# XGN –an XML Based Data Format for Representing Chess Games

Santhos Baala RS
VIT University, Vellore 632007
rssanthosbaala@gmail.com

Saikiran Chandha
VIT University, Vellore 632007
saikiranchandha@gmail.com

***Abstract*—Portable Game Notation (PGN) is the de–facto open data format for storing chess games for over two decades now. Although PGN is compact and human readable, it does not facilitate storage of multilingual commentary, images, visualization data, etc. XML Game Notation (XGN) is an effort to set a new standards based data format and to address the requirements of modern chess fraternity. Since XGN is XML based, it is easy to parse, modify and transfer through networks, while retaining human readability at acceptable levels. XGN also achieves integrity and compression by utilizing the popular ZIP compression technique. XGN being extensible, lends itself towards future enhancements through community participation and active adoption. Since XML is ubiquitous, several highly optimized parsers already exist for most platforms and APIs are abundantly available for numerous high level programming languages, which makes the transition from PGN to XGN a lot quicker and easier. Internet chess can hugely benefit from XGN as XML and associated technologies such as XSLT, XPATH, XQUERY, etc. XGN is similar to the popular XML based data formats for other applications, such as Microsoft DOCX file format for word processing and International Digital Publishing Forum (IDPF) EPUB for digital books.**

***Index Terms*—XML applications, XML Based Data Format, Open Data Format, Sports Data Format, Chess Data Representation**

## I. Introduction

Portable Game Notation (PGN) is a plain text based data format to represent one or more chess games with branching recursive variations and text commentary. PGN, although its specification is incomplete [1] , has become the de–facto standard and hence adopted by virtually all the popular chess software. Since PGN is simple and text based, conversion to and from other formats is easy, which has made it very popular. However PGN has not seen any major revision due to the popularity of proprietary file formats and proprietary software in the market. XML Game Notation (XGN), is an attempt to define a new XML based data format, which could replace PGN as the standard for open chess data format. XGN shall provide all the features found already in PGN, maintaining backward compatibility to an extent and shall define a schema for storage of multimedia and visualization features that are common across proprietary formats. Since XGN is XML based, it can be easily adopted by existing web applications and a new breed of applications, creating innovative new applications with the combined power of an open and robust data format.

XGN may not be able to store millions of games, since XML based formats require significant processing power and memory. However, XGN shall serve well as an interchange format for e–mail, tournament bulletins, internet forums, web pages, e–magazines, etc., which contain, provide or make use of chess games in a few hundreds or thousands. XGN could also be used as a pointer to live broadcast streams and share game repository sources. In this paper we compare XGN with PGN, based on space complexity and also discuss about past attempts similar to XGN. We try to understand where XGN stands and explore its potential applications.

The paper is organized as follows. In Section II, we discuss the PGN data format, its structure, applications and the current status. In Section III the Chess Game Markup Language (ChessGML), its specification and investigate the possible reasons for the lack of its active adoption. Section IV lists the basic XML tree schema of XGN and mappings to PGN. We additionally provide the XML Schema Definition (XSD) and a sample XGN game in Appendices 1 and 2 respectively. In Section V the folder structure of XGN is introduced and the usage of ZIP compression technique is illustrated. Section VI measures the space complexity of XGN and PGN. In Section VII, we try to understand the limitations of XGN due to its XML based nature and some techniques to improve parsing. Section VIII concludes the paper by discussing the road map, possible applications of XGN in the internet, mobile and desktop spaces and thus its potential to replace PGN as the open data format for chess games.

## II. The PGN Data Format

The PGN data format is a raw plain text based data format, encoded in American Standard Code for Information Interchange (ASCII). It is supported by a vast majority of chess software and popular websites. A single PGN file can hold one or more games. A PGN game is divided into two sections; the first stores the meta data and the second stores the move text in Standard Algebraic Notation (SAN), specified in the PGN specification document [1] . The move text section also includes optional commentary and recursive variations. The last major revision of PGN was done back in ab94 and some minor enhancements were proposed in ab01 [2] by a few leading chess and software experts of that time. Since then, PGN has not been updated. We briefly examine the two parts of a PGN game here.

