



“Framework Components for Distributed Computing”

Pavel Nevski, Sasha Vanyashin, Torre Wenaus

US ATLAS Collaboration Meeting

July 21-22, 1999





NOVA

Outline

- Goals
- Requirements
- Architecture
- Components
- Status
- Summary



Motivations

- Unprecedented data volume and software complexity in new large HENP experiments (RHIC, LHC...)
 - New approaches to analysis and data handling software
 - Distributed computing environment (DCE) is vital and increasingly powerful
 - Experience in developing DCE solutions for STAR
 - Build on experience to develop DCE tools for use in similarly challenging environments
- BNL is planning to have a significant role in computing and physics analysis of LHC data



- Develop software tools for
 - coordination and control of widely distributed analysis development and physics analysis activity
 - distributed management and analysis of very large datasets
 - enhanced robustness, reusability and maintainability of analysis software
- For application in many global computing environments (ATLAS, STAR, ...)
 - generic tools not tied to specific implementation choices
 - select, templatable implementations provided such that NOVA components can be used in a baseline framework

The banner features a dark blue background. On the left is a square image of a red and white astronomical object, possibly a galaxy or nebula. To its right, the word "NOVA" is written in large, bold, blue letters with a white outline. The word "Requirements" is written in a smaller, yellow, serif font, centered over the "O" in "NOVA".

NOVA Requirements

- Support wide area data intensive analysis
- Define middleware services are required to permit analysis applications to effectively run over wide area networks
- Provide a rich set of features that applications can select and use to obtain the level of service they need to operate
- Define the features and the API's necessary to allow the application and middleware to communicate
- Integrate the middleware API's with the applications



