High-level Controls, Architecture and Development

E. Hatziangeli
AB/CO/AP

Contents

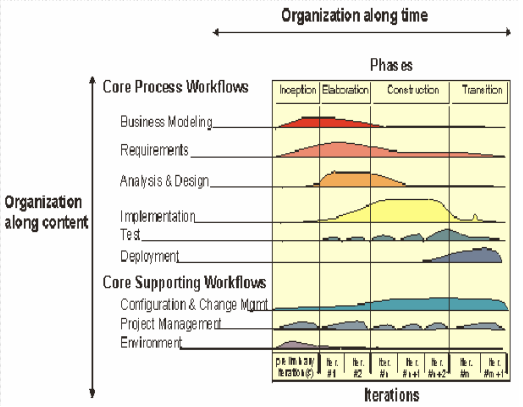
- High level development
- High level architecture
- High level controls
- LHC major milestones
- Conclusions

Contents

- High level development
  - Software Development
  - Software Implementation
  - Software Tools
- High level architecture
- High level controls
- LHC major milestones
- Conclusions

Software Development

- Our Software development process is based on Unified Software Development Process
  - Practical software process model
  - Followed by many OO industrial projects
  - Tools
    - Rational Rose
    - TogetherJ



Software Implementation

- Java
  - JDeveloper IDE (Oracle): for general development, support by IT
  - NetBeans: for GUI development, support by AB/CO/AP
- C/C++ (legacy and PVSS interfacing)
- Extensible Markup Language (XML)
  - XMLSpy, support by IT
- PVSS
  - UNICOS: PVSS development, support by AB/CO/IS

Software Tools

- Object/Relational mapping
  - TopLink (Oracle), support by IT
- Tools, support by AB/CO/AP
  - Jcover, Junit, Together Audit (testing)
  - OptimizeIt (optimisation)
  - JStyle (Quality Analysis)
- Software building, support by AB/CO/AP
  - *"common build",* AP/CO/AP made tool for Java,
    - Based on Ant
    - Covers compiling, testing, deploying, publishing, document generation
    - Will be integrated in our software Release procedure

Software Tools

- Software Configuration & Change management
  - CVS based on new IT central supported server
  - Razor © based on AB/CO/AP supported service (TBFO for Java)
- Project management
  - Goal Directed Project Management (GDPM)
    - milestone, activity and responsibility charts

Contents

High Level Architecture

Why 3 tier

- Not a new concept, already had: X-terminal consoles, Unix servers, front-ends. This is similar, but with new technology
- To avoid complete development accumulating on the top end (monolithic applications -"fat client")
- Clear separation of user-interface from domain-logic
- Domain logic can be reused, services shared by many clients
- Middle tier does common things for many clients
  - Resources sheltered from client. Reduced network load
- Middle tier is more stable
  - Deployed on a centralized, well managed server platform
  - Enhanced security
  - Scalability and availability
- Change management
  - Implementation changes are transparent from the client

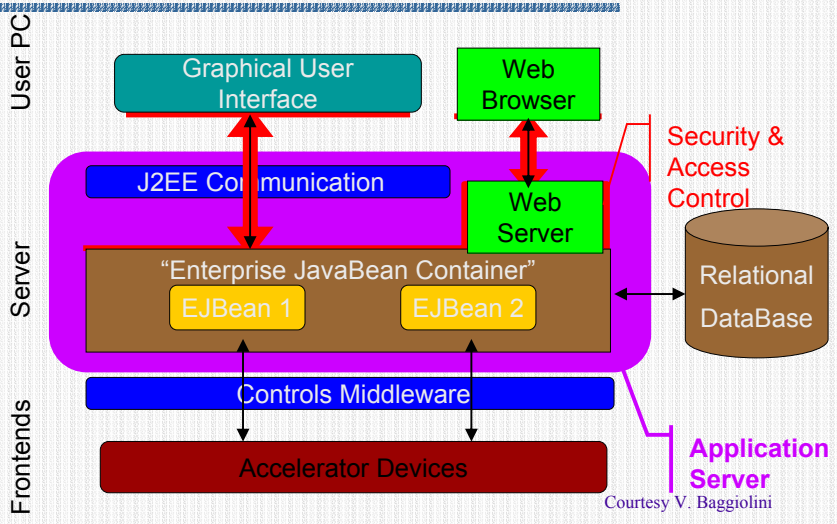
High Level Architecture

Java 2 Platform, Enterprise Edition (J2EE)

- Industry Standard to build 3-tier applications
- A recommended architecture + development guidelines
- J2EE application is component-based (Java appl, EJB,..)
- A "container" in which to run your application
- Container implements all system services, developers concentrate on domain-specific functionality
  - Integration Objects + Relational Databases
  - Transparent persistence
  - Transactions
  - Resource management (memory, threads, DB connections, …)
  - Security & Access control (authentication, authorization)
- Containers deployed on Application Servers (AS) (> 30 vendors)

