

An XML-Format for Conjectures in Geometry (Work-in-Progress)

Pedro Quaresma

CISUC/Department of Mathematics, University of Coimbra
3001-454 Coimbra, Portugal, pedro@mat.uc.pt

Abstract. With a large number of software tools dedicated to the visualisation and/or demonstration of properties of geometric constructions and also with the emerging of repositories of geometric constructions, there is a strong need of linking them, and making them and their corpora, widely usable. A common setting for interoperable interactive geometry was already proposed, the i2G format, but, in this format, the conjectures and proofs counterparts are missing. A common format capable of linking all the tools in the field of geometry is missing. In this paper an extension of the i2G format is proposed, this extension is capable of describing not only the geometric constructions but also the geometric conjectures. The integration of this format into the Web-based *GeoThms*, *TGTP* and *Web Geometry Laboratory* systems is also discussed.

1 Introduction

In many dynamic geometry software tools (DGSs), a geometric construction is specified using, explicitly, a formal language. In others, the construction is made interactively, by clicking specific buttons and/or icons, but behind this approach there is also a formal geometric language, although usually hidden from the user. All these languages share many primitive commands (related to geometric constructions), but there are also differences in the set of supported commands, and they follow different syntax rules.

Another important set of tools related to geometric constructions is given by the geometry automated theorem proving software tools (GATPs). Given a geometric construction (eventually created with a given DGS) and a conjecture related to that construction, the GATPs are capable of proving or disproving (although not always) the conjecture. Some of them aim at producing traditional, human readable, geometric proofs [2,5,12].

With a large number of tools focusing on visualising geometric constructions, on proving properties of constructed objects (or both) and repositories of geometric problems (RGPs), there is an emerging need of linking them and making widely usable: constructions; conjectures and proofs generated with different tools. This would help in the progress of the field of geometric constructions, including their role in education.

The i2G format [18] was designed to describe constructions created with a DGS allowing the exchange of geometric constructions between different DGSs.

This format should be complemented in such a way that it can provide support for conjectures. The new format should be a superset of the former format, i.e. a DGS should be able to read the new format, ignoring all the extra information regarding conjectures and proofs. A GATP should be able to read the new format using, if needed, the geometric construction specification. In the following such an extension, the I2GATP format, is discussed.

Some of the most important motivating arguments for using XML in storing descriptions of geometric constructions and conjectures and as an interchange format are: strictly structured files, easy to parse, process, and convert into different forms and formats; a strict content validation of documents with respect to a given set of restrictions; easier communication and exchange of material between unrelated tools.

Paper overview. In Section 2 some background regarding DGSs, the I2G format, GATPs and RGP is given. In Section 3 the overall structure of the new format is described. In Sections 4 implementations issues are discussed. Finally in Section 5 some final conclusions are drawn and future work is discussed.

2 Background

In this section some basic background information about geometric constructions, the intergeo format (I2G), geometric conjectures and proofs and repositories of geometric problems is given.

2.1 Dynamic Geometry Software

Dynamic geometry software tools (DGSs) allow an easy construction of geometric figures built from free objects, elementary constructions and constructed objects. The dynamic nature of such tools allows its users to manipulate the positions of the free objects in such a way that the constructed objects are also changed, yet preserving the geometric properties of the construction. These manipulations are not formal proofs, as the user is considering only a finite set of concrete positions. Neither the DGS are able to provide a proof of a given conjecture nor they are able to ensure the soundness of the constructions built by its users.

There are multiple DGSs available ¹: GeoGebra, Cinderella, GeometerSketchpad, C.a.R., Cabri, GCLC to name some of the most used.

2.2 Intergeo Format

The Intergeo (I2G) file format is a specification based on the markup language XML designed to describe constructions created with a DGS. It is one of the main

¹ www.geogebra.org, www.cinderella.de, www.dynamicgeometry.com/, zirkel.sourceforge.net/, www.cabri.com/, www.emis.de/misc/software/gclc/

results of the intergeo project, an eContentplus European project dedicated to the sharing of interactive geometry constructions across boundaries. For more information about the project, visit the site <http://i2geo.net> and look into the documentation available there, as well as to [8,9].

An intergeo file takes the form of a compress file package. The main file is `intergeo.xml`, which provides a textual description of the construction in three parts, the elements part describing a (static) initial instance of the configuration, the constraints part where the geometric relationships are expressed and the display part where the details regarding the rendering of the construction are placed. For more details on the file format see [18].

