Symbolic **C**omputation **S**oftware **C**omposability **P**rotocol in **GAP**

Alexander Konovalov
(joint work with Steve Linton)

*Supported by the EU FP6 project "**SCIEnce** – **S**ymbolic **C**omputation **I**nfrastructure for **E**urope"*

School of Computer Science and Centre for Interdisciplinary Research in
Computational Algebra, University of St Andrews, Scotland

GAP Packages Authors Workshop, Braunschweig, September 11-15, 2007

SCIENCE
Symbolic
Computation
Infrastructure for
Europe

Modern needs of symbolic computations

- Efficient tools for combining different computational algebra systems to solve complex problems that require capabilities not available in any single system
- Web services client and server interfaces allowing deployment of computer algebra systems as Web services and local/remote calls of facilities of another system in easy and efficient way
- This may be used to combine several copies of the same system in a parallel computing context of various scales from multi-core to grids

Some existing developments for GAP

- Interfaces:
  - Alnuth (Bjoern Assmann, Andreas Distler, Bettina Eick)
  - singular (Marco Costantini, Willem de Graaf)
  - GAP – polymake interface (Marc Röder)
  - if (Marco Costantini)
  - OpenMath (Marco Costantini, Andrew Solomon)
- Web services (clients)
  - AtlasRep (John Bray, Thomas Breuer, Simon Nickerson, Richard Parker, Robert Wilson)
  - QaoS (Sebastian Freundt, Sebastian Pauli)
  - UnitLib (A.K, Elena Yakimenko)
- Parallel computing:
  - ParGAP (Gene Cooperman)
  - direct condensation (Frank Lübeck, Max Neunhöffer)

Most common restrictions:

- interfaces do not support remote communication
- transmission of large or complex objects may be difficult
- Support of new system requires new I/O convertor. It relies upon the I/O format, may be subject to parsing errors and needs update if I/O format of the linked system changes
- not enough deeply (syntax, cd) and widely (other CAS) supported data encoding format (OpenMath)
- not interactive, just database access (Web services)
- not enough robust (ParGAP)
- less efficient for irregular parallel computing (ParGAP)
- shaped to deal with the particular problem (dc)
- may not work in some operating systems
- may be not easy customisable by the end-user

Example: maximal order of the number field

Alnuth uses the following KANT (KASH 2.5) session:

```
kash> f:=x^4+6*x^3+5*x^2-12*x-11;;
kash> o := OrderMaximal( f );
   F[1]
     |
   F[2]
  /
 /
Q
F   [ 1]     Given by transformation matrix
F   [ 2]     x^4 + 6*x^3 + 5*x^2 - 12*x - 11
Discriminant: 3600

kash> b := List( Basis( o ), EltToList );
[ [ 1, 0, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 2/7, 0, 1/7, 1/7 ] ]
```

KASH3: changes of output, session format and function names

```
kash% f:=X^4+6*X^3+5*X^2-12*X-11;;
kash% o := MaximalOrder( f );
Maximal Order of _BO
Time: 0.026065 s
kash% b := BasisMatrix( o );
[   1   0   0   0]
[   0   1   0   0]
[   0   0   1   0]
[2/7   0 1/7 1/7]
Time: 0.481205 s
```

The SCIEnce project

The SCIEnce project address this difficulties by a programme of standards developments and implementations for symbolic computation software to use Web services and OpenMath technologies, allowing them to be efficiently composed to solve complex problems.

Participating systems:

GAP	**KANT/KASH**
Maple	**MuPAD**

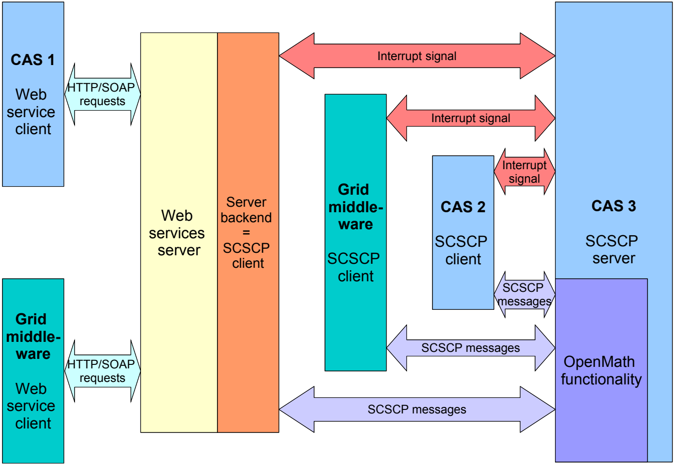
Another direction of work is development of the middleware for parallel symbolic computing on the Grid using GpH

