Experiences in Integration of the 'R' System into Kepler

Dan Higgins – National Center for Ecological Analysis and Synthesis (NCEAS), UC Santa Barbara

Prepared for Sixth Biennial Ptolemy Miniconference, May 12, 2005 at UC Berkeley

http://seek.ecoinformatics.org
http://www.kepler-project.org

What is ‘R’?

“R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.”

“R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering,...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.”

From the R Project Web page - http://www.r-project.org/
Ptolemy/Kepler and R

- R language has many similarities to the PTII/Kepler expression language.

- R language emphasizes operations on vectors, matrices, and tables (in R, ‘data frames’) rather than scalars. (This eliminates many explicit looping statements).

- Many detailed statistical operations and data manipulation routines already exist in R.

- R has ability to create sophisticated graphic displays.

- Being able to call R routines from Kepler would greatly simplify many workflows.

R Example

With only 3 lines, one can read a data table, plot all combinations of column data, and summarize the data.
Interactive R in Kepler

- First Effort --- Interactive R actor
  - No real advantage over existing R console

- Use of Command Line Actor
  - Problems: R initialization
  - How to get data in/out? (files)
  - How to display graphics? (files)

- RExpression actor
  - Use concepts from Kepler/PT Expression language/actor

- Using RServer
Adding ports automatically creates R objects with the port name [e.g. `aaa <- c(1,2,3,4)`]

Graphics automatically saved as images and sent to 'graphicsFileName' output port (as file name)

R text output automatically sent to 'output' port

Adding ports creates R objects from Kepler tokens

R script is a parameter of the RExpression actor which uses port names
Array Records and Data Frames

Tables are represented as ‘Data Frame’ objects in ‘R’

A Ptolemy ‘Record of Arrays’ can also represent a table

<table>
<thead>
<tr>
<th>AAA</th>
<th>BBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>one</td>
<td>1</td>
</tr>
<tr>
<td>two</td>
<td>2</td>
</tr>
<tr>
<td>three</td>
<td>3</td>
</tr>
<tr>
<td>four</td>
<td>4</td>
</tr>
</tbody>
</table>

R Script:
```
summary(df)
```

where ‘df’ is the R dataframe created automatically when a record of arrays is passed to an input port

RExpression Output Ports

R vectors can also be assigned to output ports

R Script:
```
CCC <- df$BBB
```

where ‘ccc’ is the R name of the second column of the dataframe
EML DataSource Sequence Inputs

EML DataSource actor provides table data from SEEK Ecogrid

Column data from table can be supplied in various ways

Sequences of tokens from EML DataSource can be converted to arrays and then to a Record for input to RExpression

EML DataSource as Column Record

EML DataSource can be configured to create a “Column Based Record” directly for input to RExpression
**R Regression Analysis Example**

This is an example of how one can carry out a simple linear regression analysis using R and add the regression line to a scatter plot.

Dan Higgins - March 2005

**R Summarize Table By Species**

...
1. Input Ports

Kepler tokens are converted to R string expressions
e.g. If port AAA has token \{1,2,3\} it is converted
to the R expression `\texttt{AAA <- c(1,2,3)}`
Automatically handles strings, numbers, arrays, and records with arrays of the same length

2. R Command Line Process

R is started as a Java subprocess with text streams
attached to standard in, out, and error

3. Input Block of R Commands

A set of R commands are sent to the input stream of the R subprocess

- **Initialization**: Create graphics device (jpeg file); Create input port objects
- **User Script**: Whatever is in user’s script
  \[
  \texttt{BBB <- 2 * AAA}
  \]
- **Finalization**: R commands for output ports (e.g. BBB)
4. Execute R
   Send input block to R subprocess and get output

5. Put R results on appropriate output ports

   ```
   AAA = (1, 2, 3)
   BBB <- 2 * AAA
   ```

   ‘BBB’ R object converted to Kepler token
   e.g. {2,4,6}

RServe

“Rserve is a TCP/IP server which allows other programs to use facilities of R without the need to initialize R or link against R library.”

Client-side implementations are available for C/C++ and Java.

```java
Rconnection c = new Rconnection();
double d[] = c.eval("rnorm(10)").asDoubleArray();
```

Use of RServe would avoid each actor ‘re-starting’ R and allow remote execution of R scripts

RServe ⦿ http://stats.math.uni-augsburg.de/RServe/
Summary

An RExpression actor that operates similarly to the existing Expression actor looks like a good way of integrating R into Kepler.

Using R in Kepler provides powerful extensions to the Ptolemy expression language that allows operations on complex structures (e.g., tables).

Existing implementation is inefficient in some ways and incomplete, but is relatively easy to use and does not require detailed knowledge of R for simple operations.