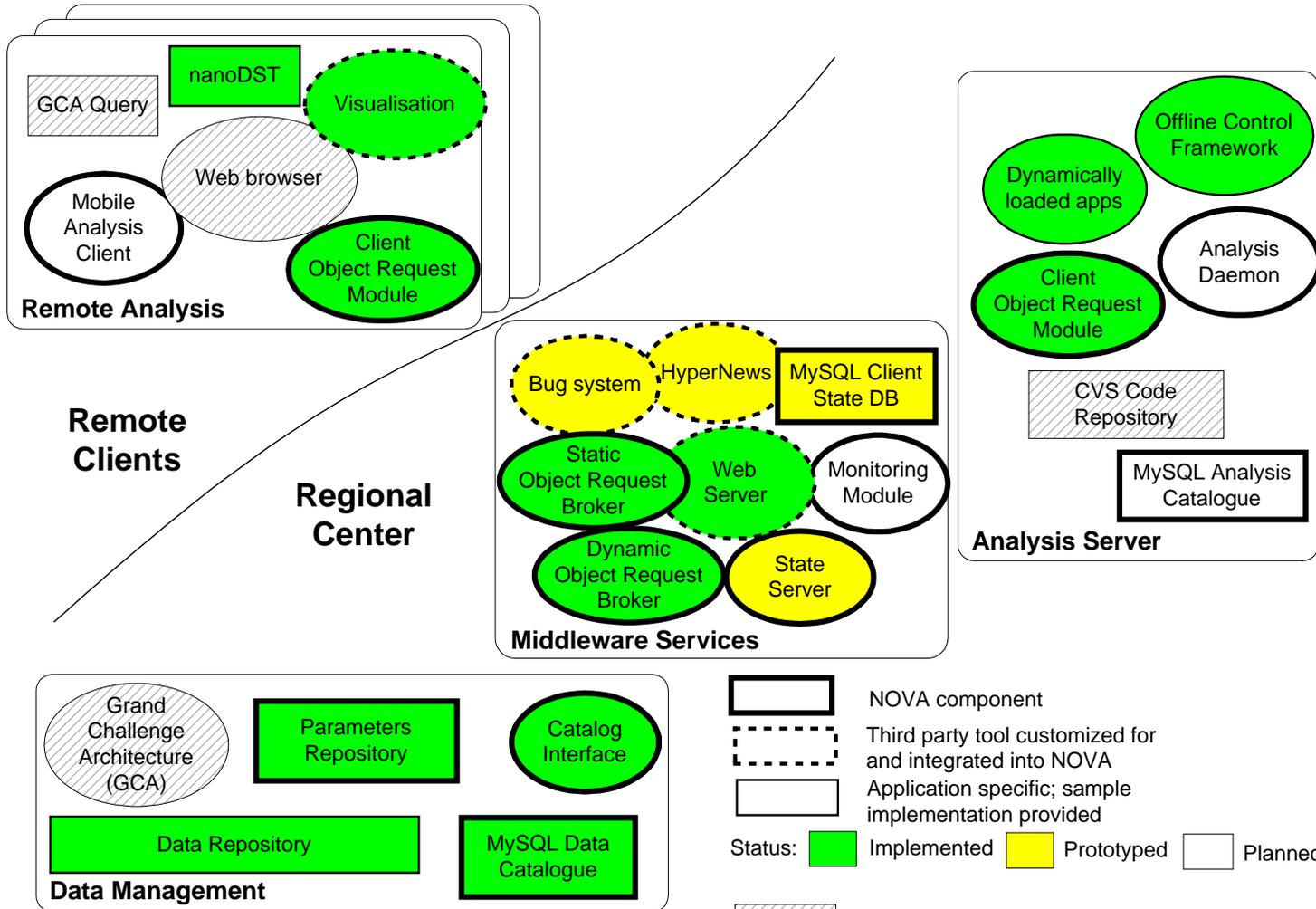
NOVA

Design Approach

- Small, modular components; application-neutral interfaces
 - Can be used as a coherent framework or in isolation to extend existing analysis systems
- Focused on support for C++ based analysis
 - Used for all RHIC, LHC, other large experiments
- Emphasis on user participation in iterative development; real-world prototyping and testing (STAR, ATLAS)
- Extensive use of existing tools and technologies
 - Must be readily available, true or de facto standards, well supported, widely used or showing good growth



NOVA Component-based Architecture





NOVA

Tools and Technologies

- Third party tools and technologies used in NOVA:
 - **MySQL**: relational database for catalogs, state information and simple objects: C-structs
 - **Apache**: customizable (**perl** & **PHP**) web server for communication and monitoring
 - **XML**: low-volume data exchange, software source distribution
 - **CORBA**: low-volume interprocess data exchange
 - **ROOT**: visualization and analysis tools



NOVA components fall into four domains

- **Regional Center**
 - Central management and execution of analysis
- **Remote Client**
 - Mobile Analysis
- **Middleware Components**
 - Data exchange and navigation tools
 - Client/Server object request brokerage
- **Data Management**
 - Data repository, catalogue, and interface
 - Data model for simple objects (C-structs)



Object Broker Use Cases

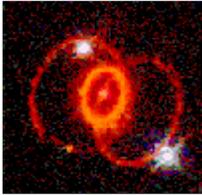
- **Problem:**

- A user has a new idea that was not foreseen at the beginning. User modifies the structure of one object in his application. Application stores new objects in the database.
- Remote applications unaware of a new functionality may request objects in old format.

