A Novel Approach for Beam Commissioning Software Using Service Oriented Architecture

G. Shen*(shengb@bnl.gov), BNL, Upton, NY 11973, P. Chu, J. Wu, SLAC, Menlo Park, CA 94025

Abstract

A novel software framework is under development, which is for accelerator beam commissioning and operation. It adopts a client/server based architecture to replace the more traditional monolithic high level application approach. A minimum set of commissioning and operational services has been defined such as simulation server service, directory service, magnet service, and bpm service, etc. Most of them have been prototyped. Services can use EPICS pvData as its data container and pvAccess as communication protocol. This paper describes conceptual design and latest progress for some services.

Introduction

- Traditional application infrastructure
  - Heavy computation resulting poor performance
  - Initialization, data acquisition and many other overhead for the same application running multiple instances concurrently
  - Program flow can be quite complicated and, therefore, less reliable and hard to debug
- How to improve the situation
  - Make data available any time an application needs the data
  - Centralized computation on powerful servers
  - Easy Application Programming Interface (API) for application developers
  - SOA can provide all these benefits

Service-oriented Architecture

- Applications become "thin" clients.
- Hide all complication on the server side
- Support scripting for quick prototyping
  - For instance, an energy-changing routine may involve many steps and heavy coding efforts. This approach can greatly reduce the headache.
- No more worry for running out of memory with individual applications.
- Services should be (nearly) transparent to the client applications. This means maintenance and any computation modification is easier than the traditional approach.
- Most communication protocols can support Web easily. This means publish data on the Web becomes trivial.

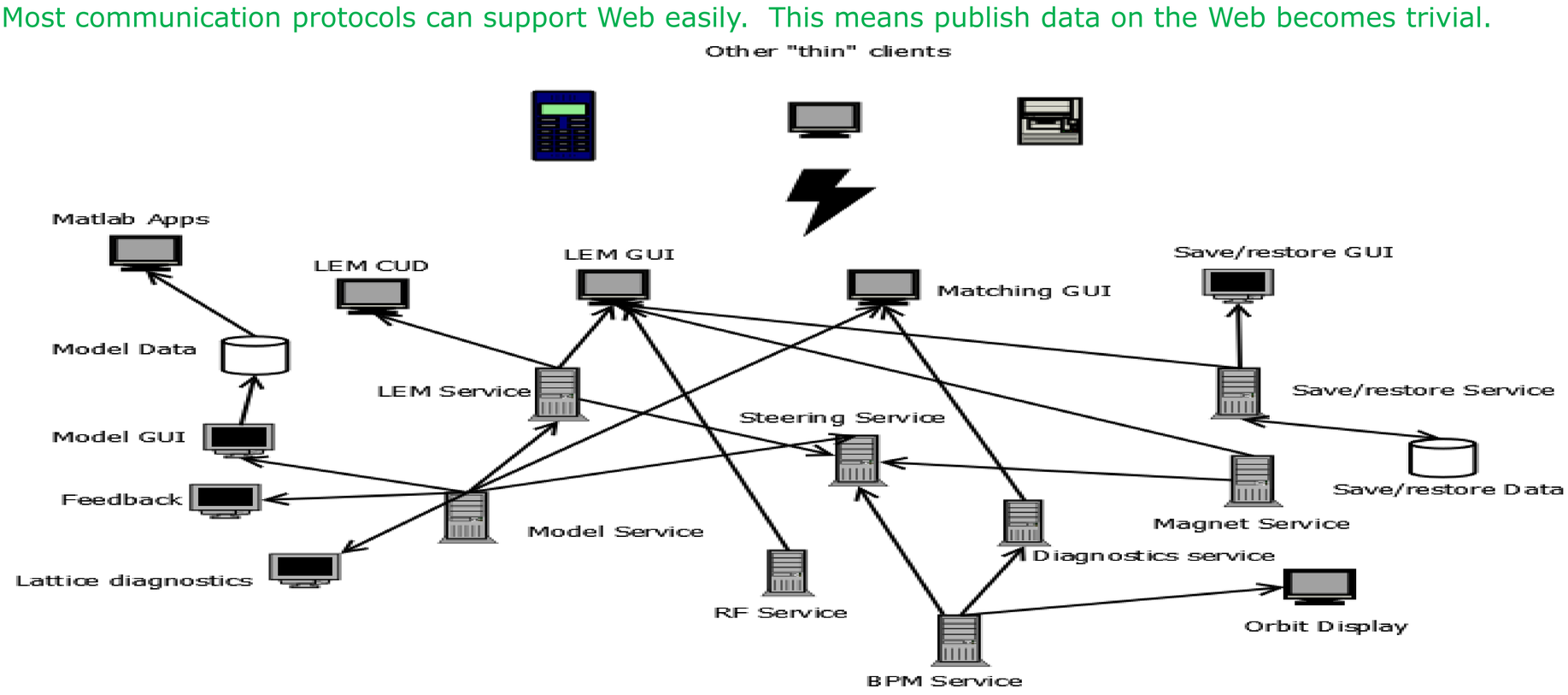

Figure 1: Top level SOA functional diagram example.

Service Provider Examples

Model Service

- Provide online model data in real time
- Can handle "what-if" model run request in addition to Design and Extant/Live models.

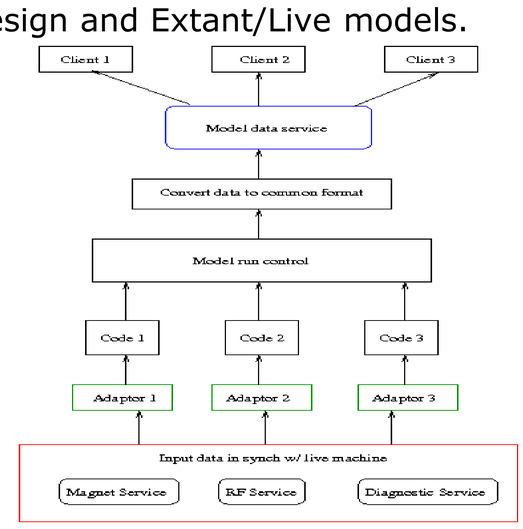

Figure 2: Data flow for model engine and service.

Linac Energy Management (LEM) Service

- Handle energy change and maintain the lattice accordingly.

Directory Service

- So-called itemFinder
- A module under epics-pvdata
- Using pvData and pvAccess
- Search from RDB according client constrain
- Return elements name and associated properties, EPICS PV names for example

Gather Service

- Accept PV list from client
- Create an array dynamically
- Organize data in service
- Monitor low level IOC EPICS

Communication protocol

An adequate communication protocol is indispensable for SOA architecture. There are many protocols available such as HTTP, XML-RPC and so on. A new generation of EPICS Channel Access protocol, pvAccess , is a better option to deliver accelerator data over the network. The main advantages are as below:

- It fully supports pvData, and depends only on project pvData. We can integrate our servers seamlessly with pvData.
- It is developed against current Channel Access, and inherits the advantages of EPICS Channel Access.

Conclusions

Some service such as Simulation Service, itemFinder, and gather service are being prototyped. They all are in the stage of choosing a good communication protocol for production and EPICS pvAccess shows a good performance as communication protocol. Some more development and benchmarking are necessary for a production server.

Acknowledgments

The authors would like to thank Matej Sekoranja at COSYLAB and Marty Kraimer for their contributions on epics-pvdata development. They also want express their thanks to Ji Qiang at LBNL for providing IMPACT-T code. They want to give their thanks to Leo Dalesio at BNL for his continuous support and encouragement.

Work supported under auspices of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC, and in part by the DOE Contract DE-AC02-76SF00515