The SymbolicData Project – Towards a Computer Algebra Social Network

Hans-Gert Gräbe, Andreas Nareike, and Simon Johanning

Universität Leipzig, Germany
(graebe|nareike|johanning)@informatik.uni-leipzig.de

Abstract. We report about a complete redesign of the tools and data of the SymbolicData project according to RDF technologies and Linked Data principles that proved to be powerful within modern semantic web approaches. During that redesign the focus of the project changed from a mere data store towards the vision of a Computer Algebra Social Network (CASN) to support technically intercommunity communication between Computer Algebra subcommunities. In the last part of the paper we describe ongoing efforts to implement a technical basis for a Distributed Semantic Social Network infrastructure to run such a CASN.

1 Introduction

The SymbolicData project grew up from the Special Session on Benchmarking at the 1998 ISSAC conference to continue the efforts started by the PoSSo [11] an FRISCO [5] projects. It aimed at building a reliable and sustainably available collection of Polynomial Systems that were reported in the literature for benchmarking and profiling of CA software, to extend and update it, to collect meta information about the records, and also to develop tools to manage the data and to set up and run reliable tests and benchmark computations on the data. A first prototype was developed during 1999–2002 by Olaf Bachmann and Hans-Gert Gräbe with data from *Polynomial Systems Solving* and *Geometry Theorem Proving*.

There was almost no advance during 2002–2005. In a second phase around 2006 the project matured again and extended its scope. Data was supplied by the CoCoA group (F. Cioffi), the Singular group (M. Dengel, M. Brickenstein, S. Steidel, M. Wenk), V. Levandovskyy (non commutative polynomial systems, G-Algebras) and R. Hemmecke (Test sets from Integer Programming). In 2005 the German Fachgruppe Computeralgebra launched the Web site `http://www.symbolicdata.org`. During the Special Semester on Gröbner Bases (GB) in March 2006 we tried to join forces with the GB-Bibliography project (B. Buchberger, A. Zapletal) and the GB-Facilities project (V. Levandovskyy).

In 2009 we started to refactor the data along standard Semantic Web concepts based on the Resource Description Framework (RDF). We completed a redesign of the data along RDF based semantic technologies, set up a Virtuoso [20] based RDF triple store and SPARQL endpoint at `http://www.symbolicdata.org` as

an Open Data service along Linked Data standards [7], and started both conceptual and practical work towards a semantic-aware Computer Algebra Social Network. The new SymbolicData data and tools were released as version 3 in September 2013.

One of the main decisions within that redesign process was a non-technical one – leave the focus on data storage and the "roots" within the Polynomial Systems Solving CA subcommunity in favour of stronger social interlinking. We reshaped the SymbolicData Project as *intercommunity project*, that addresses needs of subcommunities within the Symbolic Computation community to profile, test and benchmark implementations as a *cross cutting activity*[1], and started to develop links to other intercommunity activities as *sagemath* [13], *lmonade* [8] or *swmath* [16].

In section 2 and 3 we describe the current SymbolicData infrastructure in more detail. The rest of this paper addresses conceptual and practical problems, experiences and solutions towards a semantic-aware Computer Algebra Social Network as an intercommunity project.

Acknowledgement: This and ongoing work was realised within a project *Benchmarking in Symbolic Computations and Web 3.0* supported by the *Saxonian E-Science Initiative* [3] with a 12 months grant for Andreas Nareike in 2012/13 and a five months grant for Simon Johanning in 2014.

2 The SymbolicData Infrastructure

Our basic resources (examples for testing, profiling and benchmarking software and algorithms from different areas of symbolic computation) are publicly available in XML markup, meta data in RDF notation both from a public git repo, hosted at `http://github.org/symbolicdata`, and from our remote RDF triple store at `http://symbolicdata.org/Data`. Moreover, we offer a SPARQL endpoint [18] to explore the data by standard Linked Data methods.

The website operates on a standardized installation using an Apache web server to deliver the data, the Virtuoso RDF data store [20] as data backend, a SPARQL endpoint and (optionally) OntoWiki [10] to explore, display and edit the data. This installation can easily be rolled out on a local site[2] to support local testing, profiling and benchmarking.