- **Solution:**

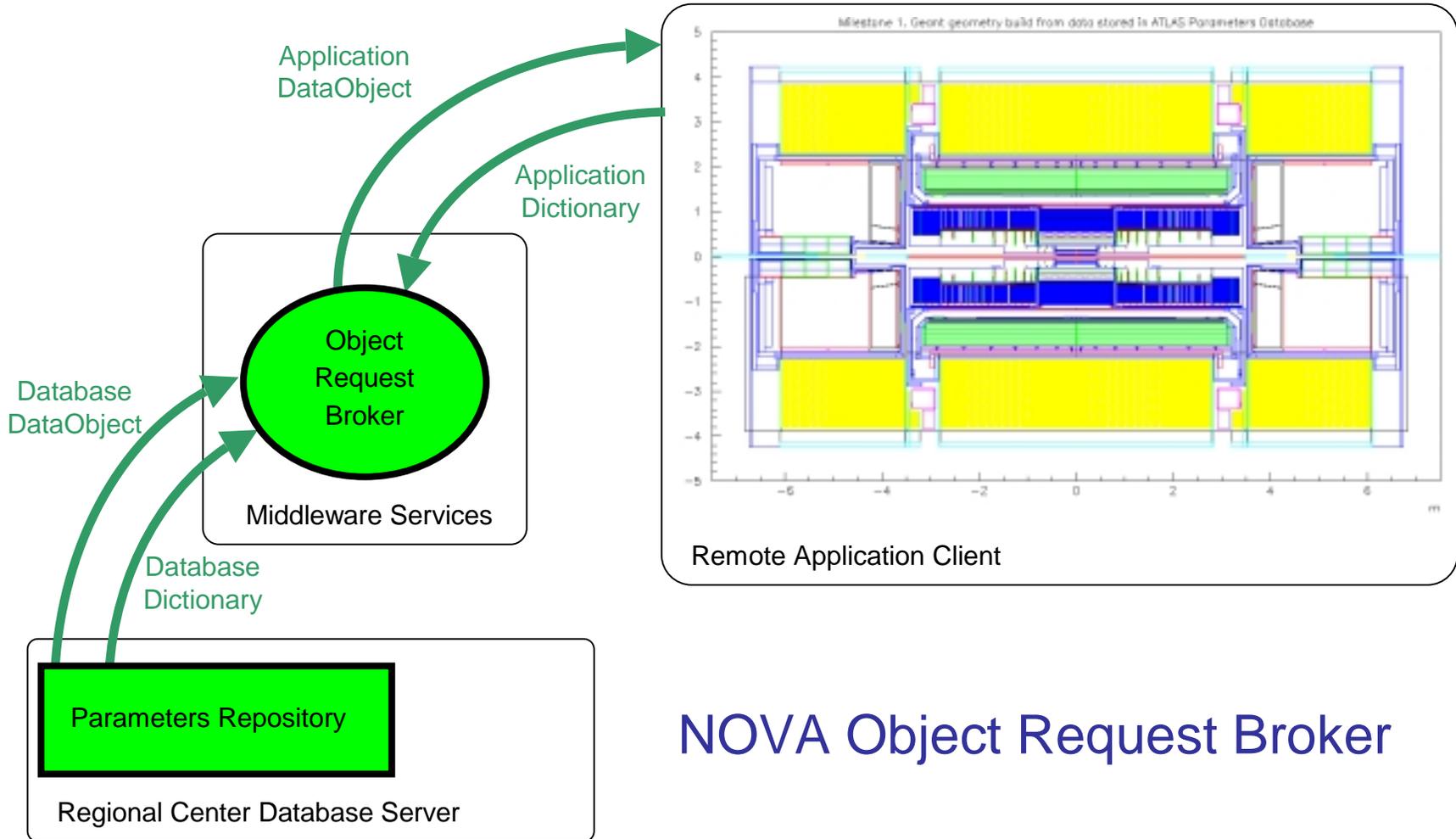
- **Application:** provides metadata request (name, time, selectors...) and the application DataObject dictionary
- **Database server:** provides DataObject and the dictionary
- **Object Request Broker module:** converts DataObject according to the application dictionary

(Dynamic Brokerage)



NOVA

Dynamic Object Broker



NOVA Object Request Broker



Dynamic Object Broker (2)

- **Benefits:**
 - Separation of database and analysis applications
 - Robust interface (via built-in type checking)
 - Dictionary built from C-header files or IDL-files
 - Database access is independent of application code version: *user can read new dataObjects with an old executable*



