thinness>
    <Compactness>0.678</Compactness>
    </Dimension>
    <Class>complex castings with cavities</Class>
    <Subclass>hollow, shape framed</Subclass>
    <Weight unit="kg">
      <Raw>9.36</Raw>
      <Finished>9.02</Finished>
    </Weight>
    <Technologies count="1">
      <Technology name="automatic sand casting">
        <Cores>5</Cores>
      </Technology>
    </Technologies>
    <Economic>
      <DemandPerYear unit="piece">2932</DemandPerYear>
      <MinSaleSize unit="piece">50</MinSaleSize>
      <Price unit="EUR" type="average">43.96</Price>
    </Economic>
    <Material xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance" xsi:noNamespaceSchemaLocation="matml.xsd">
    <!-- material description according to MatML -->
    <xsi:BulkDetails>
      <xsi:Name>A 48M</xsi:Name>
      <xsi:Class>metal</xsi:Class>
      <xsi:Subclass>gray iron alloy</xsi:Subclass>
      <xsi:Specification>ASTM A48</xsi:Specification>
      <xsi:ProcessingDetails>
        <xsi:Name>painting</xsi:Name>
      </xsi:ProcessingDetails>
      <xsi:PropertyData property="p1" source="s1">
        <xsi>Data xsi:format="integer">29,25,20</xsi>Data>
      </xsi:PropertyData>
    </xsi:BulkDetails>
```

```

<xsi:Metadata>
  <xsi:DataSourceDetails xsi:id="s1">
    <xsi:Name>
      Properties of Iron Alloys - Tensile, Creep, and Fatigue
    </xsi:Name>
  </xsi:DataSourceDetails>
  <xsi:PropertyDetails xsi:id="p1" xsi:type="mechanical">
    <xsi:Name>Axial-Stress Fatigue Strength</Name>
    <xsi:Units name="ksi" xsi:Description="kip per square inch">
      <xsi:Unit>ksi</xsi:Unit>
    </xsi:Units>
  </xsi:PropertyDetails>
</xsi:Metadata>
</xsi:Material>
</Item>
</Items>

```

Fig. 3. Product and material example instance

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Resources>
  <Resource id="F1" name="Furnace_1" type="workstation">
    <Operation>melting</Operation>
    <Labours>6</Labours>
    <Area unit="m2">20</Area>
    <Qty unit="kg">
      <Capacity>12000</Capacity>
      <Load min="4000" max="6000"/>
    </Qty>
  </Resource>
  <Resource id="C1" name="Automatic line" type="workstation">
    <Operation>pouring</Operation>
    <Labours>3</Labours>
    <Area unit="m2">30</Area>
    <Qty unit="kg">
      <Capacity>35000</Capacity>
      <Load min="0.5" max="120"/>
    </Qty>
  </Resource>
</Resources>

```

Fig. 4. Resource example instance

## 5. Conclusions

CastML is developed as a part of the research grant "System framework for information exchange of casting products and

manufacturing methods". The language is at the very early stage of development (current version 1.0) so it is not the mature standard. CastML is still being extended and rebuilt. We plan to design a website with system prototype based on CastML; in this case the advantages of unified XML specification to information exchange in e-collaborative environment is obvious.

Our researches confirm that CastML can be very good standard that could work as:

- communication protocols and ontology among e-collaborative software,
- XML schema for description of casting products and manufacturing methods,
- common model and terminology for castings classification problems.

## Acknowledgements

This work is supported by the MN from the resources assigned for scientific researches in years 2006 – 2009.

## References

- [1] S. Zhang, W. Shen, H. Ghenniwa, A review of Internet-based product information sharing and visualization, *Computers in Industry* vol. 54 (2004) 1–15.
- [2] Y.H. Tao, T.P. Hong, S.I. Sun, An XML implementation process model for enterprise applications, *Computers in Industry* vol. 55 (2004) 181–196.
- [3] M.M Akarte, B. Ravi, Casting Data Markup Language for web-based collaborative engineering, *Transactions of the AFS* vol. 112 (2002) 1-16.
- [4] XML Tutorial, <http://www.w3schools.com>
- [5] J.G. Kaufman, E.F. Begley, MatML. A data interchange mark language, *Advanced Mater. & Processes* (2003) 35–36.
- [6] Universal Business Language v2.0, <http://docs.oasis-open.org/ubl/os-UBL-2.0>
- [7] OASIS Production Planning and Scheduling (PPS), <http://www.oasis-open.org/committees/pps>