

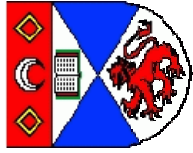
Adventures in Connecting Software



Steve Linton

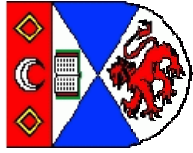
Centre for Interdisciplinary Research in
Computational Algebra

University of St Andrews



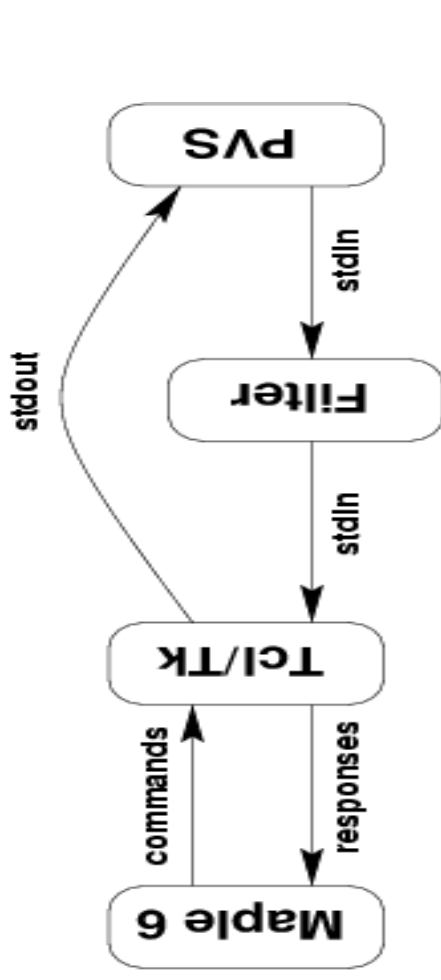
Overview

- Linking Mathematical Software
 - Maple and PVS
 - w. Adams, Dunstan, Gottliebse, Kelsey, Martin
 - GAP and X
 - work of many (several of them here)
 - GAP packages linking specialized tools
 - GAP as a specialized oracle for symmetry
 - GAP-driven Web sites
- Observations and Possible Future Directions
 - “Why aren't we using”

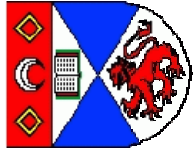


Linking Maple and PVS

- Use PVS where Maple has known deficiencies
 - Checking analytic side-conditions
- Ad Hoc interface:
 - Maple writes PVS input
 - syntactical formula translation
 - filter cleans PVS output
 - Tcl/Tk manages PVS interaction
- Custom Real analysis theory in PVS

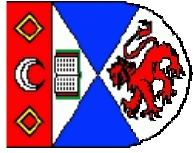


```
> f := 1/(cos(x) + 2);
      1
      cos(x) + 2
> iscont(f, x=-infinity..infinity);
      false
> PVSiscont(1/(cos(x) + 2), -infinity, infinity);
      true
```



Maple/PVS Ctd.

- Theory and tactics library can automatically prove many useful properties of real functions:
 - continuity, limits, differentiability, monotonicity, etc.
 - “High school” syntax-directed strategy
- Applications to discharging side conditions for definite integration, ODEs, etc.
- Now applying same tools to analysing control systems
 - avionics, air traffic control,...
 - Simulink/PVS, work in ClawZ, ProofPower
- NASA, Qinetiq



GAP – The Publicity Slide

Groups Algorithms Programming

Symmetry -- a universal phenomenon in science and nature

Groups -- the mathematical language for describing symmetry

GAP -- state of the art software for computing with groups

500K lines of code

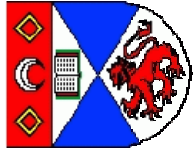
300MB of databases

20+ refereed user-contributed packages

Used by hundreds of researchers world-wide -- in mathematics, physics, chemistry and computer science



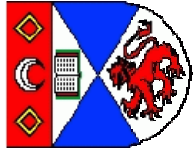
Known GAP 4 User Sites



The GAP System

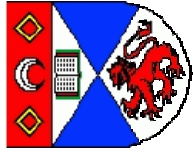
```
gap> AvgOrder :=  
> g->Sum(ConjugacyClasses  
> c-> Size(c)*Order(Representative(c)))/  
> Size(g);  
function( g ) ... end  
gap> AvgOrder(MathieuGroup(11));  
53131/7920  
gap> ForAny(AllSmallGroups([2..100]),  
> g->IsInt(AvgOrder(g)));  
false
```