There are already a significant number of DGSs supporting the i2G format (see [3] for details).

2.3 Geometry Automated Theorem Proving

The geometry automated theorem provers (GATPs) give its users the possibility to reason about a given DGS construction, this is no longer a “proof by testing”, but an actual formal proof. Another link between the GATPs and the DGSs is given by the automated deductive testing, by the GATP, of the soundness of the constructions made by the DGS [7]. Most, if not all, DGSs are capable of detecting and reporting syntactic and semantic errors, but the verification of the soundness of the construction is beyond their capabilities. If we can link DGSs and GATPs we will be able to use a given GATP in order to check the soundness of a construction created with the help of a DGS.

Automated theorem proving in geometry has two major lines of research: synthetic proof style and algebraic proof style (see [10] for a survey). Algebraic proof style methods are based on reducing geometric properties to algebraic properties expressed in terms of Cartesian coordinates. These methods are usually very efficient, but the proofs they produce do not reflect the geometric nature of the problem and they give only a yes/no conclusion. Synthetic methods attempt to automate traditional geometry proof methods producing human-readable proofs.

If the GATP is capable of producing synthetic proofs, the proof itself can be an object of study, in other cases only the conclusion matters [2,6].

2.4 Repositories of Geometric Problems

When considering repositories of geometric problems we are directly interested in a common format. If we want to provide a repository of geometric problems that can be used by DGSs and GATPs, then the constructions should be kept in a common format that can be converted to the DGS and/or GATP internal format whenever needed. The author of this paper is directly involved in this efforts having three different project that involve repositories of geometric problems.

The first (chronologically) of the mentioned projects is GeoThms², a Web-based framework for exploring geometric knowledge integrating DGSs, GATPs,

² <http://hilbert.mat.uc.pt/GeoThms/>

and a RGP. The GeoThms is a publicly accessible system with a growing body of geometric constructions and formally proven geometric theorems, its users can easily use/browse through existing geometric contents and build new contents [17]. Within this project a common, XML-based, interchange format for descriptions of geometric constructions, conjectures and proofs was developed [14]. This format predates the i2G format.

A more recent project is the Thousands of Geometric problems for geometric Theorem Provers (*TGTP*)³. This is a Web-based library of problems in geometry. *TGTP* aims, in a similar spirit of *TPPTP* and other libraries, to provide the automated reasoning in geometry community with a comprehensive and easily accessible library of GATP test problems [15]. The i2GATP format is being developed for this project. For the moment the *TGTP* system still uses the XML-based, interchange format developed for the GeoThms system (the two system share a common database), but it will change to the new format as soon as it becomes stable.

In an educational setting, the project Web Geometry Laboratory (WGL)⁴ is an asynchronous/synchronous Web environment that integrates a DGS and a RGP (and it will integrate a GATP in a next version), aiming to provide an adaptative and collaborative blended-learning environment for geometry [19]. Here the need for a common interchange format is less important, nevertheless it will be useful to allow the system to be more easily configurable, i.e. using a common format will allow choosing the DGS and/or the GATP more freely.

2.5 Integration Issues

There are already some systems integrating a DGS with one, or more, GATP and a set of examples (e.g. GCLC [4,6], GeoProof [13], JGEX [1]), but all this systems provide closed tools with a tight integration between different internal functionalities. If we want to be more generic, loosely linking DGSs, GATPs and RGPs, we need a way to establish the communication between tools as unrelated modules, i.e. we need a common format that can be used as a communication channel between tools.

3 Overall Architecture

A common format for geometric constructions, conjectures and proofs should address the communication between DGSs and GATPs, to establish the soundness, by the GATP, of a construction made with the help of the DGS or to prove (or disprove) a given conjecture about a construction made in the DGS:

- The communication between DGSs and GATPs, to establish the soundness, by the GATP, of a construction made with the help of the DGS or to prove (or disprove) a given conjecture about a construction made in the DGS.

³ <http://hilbert.mat.uc.pt/TGTP>

⁴ In prototype stage: <http://hilbert.mat.uc.pt/WebGeometryLab/>

- The rendering of the proof. If the GATP uses an algebraic method only the final result will be usable, but if the GATP uses a synthetic method, the proof itself can be an object of study.

Geometric proofs could appear in many different forms, for instance in axiomatic form (e.g., in Hilbert-style, sequent calculus style, etc.); representing higher-level proofs, produced by the area method; as algebraic proofs produced by the algebraic methods like the Gröbner basis method, etc. The representation of the proof and/or its rendering will always be linked to the method used in its development. This will be addressed in Section 3.2.

