S: past, present and future
Some thoughts

Duncan Temple Lang
Department of Statistics
UC Davis

Outline

- Good idea that was either taken up or missed. "Interfaces"
- Thoughts on how S evolved and what implications that might have for the future.
- Current work and topics in R.

Interfaces - 60s & 70s

- The word interface and some of its different meanings have been in S for a long time.
- Diagram from a meeting in the 60s between John Chambers and John Tukey showing the concept of an interface was always present.
- And originally, S was a "shell" language to interface to statistical routines.

The 80s

- S had access to the operating system shell
- C and Fortran (dynamic loading)
- In the white book, John mentions the interface concept again.
- Focus is on importing facilities to S.
90s and 2000s
- Inter-system interface model.
- Evolution from specific tools to a general approach.
- Focus on importing to R from Info. Technologies, exporting from R to non-statisticians.
- Allow each group to program in their natural environment.
- Focus on broadening the use of modern statistical techniques and software to practitioners.

Functions
- Pass function to Matlab's optimization routines.
- Allows for implementing foreign classes and methods with R functions: e.g. 
  `javaButton$AddActionListener(function(event) { ...})`
- Event handlers in R for DCOM – Excel, Word, ...  
  With the infrastructure we have (external pointers/new, S4 classes, RObjectTables, registration, ...), can make external functions 1st class objects in R.  
  `croutine(1:10, function(x) ...)`

An interface
- Computational interface
  - i.e. make functions/routines in one system appear native to the other.
  - Avoid the need to program in multiple languages.
  - Handle arbitrary data types via opaque references.
  - Bi-directional so each language can call the other makes for complete computational model.

Databases
- Using Relational databases and distributed computing within R's syntax.
- An hour of work and we have virtual DBMS tables
  `tb = RDBMSTable(dbConnect(a, "salaries"))`
  `tb[c("Salary", "Year", "playerID")]
  `tb[(30:50, 20L,302), "Salary"]`
- And use the R syntax to do SQL restrictions
  `tb[(player %in% shortlist & & Year > 1985, ]`
- Can expand to deal with multiple tables, databases, values in R.
- See S4DBMSInterface in Omegahat.
Fundamental concept is to include non-statisticians that need statistical functionality into our world and access theirs.
- The key is meta-computing programatically accessing data about the code we are interfacing to.
- Dynamically and/or statically run-time distribution time.
- Dynamic is reasonably complete.
- Static compilation of interfaces is one current area.

Benefits.
- Machine generate connections to large collection of software - existing and future.
- Can extend the generator to read different systems, generate bindings for different systems.
- Add convenience interface on top of generated accessors.
- Similar to SWIG (Simplified Wrapper and Interface Generator) but can deal with numerous input languages can process C code, not just declarations code analysis, refactoring, signatures and debugging.

Next Stage
- Ensure that we can export S functionality in the same way!
- Need support for attributes on code e.g. type specification.
- S4 is a big start.
- Perhaps need new language for software development, different from interactive use but shares the same engine.
Lessons from the early years

- Take risks.
- John’s willingness to go outside of the active scientific computing work to create a different view.
- Rather than adding more functionality to existing libraries, create a new system.
- But then again, how excited are we about new paradigms today? Skepticism, conservatism, inertia and modest goals are the norm w.r.t. computing systems.

Change the tone

- Currently, partly because of the success of S and R, we are moving to a place where we publish programming efforts rather than new concepts.
- And we are making it difficult to express ideas and computational models and frameworks because there is an expectation of software, and software that runs on all systems.
- We also need to encourage visions, big picture discussions, niche views that promote a new paradigm.

Extensibility

- And we need to make it easier for people to experiment and innovate in the system itself.
- Rather than just building new packages, we want to facilitate research in changes to the computational model of R itself.
- R has been great for enlarging the user community and involving them in adding new statistical methodology – on top of R.
- We need to also ensure that we foster involvement at the system level so we keep up with Info. Tech. advances and remain up-to-date.

Need to either make the R system easily accessible and extensible via, e.g. OOP methods.
- Or build a new infrastructure, perhaps with exactly the same model on top.
- Need funding for infrastructure projects.
Further in the future
- Extensions of package concept
- Frameworks for sharing C code between packages (on all platforms)
- Attaching DLLs, Perl modules, DBMS tables,...
- Customized and Dynamic help XML, Word, and virtual.

Near-term Ideas
- Multiple Evaluators in R (Summer 05)
- ...Threads
- Asynchronous user-level events for "reactionary programming"
- Extensibility at the system level
- Removal of switch() to allow C routines/R functions as arguments
dist(x, method = "metric_routine")
- User-defined SEPs and methods in C.

Visual Programming
- GUIs that change the paradigm of command line
- Interactive and reproducible documents.
- Authoring tools that integrate text
- Connections to Excel, Gnomic
dynamic add-ins, event handling generally for DCOM.

Education
- Need to teach statistical computing as an integral part of a program.
- Not just recipes, but faster understanding of concepts
- Language, Information Technology, algorithms, ...
- Need a few students who will be able to keep the innovation going at the system development level.
- Possibility: "workshop" for introducing the R system internals to interested, motivated researchers.
<table>
<thead>
<tr>
<th>Interactive Documents</th>
<th>Lots of good ideas about</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take text with embedded code as reproducible document, distributable source and generate views general and extensible via e.g. XML/XSL.</td>
<td>Security</td>
</tr>
<tr>
<td>Create interactive views of the text with controls a la Web browser.</td>
<td>Byte-Compilation</td>
</tr>
<tr>
<td>Robert Gentleman, Deb Nolan (StatDocs)</td>
<td>Common byte-code and interpreter with Perl, Python, etc.</td>
</tr>
</tbody>
</table>

| Events. | Streaming algorithms framework. |
| Extensible connections. | Events. |
| Mutable object class system interface (mocsi) | Extensible connections. |
| OOP, R.oo, Rproto, | Mutable object class system interface (mocsi) |

| Using statistics in the run-time of S. | OOP, R.oo, Rproto, |
| Classifying bugs and development signatures/styles via statistics. | OOP, R.oo, Rproto, |
| Caching values | OOP, R.oo, Rproto, |

...