Common protocol for communication

In the direction of the software composability, on the first step we designed the *Symbolic Computation Software Composibility Protocol* (**SCSCP**) by which a computer algebra system (CAS) may offer services for the following clients:

- A Web server which passes on the same services as Web services using SOAP/HTTP protocols to another clients
- Grid middleware
- Another instance of the same CAS (in a parallel computing context)
- Another CAS running on the same computer or remotely

Current vision of SCSCP usage



OpenMath inside

- Protocol messages represented as OpenMath objects
- Content Dictionary **cascall1** developed for this purpose
- **SCSCP** specification defines semantical and technical descriptions and allowed sequences of OpenMath-encoded messages to and from CAS:
  - remote procedure call
  - returning result of successfully completed procedure
  - returning a signal about procedure termination
- Both transmission of actual mathematical objects and references to them are supported
- Flexibility: service designer can choose the data to be `OMSTR`, `OMB`, `OMFOREIGN`, containing information in some other format, including MathML

cascall1 CD defines:

- three main kinds of messages: procedure_call, procedure_completed, procedure_terminated
- options that may be added to the procedure_call message: option_runtime, option_debuglevel, option_min_memory, option_max_memory, option_return_object, option_return_cookie
- information that may be supplied with the result: info_runtime, info_memory, cookie
- standard errors: error_runtime, error_memory, error_system_specific

Example: `procedure_call` message

Identifuing permutation group in the GAP Small Groups Library

```
<OMOBJ>
    <OMATTR>
        <OMATP>
            <OMS cd="cascall1" name="call_ID"/>
            <OMSTR>01-234567890@gap1</OMSTR>
        </OMATP>
        <OMA>
            <OMS cd="cascall1" name="procedure_call"/>
            <OMSTR>GroupIdentificationService</OMSTR>
            <OMA>
                <OMS cd="list1" name="list"/>
                <OMA>
                    <OMS cd="permut1" name="Permutation"/>
                    <OMI> 2</OMI>
                    <OMI> 3</OMI>
                    <OMI> 1</OMI>
                </OMA>
                <OMA>
                    <OMS cd="permut1" name="Permutation"/>
                    <OMI> 2</OMI>
                    <OMI> 1</OMI>
                </OMA>
            </OMA>
        </OMA>
    </OMATTR>
</OMOBJ>
```

Example: procedure_completed message

The procedure GroupIdentificationService evaluated the procedure_call message and determined that the permutation group has the number [6,1]:

```
<OMOBJ>
    <OMATP>
        <OMS cd="cascall1" name="call_ID"/>
        <OMSTR>01-234567890@gap1</OMSTR>
    </OMATP>
    <OMA>
        <OMS cd="cascall1" name="procedure_completed"/>
        <OMA>
            <OMS cd="linalg2" name="vector"/>
            <OMI> 6</OMI>
            <OMI> 1</OMI>
        </OMA>
    </OMA>
</OMOBJ>
```

GAP implementation of the SCSCP

- GAP Package SCSCP (in development)
- Allows GAP to work as an SCSCP server and client
- Uses GAP packages IO, GAPDoc and OpenMath.dev
- Uses recent developments by Steve Linton for the exception and error handling in GAP

Communication via TCP/IP protocol

- On the server side:
  - creates a socket
  - binds local address to the socket
  - switches socket to listening
  - waits for an incoming network connection
- On the client side:
  - creates a socket
  - connects to the remote socket
- This interface is wrapped into InputOutputTCPstreams:
  - compatible with other kinds of GAP streams
  - transparent access from other stream-using code
  - usage of stream-based functionality of OpenMath and GAPDoc packages

OpenMath functionality:

- Writing and reading OpenMath messages into/from InputOutputTCPStreams
- Parsing OpenMath code using GAPDoc parsers for XML encoding
- Processing the result using our extensions from the SCSCP package:
  - new symbols from the cascall1 OM CD
  - support of OM attributes (OMATTR, OMATP)
  - support of OM references (OMR)

User-level functionality:

- Installing procedures available as SCSCP services
- Running the SCSCP server
- Sending request to the server and getting result
- Store/Retrieve procedures allowing to work with remote objects

Example: identification of groups of order 512

The following function accepts an integer that is a code for pcgs of a group of order 512 and returns the number of this group in the GAP Small Groups library using the ANUPQ package:

```
IdGroup512ByCode:=function( code )
local g, f, h;
g := PcGroupCode( code, 512 );
f := PqStandardPresentation( g );
h := PcGroupFpGroup( f );
return IdStandardPresented512Group( h );
end;
```

It is installed on the SCSCP server by the command

```
InstallSCSCPprocedure( "IdGroup512ByCode", IdGroup512ByCode );
```