3.1 Representation of Constructions, Conjectures and Proofs

In order to enable communication between the geometric tools (i.e DGSs and GATPs) and converting files between different formats a single target format should exist: a format that could define a common normal form for the different tools. The proposal is to extend the I2G format in such a way that the new format would complement the construction description (made by a DGS) with the conjecture description. This new format will be called the I2GATP format.

Converting from a DGS/GATP language to XML, would be performed by a specific converter, naturally relying on the DGS/GATP's parsing mechanism. Converting from XML to a DGS/GATP language, will be implemented via XML-parsing tools.

Having converters from, and to, the I2GATP format for all DGSs and GATPs, we (indirectly) have converters from each tool to any other tool. Thus, in this way, the base for a common interchange format is provided. XML is a natural framework for such interchange format, because of its strict syntax, verification mechanisms, suitable usage on the Internet, and a large number of available supporting tools.

XML descriptions of constructions, conjectures and proofs can be, by means of XSLT, also rendered into other formats that are convenient for human-readable display in browsers. It can also be transformed into different representations, such as natural language form.

A specific XML scheme document could define syntactical restrictions for construction descriptions, conjectures and proofs. This document could then be used, in conjunction with the generic XML validation mechanism, for verifying whether a given file in the I2GATP format is correct (or not).

3.2 Structure of I2GATP Format

Following the ideas of the I2G common format all the files related to the I2GATP format will be packed in a single compressed file, the *container*, which is nothing more than a I2G container with three additional directories. The I2GATP format will be spread in four, at least, XML files (see Figure 1).

Apart from the `intergeo.xml` file, which is mandatory (see the I2G format specification [18]), the other files are optional.

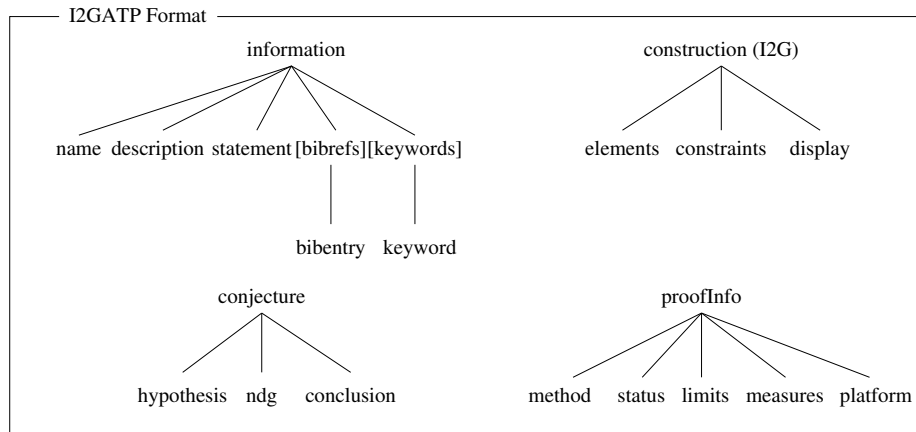


Fig. 1. Structure of the I2GATP File Format

Information The `information.xml` file contains all the generic (human) information about the problem. The *name* of the problem; a brief, informal, *description* of the problem; an informal (rigorous) mathematical description (*statement*) of the problem; a list of bibliographic references; a list of keywords.

Construction The `intergeo.xml` file contains the construction in the I2G format. The I2G format has as main tag the *construction* tag with three sub-nodes: *elements* for the free objects; *constraints* for the objects fixed by construction constraints and *display* for the display details.

Conjecture This is the core of the I2GATP format. In here the *hypothesis*, the *ndg* (non-degenerate conditions) and the *conclusion*, establishing the conjecture to be proved, are specified. The non-degenerate conditions could be a side-effect of the proving process, e.g. automatically generated by a GATP based in the area method, or provided manually.

Proofs For a given problem/conjecture we can have many proof attempts: different approaches, for instance synthetic proof versus algebraic proof; different methods, Gröbner bases method versus Wu's method; different GATPs, GCLCprover versus CoqAM, and all the possible combinations of this three different aspects.

Each proof attempt will be kept in a file `proofInfo.xml` in a sub-directory of the proofs directory (see Section 3.3 for more details).