### A. The Meta Data Section

The meta data of the game such as white player's name, black player's name, event, date, etc., are contained in this section. The PGN specification defines a simple syntax of enclosing the roster data in a tag pair as shown in figure 1. The tag–pair section is case sensitive, order sensitive and must include values for at least 7 tags, Event (name of the tournament or match event), Site (the location of the event), Date (the starting date of the game), Round (the playing round ordinal of the game), White (the player of the white pieces), Black (the player of the black pieces), Result (the result of the game), respectively. Any additional tag pairs defined in the PGN specification or vendor–specific extensions should be placed only after the seven mandatory tags.

### B. The Move Text Section

The move text section contains the moves of the game denoted by SAN, specified in [1] . It can also optionally include variations escaped by parentheses and text commentary escaped by curly braces. Additionally, each move may be commented upon using standard symbols defined by various Numeric Annotation Glyphs (NAG), specified in the PGN Specification document cite{PGN}. Various vendor–specific extensions have been made in the move text section to include additional types of commentary. An excerpt of the move text section as defined by the PGN specification is shown in figure 2. The move text section is terminated by a single game concluding marker, which is same as the value of the Result tag in the Meta Data section, shown in figure 2.

### C. Observations

We observe that the PGN data format, is extremely simple and human readable. One can create and manipulate PGN files with a basic text editing software at his/her disposal. However, it should be noted that PGN can quickly become cluttered and lose its human readability if the game is filled with commentary and variations. This defeats the purpose of maintaining human readability in PGN. We should also observe that PGN, being a text based data format occupies a lot of space as compared to a serialized or binary data format and has an inherent limit in the number of games it can store, based on the processing machine's performance characteristics and hardware.

PGN does not facilitate storage of multilingual commentary since it is being encoded in ASCII. However, vendors have already started to use other encoding formats like UTF–8 [3] to enable multilingual commentary. Several such vendor specific modifications exist for PGN, mostly by ChessBase GmbH, Germany, but they have neither been documented anywhere nor standardized.

## III. The ChessGML Data Format

ChessGML is the first ever attempt to move away from the PGN format and define a concrete XML based format to represent chess games. However, ChessGML has not been adopted by any major software vendor till date. ChessGML, unlike PGN has the capability to use any encoding format, thus making it easy to add multilingual text. ChessGML tries to pack a set of games into a single <tournament>element, further by the <eventinfo>, <players>, <crosstable>and <rounds>elements. <rounds>element contains the actual games and the per–game info. We do not dwell deeper into ChessGML as its specification is vast and beyond the scope of this paper. However, we encourage the reader to visit the ChessGML website and refer to its distribution [4] .

### A. Shortcomings of ChessGML

ChessGML format groups its game contents into a single <tournament>element, which may be semantically misleading in some cases. Also, meta information is too granular in several places resulting in the increase in the size of the ChessGML document. ChessGML does not focus on the inclusion of multimedia and visualization content. ChessGML <moves>element can hold SAN moves or pure XML moves. Obviously pure XML moves increase the size of the document significantly, making it unsuitable for holding a large number of games. The shortcomings listed, mainly the size problem should have been the primary reason for the lack of success of ChessGML. We however must appreciate the attempt by the ChessGML author to create a new data format as the first step towards replacing PGN.

## IV. XGN –Schema

Fig. 3 illustrates the basic schema of XGN through a tree representation. The <xgn>element serves as the root element of the document. The <xgn>element contains one or more <game>element, which acts as a container for the game. We discuss in detail, what each element means and how XGN incorporates features of PGN. Refer to Appendix A for the XSD of XGN.

### A. The <meta>element

The <meta>element, as its name says, itself and its children hold the information about the game as described in Section II. The name meta is semantically understood, even by generic XML parsers and we collect all the meta data for a particular game in one place rather than scatter them like ChessGML. Here, we define four mandatory elements, <event>, <white>, <black>and <result>, respectively. The <event>element holds information about the event and must have the mandatory attributes, site, date and round. The <white>and <black>element hold information of the players of white and black sides respectively. The <result>element holds the game result, 1–0, 0–1, 1/2–1/2 or *. Additional attributes and elements can be added to the <meta>element, for e.g. fideid could be added as an attribute to <white>and <black>elements. Thus, with the meta tag, we define a mapping to PGN's meta data section described in Section II–A.

