Sarcomere: A System for Data Interoperability

Nassib Nassar[1], John A. Kunze[2], Gregory B. Newby[3], Kevin Gamiel[1]

1. Renaissance Computing Institute 2. California Digital Library 3. Arctic Region Supercomputing Center

Curation, sharing, and integration of data: what are the problems for scientists?

There are many challenges, but today even basic data management is still difficult and remains the province of specialists.

• Databases are hard to use. Many scientists use spreadsheets instead.
• Data sharing and integration are complicated by diversity of formats, schemas, and communication protocols.
• Developing good quality, normalized schemas is a time-consuming and exacting process.
• Integration with high performance computing resources is a cumbersome and ad hoc activity.
• Software support for schema sharing and standardization is limited.

Sarcomere and data interoperability

Sarcomere is a new system for data management. It implements a set of simple models focused on interoperability of existing database systems. Sarcomere seeks to:

• Define basic operations for data management, data exchange, and integration of data with computing resources.
• Define a data abstraction over the relational model that simplifies development of normalized schemas. (Schema-N)
• Define a protocol to make data more accessible by both people and software. (THUMP-DL)
• Provide server software and a graphical user interface that implement these proposals.
• Demonstrate integration with a variety of scientific applications and programming languages.

(Right: A Flex application that reads data from a user-specified Sarcomere URL.)

For more information

Please contact nassar@renci.org or visit the RENCI DataLab website: http://data.renci.org/

Schema-N

• Adds only semantics that are latent in the relational model (those giving rise to table decomposition), and not complex object semantics.
• Independent of both database software and programming language.
• Two special domain types: repeating groups (denoted by nested parentheses) and user defined domains (foreign keys).
• Repeating group means "0 or more tuples" and avoids problematic null values.
• Offers an alternative, hierarchical view of the database that translates to the relational schema without explicit O/R mapping.

Relational

Individuals: (name, (organism [Organisms], certainty ) )
Organisms: (genus, species , description )

The goal is to align schema creation with the logic of normalization, in order to promote development of normalized schemas.

Lightweight services

• Lightweights services: software that can be installed and run on a Sarcomere server by a user, and shared with other users.
• Executes within a per-user virtual machine.
• Envisioned for running data analysis or visualization.
• Schema sharing can be augmented with conversion software shared as lightweight services, to promote schema standardization.
• Lightweight services use Sarcomere data sources for I/O, allowing them to be relocatable. Future plans include automating deployment of lightweight services to high performance cloud computing resources.

The goal is to make sharing structured data as easy as sharing web pages, to promote broad community participation.

Integration with scientific applications

• Most scientific applications can already retrieve tab-delimited data via URL.
• An example of reading Sarcomere data in R:
  > ocean <- read.delim("http://datalab.renci.org/?in(sample:csd)")
  > plot(wind_speed ~ wind_dir)

• URL-based queries run over HTTP and return data in tab-delimited format; supports writing to databases and other Sarcomere operations.
• Queries can be submitted and results can be parsed in virtually any programming language or environment, typically without any extra drivers or software libraries.
• A THUMP-DL URL is a portable reference to a data set or subset.
• Supports schema sharing and will in the future support data exchange.

http://datalab.renci.org/?in(sample:csd)

find(record=14493)show(t|record|air_temp_avg)

2009-05-01 19:04:00 14493 23.69
2009-05-01 19:34:00 14494 23.74
2009-05-01 20:04:00 14495 23.92
2009-05-01 20:34:00 14496 24.71

Sarcomere software

• Open source Sarcomere Server written in C, layers over RDBMS via ODBC, runs as a standalone daemon or behind a web server, and implements Schema-N, THUMP-DL, etc.
• Open source Sarcomere Client written in Java, connects to the Sarcomere Server, and provides a graphical user interface for schema and data management. (Right: The Sarcomere Client interacts with a PostgreSQL database via THUMP-DL URLs and the hierarchical Schema-N abstraction.)