Each individual proof node will have: the information regarding the GATP, its version and method used; the status of the proof, e.g. *proved*; the computational constraint regarding the proof attempt made by the GATP, e.g. maximum CPU time and RAM space allowed by the system; the proof metrics, e.g. number of proof steps (area method) and the platform used when doing the proof, e.g. CPU, RAM, and other details about the computational platform.

For the proof status the SZS ontology [20] will be used as a base. The “Unsolved” branch will be used as it is, the “Solved” branch has to be adapted to the I2GATP settings.

Given the fact that the proofs produced by different GATPs/Methods are, and should continue to be, quite different we do not try to create a common format for the proofs. The outcomes produced by the different GATPs will be kept as they are produced (see the *container* in Section 3.3).

3.3 The container

As said above, the I2GATP *container* is a superset of the I2G container, with three additional directories: (**information**; **construction** and **proofs**). This means that it will be possible to extract the I2G container out of this file, it will be a simple question of unpacking the file, erasing the additional directories and repacking, if needed, the resulting files.

information/	mandatory
information/information.xml	optional
construction/	mandatory
construction/intergeo.xml	mandatory
construction/preview.pdf	optional
construction/preview.svg	optional
construction/(...)	
conjecture/	mandatory
conjecture/conjecture.xml	optional
proofs/	mandatory
proofs/proof<GATP><Version><Method>/	optional
proofs/proof<GATP><Version><Method>/proofInfo.xml	optional
proofs/proof<GATP><Version><Method>/proofOutput.pdf	optional
proofs/proof<GATP><Version><Method>/(...)	
metadata/	optional
metadata/i2g-lom.xml	optional
resources/	optional
resources/<image_files>	optional
resources/(...)	
private/	optional
private/<domain-name>	optional
private/<domain-name>/<files>	optional

Table 1. The I2GATP container

The structure of the container follows closely the structure of the I2GATP format. The **information**, **construction** and **conjecture** directories will contain the files **information.xml**, **intergeo.xml** and **conjecture.xml** respectively. The directory **construction** may also contain the rendering of the construction in various graphical formats (e.g. PDF, SVG, PNG, etc.).

The directory `proofs` will contain as many sub-directories as proofs attempts were made for the problem in question. The naming convention follows the ideas in the I2G format, that is, after the prefix “proof”, the name of the GATP, its version and finally the method used. Given the fact that this is a directory identifier the strings used in these last fields should be conform to the standard naming conventions. In each of this sub-directories the file `proofInfo.xml` will contain the information regarding the proof attempt. This directory may also contain files with the rendering of the proof in different formats (e.g. PDF, HTML, etc.).

The remaining directories follow the structure of the I2G format and can be used to place additional contents produced by the GATPs.

Following the I2G conventions, the suggest naming convention to the container is `problem<problem_name>.zip`.

In the next section the symbol lists, i.e. the tags proposed to this XML-format, are described.

3.4 Symbol Lists

As said above, the container will have “four” (main) XML files: `information.xml`; `intergeo.xml`; `conjecture.xml` and as many `proofInfo.xml` files as proof attempts were made for a given problem. The `intergeo.xml` is described in the I2G common file format, technical report D3.10 [18]. The other three are specific for the I2GATP format and their symbol lists will be described in the next sections.

The symbol lists will be describe in a coarse fashion. For a more detailed account see [16].

Generic Information (`information.xml`) Generic information about the problem. All fields, except the *name*, may be empty.

The tags are: *name*; *description*; *statement*; *bibrefs* and *bibentry*; *keywords* and *keyword*.

The *description* will be a brief, informal, description of the problem in text format and the *statement* will be an informal (rigorous) mathematical description of the problem in MATHML [11].

The *bibrefs* is a list (it may be empty) of bibliographic references in BIBTEXML format⁵.

The contents of the *description* and *bibrefs* tags could be automatically converted from L^AT_EX and BIBTEX using, for example, *tex4ht*⁶ and BIBTEXML converters respectively.

The *keywords* is a list of keywords in text format. For the moment this field is a free-form text field. For better querying the repositories, an index or a geometric ontology should be considered. Maybe an “open classification”, that

⁵ <http://bibtexml.sourceforge.net/>

⁶ <http://tug.org/applications/tex4ht/>

is, a classification index open to users additions and where the most chosen keywords became, in time, fixed.

Conjecture Information `conjecture.xml` The main tags are: *conjecture*; *hypothesis*, *ndg* (for non-degenerate conditions) and *conclusion*. The three last tags can contain a large number of other tags used to write down the geometric (logical) statements.

