

## PMXML: An XML Vocabulary Intended for the Exchange of Task Planning and Tracking Information

Kevin Curran, Liam Flanagan and Mark Callan  
Internet Technologies Research Group,  
University of Ulster, Magee Campus, Northland Road, Northern Ireland, BT48 7JL, UK

---

**Abstract:** The project management XML specification (PMXML) is an extension to the XML schema that specifies the elements in use within project management applications. As it is a schema designed specifically for use with project management, it contains a quite an amount of elements that are specific to project management. Furthermore as it is an extension of the XML concept in the project management realm; it inherits the properties of the XML data type. As XML is an open technology, it is free from restraints on cost and platform compatibility and is widely supported. However the problem occurs when we need to understand what the data means, to this effect we need some form of metadata to describe the data, what it is and where it is to be used. This is where the PMXML schema comes into effect, using standardized schemas each PMXML compatible application can exchange data with each other and interpret the data accurately. For this to be effective virtually all objects or elements that may be held in the XML file need to be specified. This has been the goal of the PMXML consortium, to define a standardized vocabulary to facilitate the exchange of information between various project management applications and tools. This paper examines whether it has achieved this goal.

**Key words:** Project management, PMXML, task planning, tracking, SGML, XML

---

### INTRODUCTION

SGML<sup>[1]</sup> and XML<sup>[2,3]</sup> are formal metalanguage facilities for defining markup languages. SGML has the full power to configure a set of features for mark-up languages, whereas XML has a fixed set of these SGML features. XML has the facility for specifying the structure of the document, using a document type definition (DTD), which is a formal description in XML Declaration Syntax of a particular type of document. It sets out what names are to be used for the different types of elements, where they may occur and how they all fit together.

The alternative to a DTD is the Schema<sup>[4]</sup> the specification of which can be found at <http://www.w3.org/XML/Schema#dev>. XML Schema also offers several features, which are urgently required for data processing applications. Such as a set of data types including dates, numbers and strings. According to the W3C<sup>[1]</sup> XML is a language used for the storage, structuring and the exchange of data. It is therefore possible to specify a Schema/DTD that will allow any application to read the data held in the XML files when attempting to make different systems talk to each other, using the XML schema as the definition language and

XML as the data language we can easily import and even get different systems talking to each other. The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD. An XML Schema defines elements and attributes that can appear in a document, which elements are child elements and the order and number of said elements. Therefore if XML is the standard that has been designed for Data Storage, Structuring and Exchange it is only natural that we should have an extension to facilitate this exchange of data between different systems<sup>[5]</sup>.

**PMXML:** PMXML is an XML vocabulary intended to facilitate the exchange and tracking of information between project management suites and tools<sup>[6]</sup>. It provides an extension of the XML schema thereby enabling applications and data providers to provide common definitions for specific types of data. Using this common definition data can be exchanged so that when project management tools interface they can understand each other and can exchange information regarding task and project status, cost, resource assignments and other data used by project management applications.

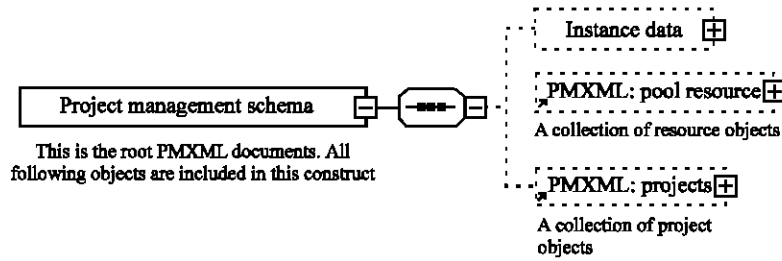


Fig. 1: Project management schema and subsets

PMXML is currently under development as a standard for data interchange between project management applications. It was proposed by Pacific Edge Software Inc. in 2000 under the auspices of the PMXML consortium<sup>[7]</sup>. Primavera Systems, Welcom, eProject.com, Great Plains, PlanView, NASA, Oracle and others have since joined the consortium, which maintains the PMXML standard. The standard itself is in an early stage of development, with revision 2 having been published by Pacific Edge in March 2002. Currently the standard consists of data definitions specifically the four major project management data types and a few minor ones. The four major data types are project, resource, task and assignment. The definition starts with a project management schema (Fig. 1) as a root element and contains Instance data, pool resources and projects<sup>[8]</sup>.

The Instance data is a collection of data that refers to the application that produced the data. It is used to interpret the data from one project management application to another. It contains information on User ID, Application ID and Name and the schema used per application. Pool resources is a collection of object resources linked to the material or working resources, they define the materials and resources used and contain data such as Name and address data, cost (rates, overtime), availability data, plan and baseline, metrics and other applicable data. Currently only 10 or so major currency types have been integrated into the standard. The remaining major data types are task and assignment.