High Level Architecture

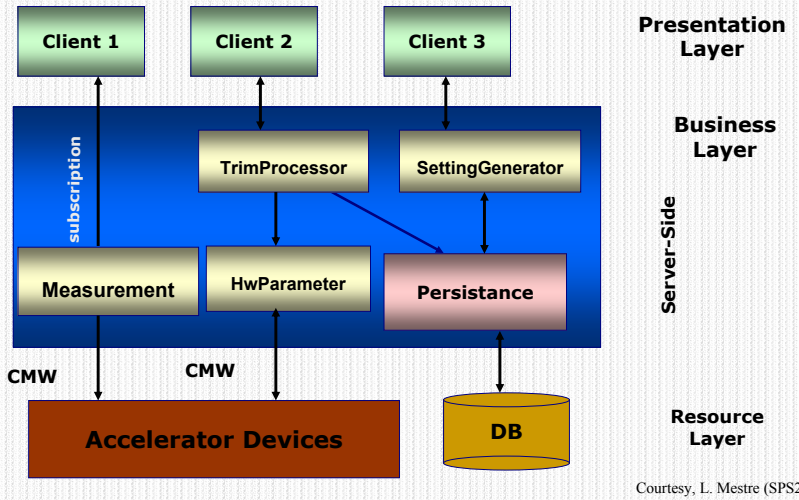
J2EE Architecture



Courtesy V. Baggiolini

High Level Architecture

SPS2001 Architecture



Courtesy, L. Mestre (SPS2001)

High Level Architecture

SPS2001 3-tier

- Presentation layer
  - Java based on the GUI platform (GP)
  - Well defined interfaces (isolated from business layer implementation)
  - Transparent communication with business layer (RMI-IIOP/JMS)
- Business Layer
  - Provides core functionality
    - Settings generation, parameter maintenance, measurements, cycle handling, interlocks
  - J2EE (Oracle AS9i) application server implementing EJB
  - Persistency via TopLink mapping to Oracle RDBMS
  - Relational database design (Oracle)
  - Device access through JAPC(CMW)
- Resource layer
  - Beam instrumentation and accelerator hardware
  - Accessed via CMW

High Level Controls, Architecture & Development

Contents

High Level Controls

Operational Environment

Operational Services offered	Operational machines offering the above services
A - Consoles to execute and display GUIs to control and operate the accelerators	A&B - Consoles LINUX dual screens and W2K triple screens supported
B - Consoles for fixed displays	C - W2K servers (PVSS & OP specific) supported by IT
C - Dedicated machines for specific services like web services, middleware brokers, digital video, system administration, etc.	C - LINUX standard PC servers for specific services
D - *Infinite* file space	D - HP-UX Fileservers
E - *Infinite* Oracle Database space	E - Oracle server supported by IT
F - *Infinite* CPU resource to run application business layer	F - Oracle 9i Application Servers (main and backup)

High Level Controls

Operational Applications

- High Level Controls
  - SPS2001 (predecessor of LHC controls applications)
  - CESAR (Cern Ea SoftwAre Renovation)
- Equipment Control
  - CMW
  - Timing
  - Device Servers (FEComSA)
- Support systems
  - LHC Logging System (+ Java, C/C++ APIs)
  - Laser (LHC Alarm system, APIs)
  - Shot by Shot Logging (SbS)
  - Post Mortem
  - OASIS (nAos replacement)
- Interlocks
  - BIC, PIC

Applications Standard Components 1/3

High Level Controls

- GUI Platform (GP)
  - Framework for building Operational Java GUIs in the LHC era
    - Goal: tools for unified and multi-team GUI development
    - Shared general purpose components for building Java GUIs
    - Mechanisms for building “pluggable” Controls applications
  - Current Users:
    - SPS-2001, Laser, CESAR, Cyan/Jaguar, OASIS
  - Future
    - One GUI platform for building new GUIs for PS (PS frame), SPS & LHC
    - Make the PS accelerator components (ASC beans) available

Applications Standard Components 2/3

High Level Controls

- Data visualization tools (DVT) to display, zoom, edit, save and print data
  - JDataviewer widely used in LEP and SPS
  - EdPlot widely used in PS
- Future
  - Aim to provide a reduced set of DVT to be used in all accelerators

Applications Standard Components 3/3

High Level Controls

- Java API for Parameter Control (JAPC)
  - Java API for parameter control that support get/set/monitor paradigm
  - Used for accessing parameters from SPS2001 business logic
  - Used for accessing properties of physical devices
  - One API, independent from underlying implementation (JMS, CMW-RDA, EJB)
  - Applications are sheltered from implementation
- Future
  - Aim to provide an implementation of the ASC Layer (PS)
  - An implementation of the SPS2001 contracts is needed for accessing BT equipments