The distribution contains also tools and prototypical solutions for a local compute environment as, e.g., provided by Sagemath [13]. The Python based *SDEval package* [6] by Albert Heinle offers a JUnit like framework to set up, run, log, monitor and interrupt testing and benchmarking computations. The *SDSage package* [9] by Andreas Nareike provides a showcase for SymbolicData integration with the Sagemath [13] compute environment.

[1] See `http://en.wikipedia.org/wiki/Cross-cutting_concern`.

[2] Tested with Linux Debian and Ubuntu 12.04 LTS standard distributions; a more detailed description can be found in the SymbolicData wiki [17].

We follow a development process along the Integration-Manager-Workflow Model[3]. This makes it easy to join forces with the SymbolicData team: Fork the repo to your github account, start development and send a pull request to the Integration Manager if you think you produced something worth to be integrated into the upstream master branch. Even if your contribution is not pulled to the upstream, people can use it, since they can pull it from your github repo to their github repo. This allows even for agile common small feature development – a widely practised way to advance projects hosted at `github.com`. You are encouraged to start a discussion about your plans early in the process and regularly report your progress on the SymbolicData mailing list.

Currently the SymbolicData data collection contains resources from Polynomial Systems Solving (390 records, 633 configurations), Free Algebras (83 records), G-Algebras (8 records), GeoProofSchemes (297 records) and Test Sets from Integer Programming (28 records). These resources are stored in a flat XSchema based XML syntax developed within SymbolicData version 2 that uses well established intracommunity syntaxes for the internal data.

3 Towards a Decentralized Infrastructure of Resources

Note that RDF provides a strong conceptual distinction between *resources* (basic information) and *resource descriptions* (meta information) and with SymbolicData version 3 we use XML representations more concisely to focus on the basic information structure itself.

Since the basic information is provided by different CA subcommunities it is a good advice to use the (textual) syntactical notations well established within a subcommunity to store data. In most cases such syntactical notations are not XML based, so one cannot use a standard XML parser to parse the internal textual representations "out of the box".

Since the subcommunity has plenty of parsers and tools at hand to input textual representations into their *semantic-aware tools* this is not a real obstacle in practise. In particular, in the early times of SymbolicData we had a dispute about representation of polynomials – use the well established operator syntax as, e.g., `x^3+5*x-2`, or provide polynomials in XML-based OpenMath or MathML syntax. We decided to store polynomials in the compact human readable operator syntax, as in the PoSSo project.

In the SymbolicData data structure concept we use XML markup mainly to compile heterogeneously structured data into a single resource in such a way that the different parts of this data can be extracted by a standard XML parser and passed to appropriate semantic-aware tools for further processing. To start a new data collection within SymbolicData you have to decide about the parts of data that have to be bundled for a single resource, to decide about the syntactical representation of these parts according to the standards of your scientific subcommunity, to develop a XSchema based XML representation for the bundling,

[3] See `http://git-scm.com/book/en/Distributed-Git-Distributed-Workflows`.

and provide data along that standard. The main point about the resources is the reliable and sustainable availability of the data through a *permanent web address*, a *Unique Resource Identifier* (URI). We provide access to our centrally managed resources via `http://symbolicdata.org/XMLResources/`.

Such a concept is not restricted to centrally managed resources, but can easily be extended to other data stores on the web that are operated by different CA subcommunities and offer a minimum of Linked Data facilities. There are draft versions of resource descriptions about Fano Polytopes (8630 records) and Birkhoff Polytopes (5399 records) hosted by Andreas Paffenholz and about Transitive Groups (3605 records) from the Database for Number Fields of Jürgen Klüners and Gunter Malle that point to such external resources.

4 Resources and Resource Descriptions

Preparing SymbolicData version 3 we decided to strengthen the part of intercommunity communication aspects. From this point of view *resources* are owned and maintained by different CA subcommunities, and meta data or *resource descriptions* are important for technically supported interchange of data between such subcommunities and for intercommunity communication, and hence should be managed and maintained within a cooperative intercommunity process.

A first question to be solved was about data representation for resources and resource descriptions. XML based design principles mainly distinguish between information (XML records) and information structure (described with XSchema) and are well suited for data representation of (basic) resources but proved to be not expressive enough to represent interrelations between different resources in a flexible way.