The task vocabulary consists of attributes related to tasks in project management and contains planned start, planned finish, priority, planned cost, actual work, actual finish, baseline attributes, including baseline, start and finish, baseline duration etc. The assignment data type contains a vocabulary to provide details on the assignments within each application, The standard vocabulary is very similar to that of the task definition in that they both contain plan, baseline and actual data objects. However the task definition contains attributes to define a milestone, a summary task and completion status. While the Assignment adds ID's for the tasks and assigned resources.

The project definition type contains data and attributes pertinent to the project itself, such as project name, start date, last modified, description, author, planned start, planned finish, planned cost, actual cost. Once more these values are very similar to that of the previously defined vocabularies. However with the project definition we see in introduction to the author, manager, category, company, subject, start and finish variance and custom attributes. Custom attributes define the custom costs, custom finish dates, flags, numbers and custom start dates. We also see that the project definition has included the sub definitions of local resources, resources, tasks and assignments. Furthermore there is also scope for an extended attributed element which although defined at the root of the project can be used either at the project level or under tasks, assignments or resources.

**A critique of PMXML:** Due to the nature of all XML-based languages, PMXML can be as simple or complex as the language creator wishes it without implying any reflection on that language's 'stage of development'<sup>[9]</sup>. The major determinant in this respect is the complexity of its respective document type definition (DTD), which defines the language by standardizing its vocabulary. A more complex DTD could constitute greater control of the document, allowing more appropriate screen rendering of relevant information, for example. As a language, PMXML will be, primarily, a means of communication. This implies that the more it is used, the bigger it will become and thus more developed. But there are difficulties assessing how much PMXML is used. As a text-based mark-up language it produces few distinguishable characteristics that can be easily measured, traced or researched. As a software language, PMXML is unlike the JAVA software which can be downloaded, or any of the Microsoft set of programs where individual copies of it are sold, making them both more easily quantified. Unlike HTML, JSP or ASP language files, which declare themselves, PMXML has essentially only one traceable characteristic; the accessing of its DTD.

For a language to become 'fuller' it would seem reasonable that any discussion on the development of a language must include some appreciation of how it can be used. As a XML-based language, this is more of a generic discussion on all these language, though with specific reference to PMXML. Finally, continued development of PMXML is only possible with the support, or lack of support, given to the language from other software developers and other interested parties. Otherwise, the development may be of no or little vale, except possibly in a historical sense. Hence, it would seem reasonable to discuss the state of the art of PMXML with reference to these headings:

- The definition of PMXML.
- The potential applications for its use.
- The present support for its development

**The definition of PMXML:** PMXML, like all XML markup languages, is a text-based markup language, which defines what information is, rather than how it is presented. This definition is contained within a document type definition (DTD) file, in this case maintained by the PMXML consortium. In general, when creating a DTD, the maintaining authority may be as lax or as restrictive as they see fit and hope to strike a balance between the convenience of following simple rules and the complexity of handling real data. In the case of PMXML data definition, version 2, they have opted to include 4 major data types, namely project, resource, task and assignment with their respective schemas. The definition starts with the Project management schema as a root element containing 3 collections of objects, namely Instance Data, Pool Resources and Projects. Each of the other 3 data types is broken down into similar collections of objects<sup>[10]</sup>. In essence, it would be fair to say that, because of the nature of the data and how they feel that data could be best shared the Consortium has chosen a very restrictive definition with the resultant formation of complex documents. This has immediate implications for the next point of discussion, the potential application for its use.

**The potential applications of PMXML:** The great potential advantage for PMXML is that it is platform dependent, so that firms with many disparate legacy task planning software applications can use it to enable sharing of their collected data. Whereas in the past, this was solely in the realm of large firms, now, with increasing mobile inter-connectivity using PDAs and mobile phones, for example, this is becoming an increasingly feature for all sizes of business. Using PMXML, with descriptive instead of procedural markup, the same document can

readily be processed in many different ways as a web service application, using only those parts of it which are considered relevant. The DOM API, for instance, can be used to parse the document in a way which makes it easy to perform any specified business logic on the contained chosen data. Web services run on the basic platform of XML plus HTTP. With cheaper bandwidth and storage, increasing use of dynamic content and the plethora of computing devices now present, these can constitute an increasingly efficient and effective means of communication. An application can be directed by the 'client' accessing the PMXML document through a simple text-messaging type service with data/returning in a similar fashion, or through voice commands being issued to a VoiceXML wrapper, which envelops that same document. The data is returned presumably in some text-based form, although synthesized voice could be a future development. Development environments such as J2ME and .NET Compact Framework can give developers robust platforms for developing advanced mobile clients, which can utilize the advantages offer by any XML language in any number of ways. Unfortunately the latter, in particular, also gives the option of other solutions, which do not include PMXML and adversely affect its development.