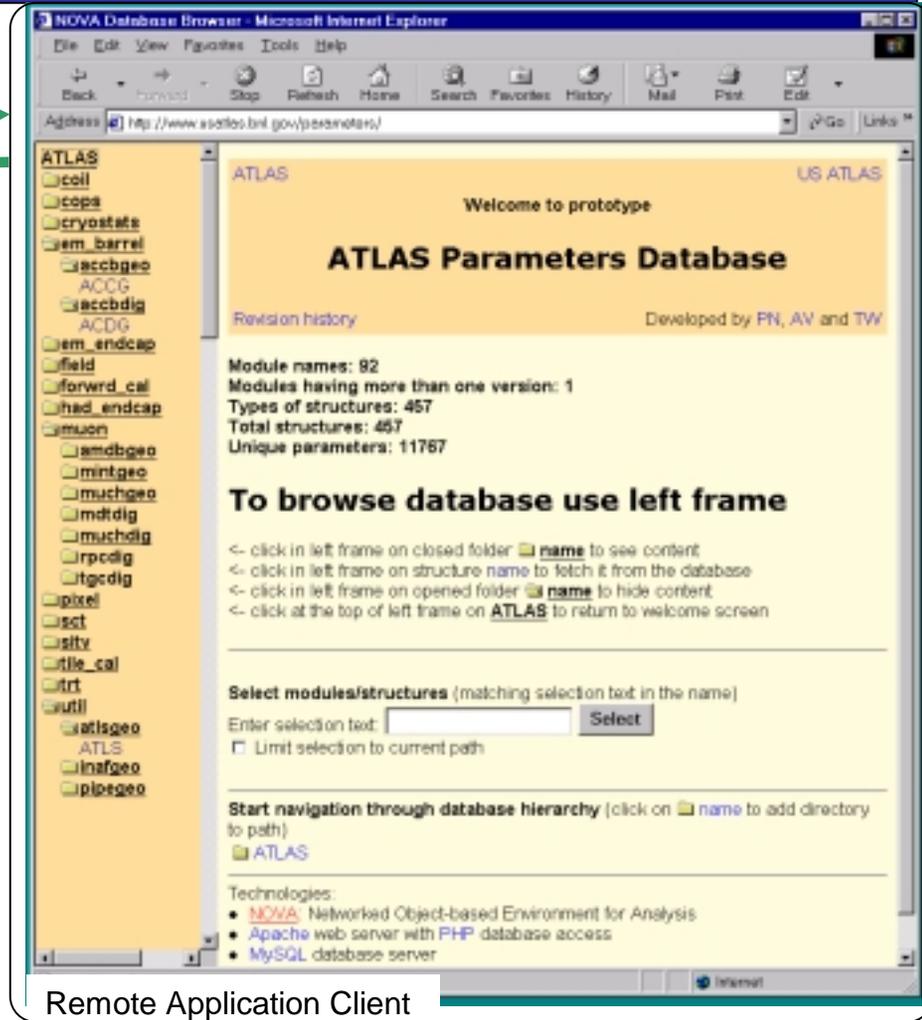
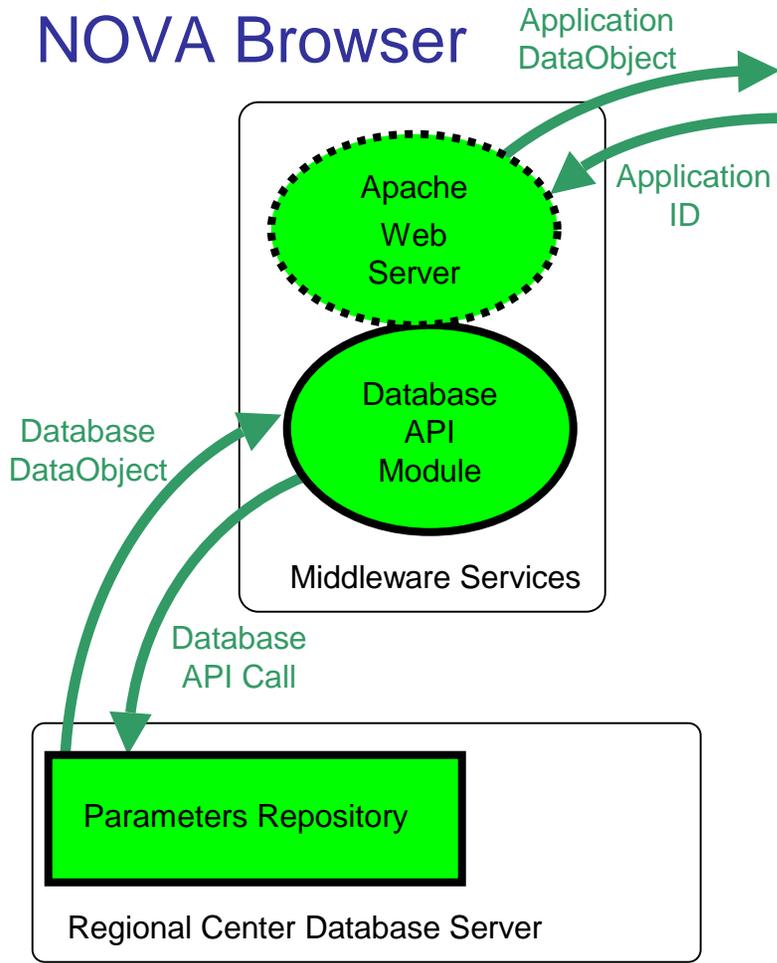
Object Broker Use Cases (2)