4.1 Why RDF?

We decided to switch to RDF as basic representation for resource descriptions by several reasons. First, we could join forces with the Agile Knowledge Engineering and Semantic Web (AKSW) Group at Leipzig University[4], a leading research group in semantic technologies, and exploit their experience about concepts and tools in that area. Second, RDF gets established more and more for exchange of meta information not only within the Linked Open Data world [7], but also for the big projects on standardization of scientific communication as the Dublin Core DCMI Metadata Terms Initiative [2] or the Joint Steering Committee for Development of Resource Description and Access [12]. Third, there are well elaborated concepts and tools how to exchange RDF based information by a protocol as simple and widely spread as HTTP Get and how to manage that within a standard web server infrastructure.

[4] See `http://aksw.org/About.html`.

4.2 RDF Basics

RDF – the Resource Description Framework – is about *description* of *resources*, represented by (globally unique) *resource identifiers* (URIs). RDF provides a unified scheme to represent relational information as *triples*. There are several notational standards (ntriples, turtle, rdf/xml, json) for triples and plenty of tools to manage sets of triples, i.e., *RDF graphs*.

Each such triple can be considered as a *sentence* of a story that consists of a subject s, a predicate p and an object o, but different to real stories the semantics of an RDF graph is that of a *set*, i.e., the order of the sentences does not matter. Hence the expressiveness of RDF stories is very restricted compared to natural languages. The main advantage however, is a separation between data and search algorithms on data patterns as in rule based programming. RDF comes with the standardized pattern based query language SPARQL to operate such search queries on RDF data stores. For SymbolicData we use Virtuoso [20] as RDF data store and SPARQL endpoint.

RDF has another advantage compared to classical database approaches – one can express *descriptions of descriptions*, i.e., database design, within the same language concepts, and thus share not only descriptions of data but also descriptions of data descriptions, i.e., information about the semantics of the data in a machine readable way.

Subjects and predicates have to be URIs while objects (or 'values') can be either URIs or (plain or typed) literals in lexical form (a string included in quotes). There are some predefined common types (e.g., `xsd:integer`) but custom types can be defined as well.

A set of triples can be interpreted as a directed graph (RDF graph) with subjects and objects as nodes (replacing literals by labelled blank nodes) and predicates as labelled edges between nodes. On the opposite, a directed graph can be written as a set of triples (and is commonly represented in such a way as internal data structure of graph programs). Another representation uses sets of key-value pairs $p \to o$ assigned to the different subjects s. Note that, different to database columns, a key p can have multiple values.

RDF uses some more basic concepts – *character sets* to compose URIs and literals and *name spaces* to structure information spaces and to resolve conflicts within URI creation. There are more elaborated concepts as OWL, RDFS etc. on top of RDF as explained in the *Semantic Web Stack* [15], but not yet used within SymbolicData. Note that nowadays the syntax layer below the RDF data interchange layer in the Semantic Web Stack is no more bound solely to XML as [15] might suggest – the most widespread syntax representation is in Turtle format[5].

4.3 Linked Data Principles

The real power of RDF does not originate in an alleged superiority of concepts but in the *practical availability* of data stores all over the world that are orga-

[5] See `http://en.wikipedia.org/wiki/Turtle_(syntax)`.

nized on RDF based principles. Whereas the (traditional) Web 2.0 is build upon interlinked dynamical HTML web pages based on private databases, the Semantic Web as part of Web 3.0 [14] focuses on interlinking these private databases themselves into a single big distributed data store.

RDF supports both ways of data dissemination – by file transfer as in Web 2.0 and by remote access to RDF triple stores as in Web 3.0 –, and so does SymbolicData: You can download whole RDF graphs as data files from our remote host, upload this data into your local data store and process it locally, but you can also directly access our remote RDF data store. RDF data stores operate on the HTTP protocol and hence are best deployed within a webserver infrastructure, either remote or local. The only difference between remote and local approaches are the stronger web security requirements for a remote location.

To achieve web access of RDF data on a remote host, URIs should be available as URLs, i.e., a *HTTP Get* request to an URI should deliver a valuable portion of RDF information about that subject. This is the core of the Linked Data Principle [7] and realised for the `http://symbolicdata.org/Data/` name space within the SymbolicData project.

5 SymbolicData Resource Descriptions

RDF resource descriptions are the main part of the meta information collected within the SymbolicData project. We offer resource descriptions for several purposes – resource fingerprints to navigate within the examples, relational information to, e.g., bibliographical references or CA software descriptions, and information about activities of people involved with CA research.