Without pretending to be exhaustive we have: *not_equal*; *not_parallel*; *equal*; *plus*; *mult*; *collinear*; *perpendicular*; *parallel*; *midpoint*; *same_length*; *harmonic*; *segment_ratio*. The symbols of the intergeo format regarding the geometric construction can occur here.

Proofs Information `proofInfo.xml` Contains all the information regarding a proof attempt for given problem.

This is a record of the conditions under which the proof was attempted, i.e. the method used (*method*), the limits imposed to the GATP and the computer system used (*limits* and *platform*). Adding to this the proof outcome, i.e. proved, not proved, etc. and also, measures of efficiency, e.g. CPU time used, number of steps, etc. (*status* and *measures*).

In the list of symbols we have (among others): *status*; *limits*; *time_limit_seconds*; *iterations_limit*; *measures*; *CPU_time*; *elimination_steps*; *number_terms_largest_polynomial*; *computer_name*; *clock_speed*; *RAM*; *operating_system*.

4 Implementation

Having defined a XML format for geometric constructions and conjectures its usefulness depends on its support from other tools, i.e. the capability of tools such as DGSs (see [3] to the list of tools already supporting the I2G format) and GATPs to export to the I2GATP format and, of course, its support to other tools in the shape of converters from I2GATP format to the internal format of tools such as the DGSs and GATPs (see Figure 2).

Using the *TGTP* project as a catalyst for this task I will try to provide (working in conjunction with the authors of the tools):

- Converters from dynamic DGSs and GATPs tools (GCL language, Coq AM, etc.) to I2GATP format.
- Converters from I2GATP format to DGSs and GATPs tools (GCL language, Coq AM, etc.).

The I2GATPformat will be backwards compatible with I2G format. DGSs should be able to read the I2GATP container ignoring the extra info. The GATPs should also be able to read the I2G format, adding information whenever needed.

The *TGTP* and *GeoThms* servers will use the I2GATP as its base format, providing converters to and from the different GATPs.

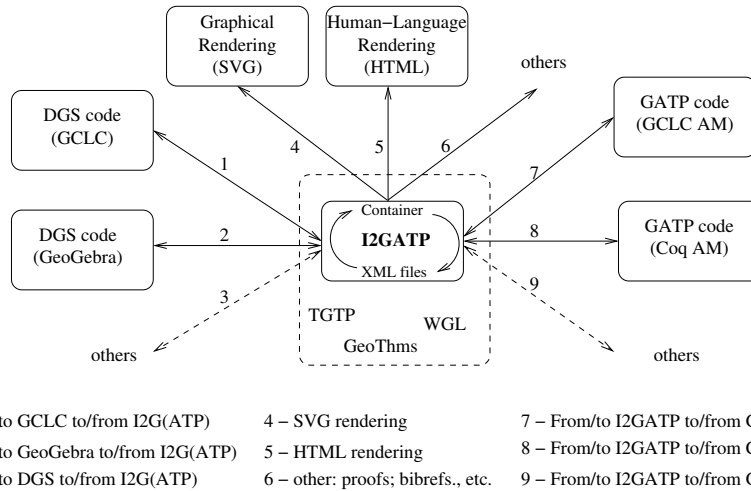


Fig. 2. Conversions From/To i2GATP To/From Geometric Tools

5 Conclusions and Further Work

A case for extending the i2G XML format to the description of geometric conjectures and as an interchange format for dynamic geometry software and geometry automated theorem proving tools was presented.

A brief description of the i2G format and the tools using it and also the tools that can benefit from the extended format was given. The overall architecture and physical organisation of the i2GATP format was described. Arguments justifying the usefulness of this extended format were discussed.

The work presented in this paper is related to work in other domains of automated reasoning where joint efforts of numerous researchers led to standards and libraries which are very fruitful for easier exchange of problems, proofs, and even program code, contributing to the advance of the underlying field (see [15]).

This is a work-in-progress. Questions and future work to be addressed:

- The XML format must be complemented with an extensive set of converters allowing the exchange of information between as many geometric tools as possible.
- The databases queries, as in *TGTP*, raise the question of selecting appropriate keywords. A fine grain index and/or an appropriate geometry ontology should be addressed.
- The i2GATP format does not address proofs. Should we try to create such a format? The GATPs produce proofs in quite different formats, maybe the construction of such unifying format it is not possible and/or desirable in this area.