- **Problem:**
 - Remote application (web browser) navigates current database hierarchy.
- **Solution:**
 - Object Request Broker at the Regional Center serves dynamic HTML DataObjects in format tailored according to application ID: Netscape or MS Internet Explorer

(Static Brokerage)



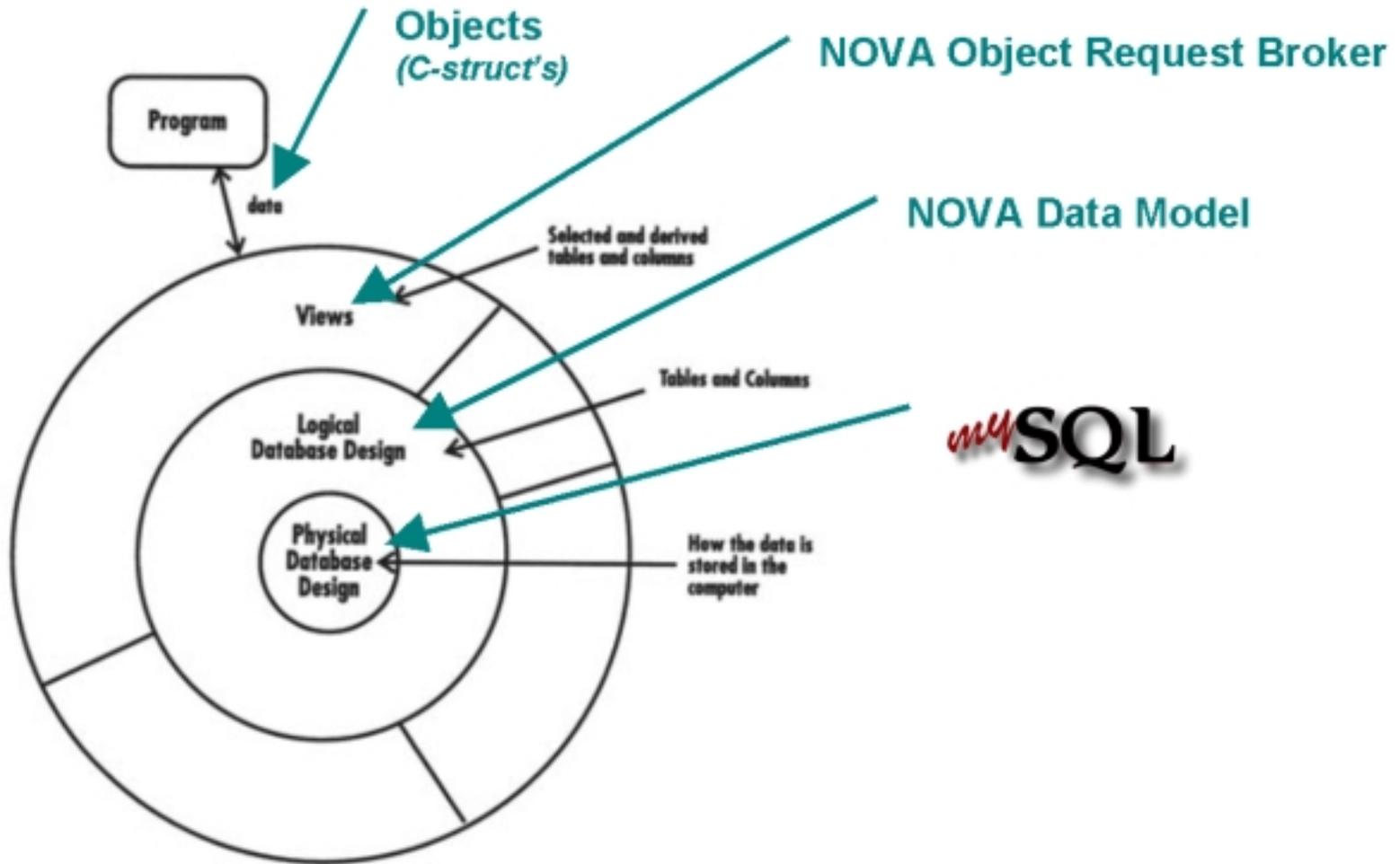
NOVA Browser





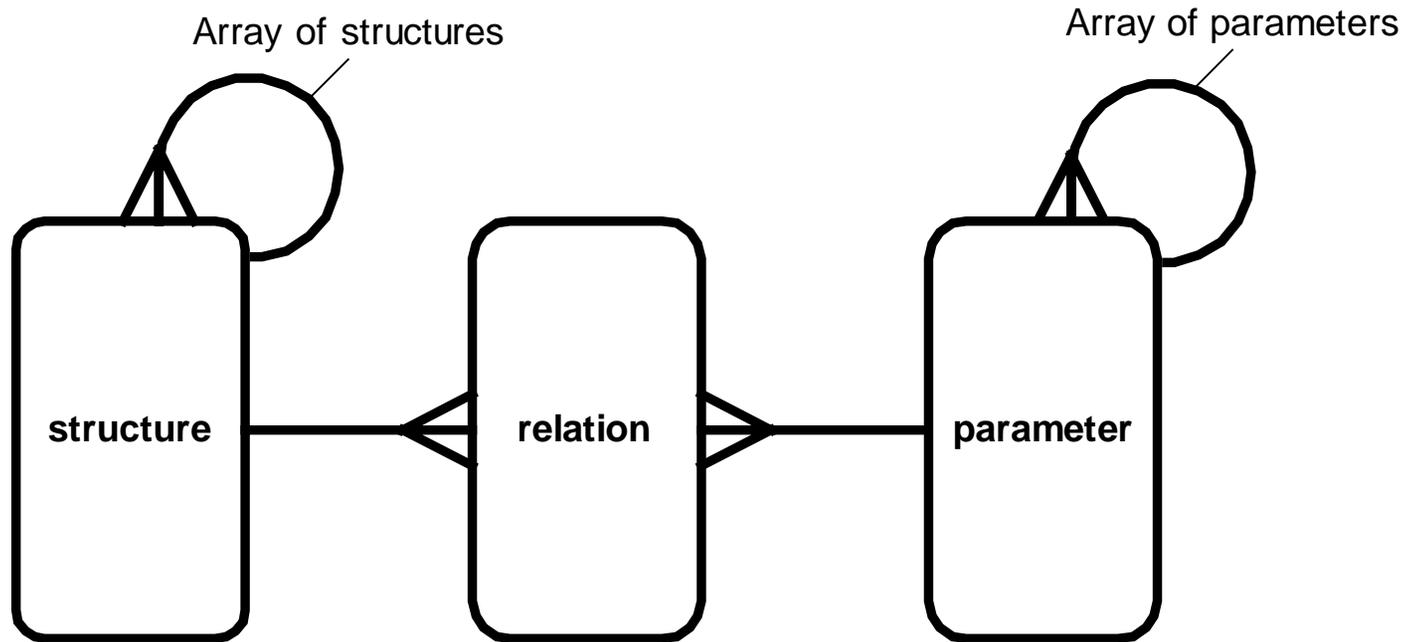
NOVA

Layered Interface





NOVA Data Model





NOVA

Summary

What is NOVA?

- framework components for distributed computing

What is NOVA status?

- several tools are tested in STAR, others to be developed
- components prototyped for ATLAS:
 - database storage of GEANT geometry/digitization parameters for simulation/reconstruction
 - web-based database navigation