5.1 Resource Fingerprints

Semantically equivalent data usually can be given in different syntactical form. For example, the same Polynomial System can be given with different variable names, in different polynomial orderings and even in different forms as, e.g., expanded or factorized polynomials.

To navigate within such data, to prestructure data for efficient search or to identify a given example within the database it is helpful to precompile *fingerprints*, i.e., (semantically sound) invariants of the different examples. For example, the set of degree lists (in standard grading) or the set of the lengths of polynomials in distributive normal form are such invariants for Polynomial Systems.

Examples with different fingerprints are surely different, examples with the same fingerprint require more elaborated methods to be distinguished. In most cases the latter is not worth to be automated since the "general nonsense" knowledge of the experts (optionally added as `rdfs:comment` to the resource description) is a more powerful "tool" to resolve such disambiguities.

The computation of fingerprints requires semantic-aware tools and both the definition of useful fingerprints and its computation are due to the CA subcommunity experts with the appropriate semantic knowledge and tools. To compare user given examples with existing ones it is a good advice to have enough invariants as fingerprints at hand that can be computed in polynomial time.

5.2 Relational Information

It was one of the great visions of the SymbolicData Project to collect not only benchmark and testing data but also valuable background information about the records in the database as, e.g., information about papers, people, history, systems etc. concerned with the examples in our collection. It was the main target of SymbolicData version 3 to redesign these data along RDF principles.

We provide a general concept of an RDF class `sd:Annotation` to store background information in a unified way. Instances of this class have predicates

- `rdfs:label` – a label,
- `rdfs:comment` – a text field for annotation,
- `sd:relatesTo` – a set of related URIs.

We use that concept in particular to relate *bibliographical information* of type `sd:Reference` to different data records. The management of bibliographical references was completely redesigned with SymbolicData version 3 exploiting RDF and the established Dublin Core ontology [2] to represent bibliographical information in a way that is queryable by standard means and tools. On the other hand, we strongly reduced the part of information about bibliographical references kept inside SymbolicData since there are comprehensive bibliographical stores available on the web that provide all required information via permanent URIs, although in most cases not yet in RDF format. At the moment we provide links to three such bibliographical stores,

- the database of Zentralblatt Mathematik (predicate `sd:hasZBentry`),
- the Gröbner Bases Bibliography database (predicate `sd:hasGBBentry`) and
- the CiteSeer database (predicate `sd:hasCSentry`).

The same applies to information about and references to CA software that is SymbolicData-internally stored as resource description of type `sd:CAS` but points as far as possible to the relevant information within the *swmath* database [16], even if *swmath* does not (yet) operate by Linked Open Data standards.

5.3 Publicly Tracking Personal Profiles

Bibliographical references, references to CA software and even references about contributions to SymbolicData itself refer to people involved with CA research. It is one of the challenges of big data stores about scientific publications to find out all publications of a given author, since the same author may be listed in different ways in the author list of different publications. The big identification

projects use elaborated evaluation algorithms of cross references to solve this problem or – as the *Zentralblatt Mathematik* did for a long time – use simple string pattern matching. Some time ago the *Zentralblatt* started a certain kind of tracking of personal profiles[6] to improve that alignment.

We argue that it is a good advise for scientific communities to support such tracking activities since the benefits much exceed the drawbacks. Moreover, active involvement of scientific communities allows to "track the trackers", i.e., to start open discussions and to influence actively the settings of the tracking process to maximize its benefits and minimize its drawbacks.

The `sd:Person` database (274 records) supports that process of disambiguation on the level of references and authors evaluating different sources of information about CA publications and relating authorship to `sd:Person` URIs that are composed following well defined naming rules. This part of the project is under heavy development with focus on activities within the German Fachgruppe.

6 Towards a Computer Algebra Social Network

From the five stars to be assigned to a Linked Data project according to Tim Berners-Lee's classification [1] SymbolicData earned four stars so far (for offering data in interoperable RDF format on the web and providing a SPARQL querable RDF triple store). For the fifth star one has to build up stable semantic relations to foreign knowledge bases and thus become part of the Linked Open Data Cloud [7].