The i2GATP format will allow to further extend the database of geometric constructions within *GeoThms* and *TGTP* and, hopefully lead then to a ma-

for public resource for geometric constructions, linking a significant number of geometry tools under this new format.

References

1. Shang-Ching Chou, Xiao-Shan Gao, and Zheng Ye. Java geometry expert. <http://www.cs.wichita.edu/~ye/>, 2004.
2. Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants, I. multiple and shortest proof generation. *Journal of Automated Reasoning*, 17:325–347, 1996.
3. The Intergeo Consortium. Intergeo implementation table. <http://i2geo.net/xwiki/bin/view/I2GFormat/ImplementationsTable>.
4. Predrag Janičić. GCLC a tool for constructive euclidean geometry and more than that. In Andrés Iglesias and Nobuki Takayama, editors, *Mathematical Software - ICMS 2006*, volume 4151 of *Lecture Notes in Computer Science*, pages 58–73. Springer Berlin / Heidelberg, 2006.
5. Predrag Janičić, Julien Narboux, and Pedro Quaresma. The Area Method: a recapitulation. *Journal of Automated Reasoning*, 48(4):489–532, 2012.
6. Predrag Janičić and Pedro Quaresma. System description: GCLCprover + GeoThms. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 145–150. Springer Berlin / Heidelberg, 2006.
7. Predrag Janičić and Pedro Quaresma. Automatic verification of regular constructions in dynamic geometry systems. In Francisco Botana and Tomás Recio, editors, *Automated Deduction in Geometry*, volume 4869 of *Lecture Notes in Computer Science*, pages 39–51. Springer Berlin / Heidelberg, 2007.
8. U. Kortenkamp, A. M. Blessing, C. Dohrmann, Y. Kreis, P. Libbrecht, and C. Mercat. Interoperable Interactive Geometry for Europe First technological and educational results and future challenges of the Intergeo Project. In *CERME 6*, 2006.
9. U. Kortenkamp, C. Dohrmann, Y. Kreis, C. Dording, P. Libbrecht, and C. Mercat. Using the Intergeo platform for teaching and research. In *Proceedings of the 9th International Conference on Technology in Mathematics Teaching (ICTMT-9)*, 2009.
10. Noboru Matsuda and Kurt Vanlehn. Gramy: A geometry theorem prover capable of construction. *Journal of Automated Reasoning*, 32:3–33, 2004.
11. W3C Math Working Group members. *Mathematical Markup Language (MathML) Version 3.0*. W3C, October 2010.
12. Julien Narboux. A decision procedure for geometry in Coq. *Lecture Notes in Computer Science*, 3223:225–240, 2004.
13. Julien Narboux. A graphical user interface for formal proofs in geometry. *Journal of Automated Reasoning*, 39:161–180, 2007.
14. P. Quaresma, Tomašević J. Janičić, P., M. V.-Janičić, and D. Tošić. *Communicating Mathematics in The Digital Era*, chapter XML-Bases Format for Descriptions of Geometric Constructions and Proofs, pages 183–197. A. K. Peters, Ltd., 2008.
15. Pedro Quaresma. Thousands of geometric problems for geometric theorem provers (tgtp). In Pascal Schreck, Julien Narboux, and Jrgen Richter-Gebert, editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*, pages 169–181. Springer Berlin / Heidelberg, 2011.

16. Pedro Quaresma. The i2GATP format. Technical report, CISUC, 2012. (<http://hilbert.mat.uc.pt/TGTP/Documents/Docs/cisucTri2gatp.pdf>).
17. Pedro Quaresma and Predrag Janičić. GeoThms – a Web System for euclidean constructive geometry. *Electronic Notes in Theoretical Computer Science*, 174(2):35 – 48, 2007.
18. E. Santiago, Maxim Hendriks, Yves Kreis, Ulrich Kortenkamp, and Daniel Marquès. i2G Common File Format Final Version. Technical Report D3.10, The Intergeo Consortium, 2010.
19. Vanda Santos and Pedro Quaresma. Integrating DGSs and GATPs in an adaptative and collaborative blended-learning Web-environment. In *First Workshop on CTP Components for Educational Software (THedu'11)*, volume 79 of *EPTCS*, 2012.
20. Geoff Sutcliffe. The SZS ontologies for automated reasoning software. In P Rudnicki, G. Sutcliffe, B. Konev, R. Schmidt, and S. Schulz, editors, *Proceedings of the Combined KEAPPA - IWIL Workshops*, pages 38–49, 2008.