Example: identification of groups of order 512

The client’s counterpart takes the group in pc-presentation and sends its pcgs code to the server to identify the group.

```
IdGroup512:=function( G )
local code, result;
if Size( G ) <> 512 then
  Error( "G must be a group of order 512 \n" );
fi;
code := CodePcGroup( G );
result := EvaluateBySCSCP( "IdGroup512ByCode",
            [ code ], "localhost", 26133 );
return result.object;
end;
```

How it works:

```
gap> IdGroup512(DihedralGroup(512));
#I  Creating a socket ...
#I  Connecting to a remote socket via TCP/IP ...
#I  Got connection initiation message SCSCP_VERSION 0 CAS_PID 1
#I  Request sent, waiting for reply ...
[ 512, 2042 ]
```

Example: communicating with other software

A java program by Dan Roozemond searches in the On-Line Encyclopedia of Integer Sequences.

What is the meaning of the following sequence:

1, 1, 1, 2, 1, 2, 1, 5, 2, 2, 1, 5, 1, 2, 1, 14 ?

Example: communicating with other software

Of course, this is the number of groups of orders $1, 2, \ldots, 16$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	2	1	2	1	5	2	2	1	5	1	2	1	14

Here we obtain a record with the search result:

```
gap> EvaluateBySCSCP("OnLineEncyclopediaOfIntegerSequences",
>     [ [ 1, 1, 1, 2, 1, 2, 1, 5, 2, 2, 1, 5, 1, 2, 1, 14 ] ],
>     "localhost", 26133 );
#I  Creating a socket ...
#I  Connecting to a remote socket via TCP/IP ...
#I  Got connection initiation message SCSCP_VERSION 0 CAS_PID 1
#I  Request sent, waiting for reply ...
rec( object := [ 1, "A000001: Number of groups of order n." ],
attributes := [ [ "call_ID", "0" ] ] )
```

Example: OR-parallelism with FactInt package

Continued Fraction Algorithm vs. Multiple Polynomial Quadratic Sieve

```
gap> for i in [1..120] do r:=FactorsCFRAC( 2^i+1 ); od; time;
4789
gap> for i in [1..120] do r:=FactorsMPQS( 2^i+1 ); od; time;
3330
```

`ParEvaluateBySCSCP` applies various methods to the same argument, waiting for the first available result. This example was made with a dual core CPU:

```
gap> for i in [1..120] do
>      r:=ParEvaluateBySCSCP( [ "WS_FactorsCFRAC", "WS_FactorsMPQS" ],
>                             [ 2^i+1 ],
>                             [ "localhost", "localhost" ],
>                             [ 26133, 26134 ] );
>    od; time;
532
```

Example: remote objects

On the SCSCP server:

```
InstallSCSCPprocedure( "WS_IdGroup", IdGroup );
RunSCSCPserver( "localhost", 26133 );
```

On the SCSCP client:

```
gap> s:=StoreAsRemoteObject( SymmetricGroup(3), "localhost", 26133 );
< remote object TEMPVarSCSCP1 at localhost:26133 >

gap> EvaluateBySCSCP( "WS_IdGroup", [ s ], "localhost", 26133 );
rec( object := [ 6, 1 ], attributes := [ [ "call_ID", "" ] ] )

gap> RetrieveRemoteObject(s);
Group([ (1,2,3), (1,2) ])

gap> UnbindRemoteObject(s);
true
```

What we have in the result

- low-overhead (compensated by easy-to-use), robust, cross-platforming, light-weight and reliable protocol
- possibility of communication not only between CASs but also between CAS and other software, including Web and Grid services
- besides the four participating systems (GAP, KANT, Maple and MuPAD), we expect more systems joining SCSCP framework later

Work in progress

Next steps:

- Adding basic SCSCP implementations to all participating systems
- Identifying and developing new OpenMath content dictionaries and other standards extensions needed
- Adding support of selected OpenMath CDs in all participating systems
- Implementing higher level interfaces in all participating systems

Short-term goals:

- GAP-KANT interface in SCSCP for Alnuth
- MS Windows support in the SCSCP package (we may provide Windows binary for IO, Browse and Edim)

References

A. Konovalov, S. Linton. *Symbolic Computation Software Composability Protocol Specification.*
CIRCA preprint 2007/5, The University of St Andrews. http://www-circa.mcs.st-and.ac.uk/pre-prints.html

A. Konovalov, S. Linton. *SCSCP — Symbolic Computation Software Composability Protocol.*
GAP package, in development.

D. Roozemond. *OpenMath Content Dictionary* **cascall1**.
http://www.win.tue.nl/SCIEnce/cds/cascall1.html