- Read-eval-print interface
- Extend and interact paradigm
- Complex “abstract” objects such as groups
- Generic operations
- Object/value oriented
 - not expressions
 - not worksheets
- GPL



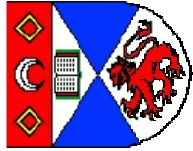
Capabilities of GAP (including Packages)

- Permutations and perm. groups
 - Finitely-presented groups
 - Polycyclic groups
 - Matrix groups
 - finite fields
 - crystallographic
 - Lie and other algebras and quantum groups
 - Semigroups and Monoids
 - Character tables
 - Graphs and Codes
- Basic functionality for:
 - Number theory and combinatorics
 - Polynomials and Groebner bases
 - System & support functions:
 - lists and records
 - files, strings, slave processes, packages
 - xgap -- basic GUI and some graphical applications



GAP Databases

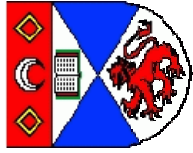
- Small Groups
 - all 423 164 062 groups of order at most 2000
 - mainly represented by polycyclic presentations
 - stored in 26MB -- an average of 2 groups/bit
 - identification info for 4 million of them
- Character Tables and Tables of Marks
 - 1323 ordinary character tables, plus Brauer tables
 - rich objects, complex links with each other and groups
- Transitive and Primitive Groups:
 - 41863 permutation groups



Interfacing to GAP

The Stone Age

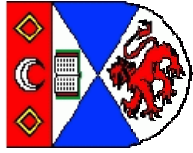
- 1990-99
- GAP 3
- Packages interfacing GAP to specialized C stand-alone software
- p-Quotient, nilpotent Quotient, Knuth-Bendix
- Vector Enumerator
- C meataxe
- nauty (twice!)
- GAP writes input files for other programme
- GAP invokes other program
- Program writes GAP input to a file, possibly aided by a translator
- GAP reads and returns result.
- Works OK.



Interfacing to GAP

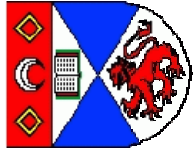
The Bronze Age

- 1999-
 - (Enhanced) packages for interacting with
 - ACE coset enumerator
 - p-Quotient, nilpotent quotient, Knuth-Bendix
 - KANT number theory package
 - Singular (finer-grained)
 - prototype OM phrasebook
- GAP 4.2 and up supports interacting with other programs via pseudo-ttys
- Steer the computation in the support program at run-time.
- String and file handling also much improved
- GAPdoc package include basic XML parser
- atlasrep package uses FTP to access database



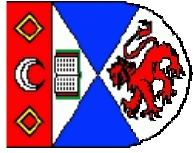
GAP as a Component

- GAP is not a good neighbour
 - intolerant memory manager
- “Have” to use GAP in a subprocess via cmd-line
- GAP UI designed for humans, not so great for automation
- Slow start-up
 - Some tips:
 - -p option adds status annotations to the output stream
 - -T disables break loop
 - saved workspaces
 - control output formats
 - Print commands
 - a GAP I/O loop
 - simplify inputs by writing GAP functions



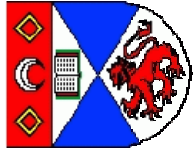
Experiences of Linking to GAP

- From Constraint Solving Systems
 - Eclipse, ILOG Solver
- GAP acts as a symmetry oracle in large search problems
- Simple direct interface via GAP command line
 - supported by a few special GAP functions
- Simple GAP on the Web
 - easy to set up a GAP program for CGI
 - slow start a problem
 - so keep a GAP process running and talk to it from CGI scripts or servlets
 - take care of GAP state!
- Algebra Interactive gapplets similar



