Symbolic Computation Software Composability Protocol and its implementations

The SCIEnce project

http://www.symbolic-computation.org
support@symbolic-computation.org

What is SCIEnce?

The EU Framework 6 SCIEnce project "SCIEnce – Symbolic Computation Infrastructure in Europe" (http://www.symbolic-computation.org) is a major 5-year project that brings together nine partners from seven countries and two continents, namely University of St Andrews (St Andrews), Research Institute for Symbolic Computation (Linz), Centre National de la Recherche Scientifique (France), Universität Kassel, Technische Universiteit Eindhoven, Technische Universität Berlin, Institute e-Austria Timisoara, Heriot Watt University, (Edinburgh) and Maplesoft (Waterloo, Canada) representing CAS developers and experts in computational algebra, OpenMath, and parallel computations.

What is SCSCP?

One of the aims of the SCIEnce project is to design a common standard interface that may be used for combining computer algebra systems (and any other compatible software). We have developed a simple lightweight XML-based remote procedure call protocol called **SCSCP** (*Symbolic Computation Software Composability Protocol*) in which both data and instructions are represented as OpenMath objects. It may be used both to link systems directly with each other, and also as the foundation for more advanced cluster and grid infrastructures. The advantage of this approach is that any system that implements SCSCP can immediately connect to all other systems that already support it. This avoids the need for special cases and minimizes repeated effort. In addition, SCSCP allows remote objects to be handled by reference so that clients may work with objects of a type that do not exist in their own system at all.

The SCSCP protocol (currently at version 1.3) is socket-based. It uses port number 26133, as assigned by the *Internet Assigned Numbers Authority* (IANA), and XML-format messages. The protocol is accompanied by two OpenMath content dictionaries, `scscp1` and `scscp2` (see [2, 3]). SCSCP specification is available in [5], and some additional details with examples of SCSCP communication may be found in [4, 6, 12]

Which systems currently support SCSCP?

In the SCIEnce project, the following extensions for computer algebra systems have been implemented:

- GAP SCSCP client and server, via packages SCSCP [13] and openmath [1];
- KANT SCSCP client and server [7];
- Maple SCSCP client and server (currently only for the development version);
- OpenMath package and SCSCP client for MuPAD [9].

Besides that, third-party implementations appeared for the following systems:

- Macaulay2 (`http://www.math.uiuc.edu/Macaulay2/`);
- Magma (as a wrapper application);
- TRIP (`http://www.imcce.fr/Equipes/ASD/trip/`).

Which SCSCP middleware is available?

The SCIence project has produced a Java library *org.symcomp.scscp* [10] that acts as a reference implementation for systems developers who would like to implement SCSCP for their own systems. This is freely available under the Apache2 license. The library homepage also contains the middleware developed with its usage, namely:

- Convertor between the OpenMath and Popcorn notation (see [12]);
- WebProxy, intended to offer a uniform, SOAP-Compliant access to a one or more (possibly distinct) CASes;
- WUPSI, a command line SCSCP-compliant interface (see [11]).

Additionally, there is also a third-party SCSCP API for C/C++ available at [8].

How to install systems and middleware?

To install computer algebra systems extensions and middleware listed above, follow links from the appropriate references or from the SCIEnce homepage `http://www.symbolic-computation.org/`.

Installing the GAP system, note that SCSCP and OpenMath packages are also included in the composite archive of GAP packages, available from the GAP homepage `http://www.gap-system.org/`. Note that packages IO and GAPDoc should also be installed, and you need to build the binaries for the IO package following instructions in its readme file.

What we are going to demonstrate at ISSAC 2010?

The software demo is intended to serve as a complement to the presentation of the paper *Easy Composition of Symbolic Computation Software: A New Lingua Franca for Symbolic Computation* by S. Linton, K. Hammond, A. Konovalov, A. D. Al Zain, P. Trinder, P. Horn and D. Roozemond, accepted at ISSAC 2010. We are going to give a brief overview of the range of SCSCP-compliant products, and then demonstrate some examples of communication between systems using combinations of software and middleware different from those presented in the ISSAC 2010 paper, in which we provided only some examples to stay within the page limit. An emphasise will be done on the Java library *org.symcomp.scscp*, and GAP will serve as a CAS back-end for this demonstration.

References

1. M. Costantini, A. Konovalov, and A. Solomon. *OpenMath – OpenMath functionality in GAP, Version 10.1*, 2010. GAP package `http://www.cs.st-andrews.ac.uk/~alexk/openmath.htm`.
2. S. Freundt, P. Horn, A. Konovalov, S. Lesseni, S. Linton, D. Roozemond. *OpenMath Content Dictionary: scscp1.* `http://www.win.tue.nl/SCIEnce/cds/scscp1.html`.
3. S. Freundt, P. Horn, A. Konovalov, S. Lesseni, S. Linton, D. Roozemond. *OpenMath Content Dictionary: scscp2.* `http://www.win.tue.nl/SCIEnce/cds/scscp2.html`.
4. S. Freundt, P. Horn, A. Konovalov, S. Lesseni, S. Linton, D. Roozemond. OpenMath in SCIEnce: Evolving of symbolic computation interaction. In proceedings of OpenMath Workshop 2009 (to appear).
5. S. Freundt, P. Horn, A. Konovalov, S. Linton, and D. Roozemond. Symbolic Computation Software Composability Protocol (SCSCP) specification. `http://www.symbolic-computation.org/scscp`, Version 1.3, 2009.
6. S. Freundt, P. Horn, A. Konovalov, S. Linton, and D. Roozemond. Symbolic computation software composability. In *AISC/MKM/Calculemus, Springer LNCS 5144*, pages 285–295, 2008.
7. S. Freundt and S. Lesseni. *KANT 4 SCSCP Package.* `http://www.math.tu-berlin.de/~kant/kantscscp.html`.
8. M. Gastineau. *SCSCP C Library – A C/C++ library for Symbolic Computation Software Composibility Protocol, Version 0.6.0.* IMCCE, 2009. `http://www.imcce.fr/Equipes/ASD/trip/scscp/`.
9. P. Horn. *MuPAD OpenMath Package*, 2009. `http://mupad.symcomp.org/`.
10. P. Horn and D. Roozemond. *java.symcomp.org – Java Library for SCSCP and OpenMath.* `http://java.symcomp.org/`.
11. P. Horn and D. Roozemond. *WUPSI –Universal Popcorn SCSCP Interface.* `http://java.symcomp.org/wupsi.html`.
12. P. Horn and D. Roozemond. OpenMath in SCIEnce: SCSCP and POPCORN. In *Intelligent Computer Mathematics – MKM 2009*, volume 5625 of *Lecture Notes in Artificial intelligence*, pages 474–479. Springer, 2009.
13. A. Konovalov and S. Linton. *SCSCP – Symbolic Computation Software Composability Protocol, Version 1.2*, 2010. GAP package `http://www.cs.st-andrews.ac.uk/~alexk/scscp.htm`.