### B. The <variation>element

The <variation>element contains the actual move text in SAN format. We make some minor modifications to SAN so as to save storage space. We do not insert move numbers and all the moves are delimited by a single space character and NAG tokens are appended to a move without space with the DOLLAR symbol placed between the move and the NAG token. Additionally, we remove characters like x, +, ++, #, e.p. and = which indicate piece capture, check, double–check, checkmate, pawn en–passant and promotion respectively. All variation elements must specify a location attribute and the main variation is indicated by setting a variation's location to the value 0. A variation branch is indicated by appending the location of the target variation in front a variation's location and a . character, followed by the move at which the branch occurs. A variation branching from another variation follows the same location pointing mechanism. For e.g two variations branching from the main variation at half move 22 and 23 would have the locations, 0.22 and 0.23 respectively and variations branching out of the variation with location 0.22 would have a location of the format 0.22.x, where x stands for the half move where branching should occur. If there happens to be two or more variations branching from the same half move in the same variation, then the location format must further distinguish the target variations by appending alphabets a, b, c, etc., to the end of the location value of the target variations. The concept of variation branching is illustrated in Fig. 4. The idea is quite simple, and solves the problem of recursion and any circular references. Although variations are not very much navigable by humans as in PGN, XGN removes a whole lot of clutter and makes things easier for parsers.

### C. The comment class of elements

Comment, is not an element by itself, but rather, represents a class of elements such as <text>, <audio>, <video>, <piecepath>, etc., each comment attaches itself to a variation by specifying the target location through the location attribute, for which the mechanism was elaborated in the previous section. The names for various elements have not been finalized as it requires community participation and approval process. However, comments include features found in PGN along with audio, video and other type of special commentary such as piece–path, pawn–structure, square–colors, arrows etc.

Text comments could be in any language and we may optionally indicate the language of the comment using the lang attribute, which holds the language code as defined by the ISO 639.2 standard [5] . Audio, Video and multimedia comments could point to a source in the adjacent folder or any valid URL address through the src attribute.

## V. XGN –Document Structure

XGN is not just a plain XML text document, but actually a few files bundled in ZIP folder. Microsoft's family of Office Open XML family of file formats [6] and EPUB 3.0 ebook format [7] are very similar in structure to XGN. However, we do not follow the Open Packaging Conventions (OPC), defined in [6] , owing to the overheads incurred.

### A. meta.xml

This file describes about the database. It contains information about the author, date of creation, version, update repositories of the database. We do not propose the schema of this file, since the application context is not well defined at this point. Nevertheless, the purpose of the file is well understood. Parsers shall use the sources provided in this file to connect to any live sources, say a broadcasting service. Also sources to update the particular database could be provided in the same file.

### B. db.xml

This file contains the actual games, for which the schema was discussed in Section IV. There can be only one database inside a single XGN package.

### C. media

This is a folder, which contains all the media files referred by the db.xml. We do not pose any restrictions on the file format of the audio and video files.

### D. index.html/index.rtf

An optional description about the database, which may provide an introduction to the user. This may be used as a tournament report, analysis report, etc., pertaining to the database. The contents referred by the HTML page, if any is also stored inside the media folder, including any CSS and Javascript files.

### E. ZIP Archive

The ZIP archive performs three functions. First, it packages all the contents into a single file for easy distribution and sharing. Second, we save a lot of space, up to 75p.c savings when files are archived as shown in Section VI. Finally, it provides optional encryption and security. We use the most common DEFLATE algorithm [8] for compression. The appnote on ZIP [9] discusses about the applications of the compression technique in terms of packaging and encryption, which helps understand the technique's usage in XGN.

## VI. Space Complexity Comparison

For comparing XGN against PGN we inspect the file sizes in both their compressed and uncompressed states. Fig. 6 shows the file–size comparison chart. The files sizes were measured in the Linux ext4 File System. For simulating the common use–case, we make use of a popular free E–chess magazine, The Week In Chess (TWIC) databases available at their website [10] . All the games we have used do not contain any variations or commentary, since majority of freely available games and databases are like such. Nevertheless, both PGN and XGN have capability to store text comments and variations, additionally XGN can store many other types of comments as discussed in Section IV.