Much of such interrelation, e.g., a list of interoperability references for people, software and bibliographical data with *Zentralblatt*, is on the way. Moreover, we joined forces with the efforts of the board of the German Fachgruppe to store and provide information about people and groups working on CA topics at their new Wordpress driven web site [4]. We developed a first prototype to store this information in RDF format, to extract it by means of SPARQL queries and to view it on the web site using the Wordpress shortcode mechanism[7] via a special Wordpress plugin. We apply the same technique to maintain information about upcoming conferences, CA projects within the SPP 1489 priority program and a list of dissertations in CA at this site.

The vision of a Computer Algebra Social Network (CASN) goes far beyond that: Get people involved themselves on a regular basis, set up and run within the CA community a semantic-aware Facebook like Social Network and contribute to it about all topics around Computer Algebra using tools that express your contributions in an RDF based vocabulary that the community agreed upon. This sounds quite visionary but is in no way utopic. We operate a first prototypical node of a tool that realizes the challenging concept of a *Distributed Semantic Social Network* (DSSN) [19].

[6] See, e.g., the entry `https://zbmath.org/authors/?q=ai:grabe.hans-gert` of the first author of this paper.

[7] See `http://codex.wordpress.org/Shortcode`.

We set up a second RDF data store at `http://symbolicdata.org/casn/` with information about

- upcoming conferences (about 20 entries of type `sd:Event`),
- publications within the "CA Rundbrief" of the German Fachgruppe,
- dissertations in CA reported to the board of the German Fachgruppe,
- CA projects (initial: projects from the German SPP 1489 priority program)
- and CA working groups (initial: as listed by the German Fachgruppe).

As the project matures this will be interrelated with the DSSN node at `http://symbolicdata.org/xodx/` running a software under development by the AKSW group in such a way that you can join the CASN and supply your contributions as you can do (also not yet semantically) in any other social network. We refer to our wiki [17] for more information.

References

1. Berners-Lee, T.: 5 stars for Open Data. `http://5stardata.info/` [2014-03-05]
2. DCMI Metadata Terms. `http://dublincore.org/documents/dcmi-terms/` [2014-02-27]
3. Das eScience-Forschungsnetzwerk Sachsen. `http://www.escience-sachsen.de` [2014-02-19]
4. Website of the German Fachgruppe Computeralgebra. `http://www.fachgruppe-computeralgebra.de/` [2014-03-06]
5. FRISCO – A Framework for Integrated Symbolic/Numeric Computation. `http://www.nag.co.uk/projects/FRISCO.html` [2014-02-19]
6. Heinle, A.: The SDEval framework. `http://symbolicdata.org/wiki/SDEval` [2014-02-28]
7. Linked Data. `http://www.w3.org/DesignIssues/LinkedData.html` [2014-05-30]
8. lmonade – a platform for development and distribution of scientific software. `http://www.lmona.de/` [2014-05-24]
9. Nareike, A.: The SDSage package. `http://symbolicdata.org/wiki/PolynomialSystems.Sage` [2014-02-28]
10. Auer, S., Dietzold, S., Lehmann, J., Riechert, T.: OntoWiki: A Tool for Social, Semantic Collaboration. Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge CKC (2007)
    See `http://ceur-ws.org/Vol-273/paper_91.pdf` [2014-05-30]
    See also `http://aksw.org/Projects/OntoWiki.html` [2014-02-19]
11. The PoSSo Project. `http://posso.dm.unipi.it/` [2014-02-19]
12. Joint Steering Committee for Development of Resource Description and Access.
    See `http://www.rda-jsc.org/rda-new-org.html` or `http://www.dnb.de/DE/Standardisierung/International/rdaFaq.html`.
13. Sage – a free open-source mathematics software system. `http://www.sagemath.org` [2014-02-19]
14. The Semantic Web. `http://en.wikipedia.org/wiki/Semantic_Web` [2014-05-24]
15. The Semantic Web Stack. `http://en.wikipedia.org/wiki/Semantic_Web_Stack` [2014-05-24]
16. swMATH – an information service for mathematical software. `http://www.swmath.org` [2014-05-24]

17. The SymbolicData project wiki. http://symbolicdata.org/wiki
18. The SymbolicData SPARQL endpoint.
http://symbolicdata.org:8890/sparql
19. Tramp, S. et al.: DSSN: towards a global Distributed Semantic Social Network.
http://aksw.org/Projects/DSSN.html [2014-03-06]
20. Virtuoso Open-Source Edition. http://virtuoso.openlinksw.com/ [2014-02-19]