**The present support of the development of PMXML:** For any software language to continue development it, unfortunately, requires the patronage of powerful forces. These can be software vendors, such as Microsoft or IBM, or, as has been proven by the Linux operating system, by the concerted support of many independent developers through the open-source system. At present, the undeniable market leader in Project management software is Microsoft's Project and its spin-off Microsoft Office Project. Whilst in some ways this simplifies data transfer procedures, because/of Microsoft's attitude, the requirement for this transfer is totally/obviated for 2 reasons.

Firstly, a precursory search of Microsoft's website produces no mention of 'PMXML ' and secondly, '[the] Microsoft Project 2002 product family now includes a separate Microsoft .NET Enterprise Server product Microsoft Project Server 2002 is a customizable platform that can be extended and integrated across business systems to meet a company's project and knowledge management needs. An open architecture and full support for industry standards, including XML and SOAP, make it easy for developers to integrate Microsoft Project Server with other line-of-business applications and databases". Microsoft has effectively curtailed any reason to develop PMXML for data transfer, or storage, from its software.

Similarly, a precursory search of IBM's website produces the same lack of mention of 'PMXML'. SYSPRO, another industry software vendor, proudly proclaims 'Microsoft leverages Microsoft.NET component architecture to deliver SYSPRO e.net solutions'. Indeed the only mention of 'PMXML' is in relation to Pacific Edge Software Inc. which has produced a Project Office Connector for Microsoft Project. Project Office Connector enables customers to implement a closed-loop solution that provides data updates between products. With the solution, users are assured that their portfolio data is always current and relevant. The Project Office Connector supports Microsoft Project versions 98/2000/2002. It also features standard Project Management Extensible Markup Language (PMXML) based connectivity<sup>[11]</sup>. This was published in January 2002, just before the PMXML message board<sup>[12]</sup> was founded. Unfortunately, although this proudly proclaims 66 members, these have only produced a total of 9 entries, the last being posted in November of that same year. Similar searches for 'PMXML' produce no entries in either of two major open-source portals sourceforge.net<sup>[13]</sup> and freshmeat.net<sup>[14]</sup>.

### CONCLUSIONS

The PMXML standard for data exchange between Project management applications is still in the early stage of development. However it currently has quite a large number of schema elements and attributes already defined. The goal of the consortium is to develop a standard to define the data produced by project management applications. Storing data on a computer system can be done in 2 distinct ways, either as a binary file or as a text file. Text files are by far the easiest way to store data, as text is text and if needed can be viewed in virtually any text editor, thereby enabling mobility of data and platform independence. The Choice of XML as the base for the exchange of information between Project management applications is currently the best method for prescribing data types. This is due to the fact that either the DTD or the Schema specifies the data held in any XML file and is a reference for any applications to relate to when attempting to read the supplied data. There are quite a few advantages in using a schema, namely the fact that XML schemas are extensible to future additions, as

can be seen by the PMXML schema. They are richer and more useful than DTD's are written in XML and support data types and namespaces.

The undeniable conclusion however is that PMXML is at the same stage of development as it was at the end of 2002, which is disappointing considering the growth and development of other XML based languages in more recent years. Whilst there are many reasons for this, the major one must be the lack of any worthwhile application. As a conduit PMXML has little relevance as an intermediary between two Microsoft Project applications. These applications can use their own integral conduit. Possibly this is a reflection on the complexity of these types of application, but there has not been the development as yet of any other viable open-source alternative.

### REFERENCES

1. World Wide Web Consortium, <http://www.w3.org/>
2. The XML Schema FAQ, <http://www.schemavalid.com/faq/xml-schema.html>.
3. The XML FAQ - Peter Flynn, <http://www.ucc.ie/xml/>
4. Walmsley, P., 2001. Definitive XML Schema, Prentice Hall.
5. Goldfarb, C.F. and P. Prescod, 2003. XML Handbook, 5th Edn, Prentice Hall.
6. Extensible Markup language (XML), <http://www.textuality.com/xml/>.
7. PacificEdge.com Schema Docs, <http://www.pacificedge.com/xml/schema%20Docs/>
8. Cover Pages: Consortium Advances Project Management Extensible Markup Language (PMXML). <http://xml.coverpages.org/ni2002-06-04-b.html>.
9. PMXML-a XML standard for project management, [http://www.vrtprj.com/content/istandards/pmxml\\_en.html](http://www.vrtprj.com/content/istandards/pmxml_en.html)
10. [http://www.vrtQrj.com/content/istandards\Dmxml\\_en.html](http://www.vrtQrj.com/content/istandards\Dmxml_en.html)
11. Product Briefs by ADT Staff. <http://www.adtmag.com/article.asp?id=6968>, January 2002
12. [http://finance.groups.yahoo.com/group/pm\\_xml/](http://finance.groups.yahoo.com/group/pm_xml/)
13. <http://sourceforge.net/search/>
14. <http://freshmeat.net/search/?q=PMXML>