The results show that XGN is always lesser in size than PGN and the gap becomes larger as the number of games increases. The reason for XGN being lesser in size, even with all the overhead of XML is due to the fact that we store meta data much more efficiently as attributes compared to tags in PGN and we have made some space–saving modifications to SAN inside the move element. We achieve an impressive compression, firstly due to DEFLATE algorithm's efficiency and secondly that, chess moves fall under a small character set, a–h, 1–8 for representing squares and the letters K, Q, R, N, B for pieces. Thus, both PGN and XGN benefit from DEFLATE compression and XGN a little more, because we eliminate some non–repeating characters such as move numbers in SAN.

## VII. Limitations

XGN is XML based, which in turn is text based. Thus, parsing files containing millions of games is CPU intensive and impractical on mobile or even desktop systems of an average user. XGN stored in its native XML format in a database may slow down the system significantly [11] . Eventually, time will solve the problem since hardware capability is increasing every single day. However, we could increase the capacity of XGN by employing efficient parsing techniques and exploiting the multi–core CPU architectures by creating specialized parsers as discussed in [12] and [13] . To reduce the storage space, we have proposed the standard ZIP compression technique in this paper, another technique is to encode XML in a binary format which may also improve the parsing performance. A new standard, Efficient XML Interchange (EXI) [5] by W3C is quickly gaining popularity and XGN could make use of the EXI format when public APIs and tools are supported by the popular operating systems.

## VIII. Conclusion

XGN, as discussed in Section VI does well, compared to PGN. Community participation and active adoption of XGN shall help improve the format and set a new standard for chess data representation and associated tools, mainly with XGN functioning as an interchange format. From the success of other open data formats, we see that a robust, community proctored data format encourages platform neutrality and fosters innovation in the area of user interface design, hugely benefiting the end user. In the future, XGN shall focus more on the live streaming of games and multimedia features, so that XGN is set to occupy a dominant position in the internet space, coupled with emerging internet standards.

## References


[1] (1994, Mar) Portable game notation specification and implementation guide. [Online]. Available: http://www.chessclub.com/help/PGN-spec

[2] A. Cowderoy. (2001, Sep) Portable game notation specification and implementation guide - draft. [Online]. Available: http://www.enpassant.dk/chess/palview/enhancedpgn.htm

[3] ——. (2000, Apr) The chessgml distribution. [Online]. Available: http://www.saremba.de/chessgml/distribution.htm

[4] The week in chess (twic). [Online]. Available: http://www.theweekinchess.com/

[5] (2011, mar) Efficient xml interchange (exi) format. [Online]. Available: http://www.w3.org/TR/exi/

[6] (2011, oct) Epub publications 3.0. [Online]. Available: http://idpf.org/epub/30/spec/epub30-publications.html

[7] (2007, sep) Zip file format specification. [Online]. Available: http://www.pkware.com/documents/casestudies/APPNOTE.TXT

[8] *Information Processing - ISO 7-bit coded character set for information exchange*, ISO Std. 646, 1983.

[9] *Codes for the Representation of Names of Languages*, ISO Std. 639-2, 1998.

[10] *Office Open XML File Formats*, ECMA Std. 376, dec 2006. [Online]. Available: http://www.ecma-international.org/publications/standards/Ecma-376.htm

[11] M. Nicola and J. John, "Xml parsing: a threat to database performance," ser. CIKM '03. New York, NY, USA: ACM, 2003, pp. 175–178. [Online]. Available: http://doi.acm.org/10.ab45/ab6863.ab6898

[12] X. Li, H. Wang, T. Liu, and W. Li, "Key elements tracing method for parallel xml parsing in multi-core system," dec 2009, pp. 439–444.

[13] Z. Gao, Y. Pan, Y. Zhang, and K. Chili, "A high performance schema-specific xml parser," dec 2007, pp. 245–252.

[14] P. Deutsch, "Deflate compressed data format specification version 1.3," RFC 1951, may 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1951.txt

[15] F. Yergeau, "Utf-8, a transformation format of iso 10646," RFC 2279, jan 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2279.txt