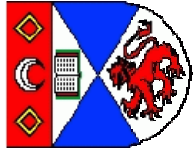
Assessment of Simple Interfaces

- Fundamentally they can work well
 - easier to build specialised interface than general ones
 - wheel gets reinvented a lot
- Software structure is totally static
 - no discovery or service broking
 - two copies of nauty
- Limitations of ASCII interfaces
 - GAP parser is very fast
 - still a problem for fine-grained cooperation



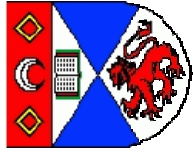
Towards an Iron Age 1

- People have experimented with more sophisticated GAP-based Web sites/services
 - OpenMath for data transport
 - Some general RPC or session protocol
 - Solomon's assign/retrieve protocol
- People here have built GAP based Web services
 - please correct me, if this is wrong!
 - mainly deal with numbers and maybe very easy calculations with permutation groups



Towards an Iron Age 2

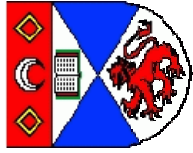
- Solomon & Struble built character table Web page
 - GAP `<--pipes--> servlet <--RMI--> applet`
 - a lot of work produced an applet that could display a character table nearly as well as GAP,
 - but do far less with it
 - GAP character table library users want to:
 - compute structure constants
 - search for conjugacy classes or representations of interest
 - exploit links between tables
 - link tables to concrete groups
 - The applet supported none of these



Handling Complex Objects

“What is a Group?”

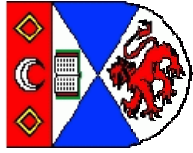
- Try to write an OM CD fragment defining a constructor for a group (or algebra or character table)
 - “simple” mathematical definition
 - group = (Set, binary $*$, unary inverse)
 - not a useful representation most of the time
 - generating elements (and maybe operations)
 - OK for permutation and matrix groups
 - Unnatural for finitely-presented or polycyclic groups
 - May lose expensive and useful information
 - Needs a serious tool to “understand” the group



“What is a Group?”

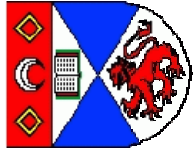
GAP's Answer

- In GAP a group is a *closed, associative* Collection of elements which support multiplication, inverse and One
 - closure and associativity are properties
 - tested or asserted
 - a Collection is required to support certain methods
 - Enumerator, Size
- So essentially to claim to GAP that an object is a group
 - the object (and related ones) must have the right methods
 - those methods must satisfy algebraic conditions
- Roughly speaking, “group” in GAP is an **Interface**



A Bit More About Groups in GAP

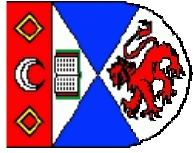
- OK, so groups in general are an interface, what about eg permutation groups
 - now we do have generators stored
 - but we may have a huge range of other things stored
 - stabilizer chain, Size, IsAbelian, derived series,....
 - can be expensive to compute, but with them, interesting questions can be answered quickly
 - the type of the object includes information on what is known about it
 - this is used by GAP in method selection



So What does this Tell Us



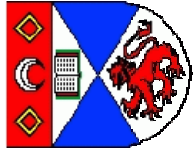
- Defining a standard serialization (OM representation) for a group is the *wrong question*
 - you can't specify a serialization for an interface
- Can specify constructors for (eg) permutation groups, but you either
 - include a huge (and evolving) collection of optional fields, or attributes; or
 - you may lose a lot of valuable and small information
- Character tables are worse
 - much information is defined by reference to other tables



OK, so it's all doom and gloom

What can we do?

- Restrict to a limited view of a specific kind of object
 - generating permutations, character values
 - useful to more programs, but only for smaller groups
- Serialize internal representation and load into an identical GAP system at the other end
 - Not silly: service based on special GAP package, special hardware or large database
 - GAP as MWS client, or local GAP accessed via gapplets, etc.
- Leave the object where it is and interrogate it
 - Needs remote object rather than RPC interface
 - assign/retrieve is a step towards this



Final Thoughts

- We see links to other computational systems as key to the future of GAP
 - keen to develop useful infrastructure for this
- Also nice to have reusable robust infrastructure for building specialized services/applications/Web sites using GAP
 - willing to try and support this with GAP system
- I feel OM ought to be the tool for these, but so far it hasn't been
- Need a stateful/object based model, not just RPC