Application Deployment

High Level Controls

- Applications Deployment
  - Java Web Start (JaWS)
    - It is provided by SUN, as part of the JDK
    - Launches Java applications, as a set of jar files, directly from the Web (slwww)
- Applications Management
  - Console Manager (YACoMa)
  - Future
    - Common Console Manager for CCR

Applications Servers (AS)

High Level Controls

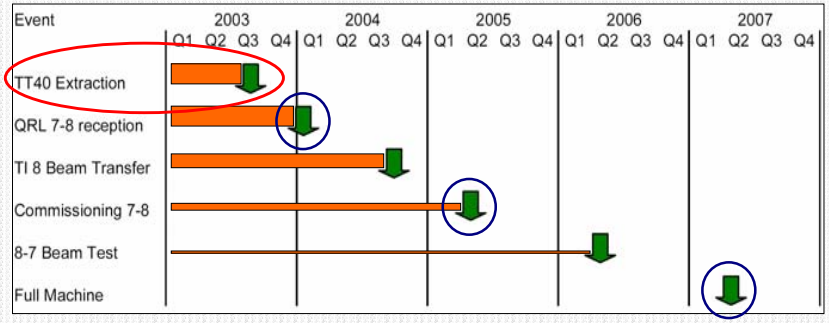
- Collaboration effort (J2EE WG) between AP, IN, DM, IT/DB, ST/MA to provide:
  - An Operational Java 2 Enterprise Edition (J2EE) Application Server
  - Support to official Controls projects (Laser, CESAR, SPS-2001)
- Deliverables:
  - Platform for Development (May 2003)
  - Platform for Operational deployment (upcoming)
- Selected product: Oracle 9i Application Server
- Challenge
  - Setup/tune the AS to ensure performance, reliability & availability needed for critical Operational applications

Contents

High Level Controls, Architecture & Development

Planning

LHC Major Milestones



TT40 2 days MDs: 8th Sept. and 1st Oct.

Height represents Robin's perception of Current Controls Activities

Courtesy R. Lauckner (LHC-Days 2003)

SPS2001 - Status Report

LHC Major Milestones

- First version of business layer Sept. 2003
  - Based on J2EE version of business layer
  - Logging (SbS)
  - Device access via JAPC
    - Subscription
- Applications Sept. 2003
  - Line Steering
    - measure trajectories & correct (magnetic elements)
    - Minimize excursions
    - Track drifts due to temperature, power converters
  - SPS selection
  - Cycle (Settings) generation
  - Measurements (+ dedicated applications)
    - Kicker & Septa settings integrated into a coherent setting measurement
  - Drive hardware
  - Trim
  - Fixed displays (BCT, Beam Loss, Beam Profiles, Trajectory)

LHC Major Milestones

SbS logging - Status Report

- Objective: record measurements on a shot by shot basis, needed for analysis of machine performance
- Data mainly from
  - beam intensities, positions, losses and profiles, power converter currents, kicker waveforms, vacuum pressures.
- Status
  - The LHC Logging System would satisfy this request
  - SPS2001 will acquire the data
  - Clarify details about the vacuum data and its time stamping
  - Decision pending about tagging data with a unique shot id

LHC Major Milestones

BIC - Status Report

- Development of the Supervision layer including operator interfaces and specialist commands
  - GUI based on GP framework
  - Using SPS2001 Business logic
  - Device access JAPC(CMW)
- First version available for hardware commissioning (1 BIC) 25th Jun. 2003
- Final version (GUI, logging) available Sep. 2003 for TT40 MDs
- Decision on final implementation (PVSS or Java) end 2003

LHC Major Milestones

QPS - Status Report

- Proposals for implementation (Java & PVSS) by end June 2003 to AB/CO-TC. If Java accepted:
  - QPS Supervision ready for surface test in Jan. 2004. This includes:
    - Basic supervision
    - Operator interface to validate the hardware tests
    - Expert commands
  - QPS Supervision ready for pre-commissioning in May 2004. This includes:
    - Basic supervision
    - Operator interface for pre-commissioning
    - Expert commands

High Level Controls, Architecture & Development

Contents

Conclusions - Development

- Software Development Process is well established and used successfully already in several projects
- Set of recommended tools is available (IDE?)
- In the process to seek central support for our recommended Java tools (CB SLAs, JavaWG)

High Level Controls, Architecture & Development

Conclusions - Architecture

- 3-tiers is the right choice of technology
- J2EE/EJB is the best way to implement it
  - Several projects (CESAR, LASER, SPS2001) are using 3-tier (J2EE/EJB) architecture => built up confidence
  - New projects (BIC, OASIS) are adopting this architecture
- Challenge: setup/tune the Application Servers to achieve the performance, reliability & availability needed for critical applications (J2EE WG)
- Cesar and TT40 tests will be a validation

High Level Controls, Architecture & Development

Conclusions - Controls

- High level controls components are prepared, based on experience with operating large machines: PS, SPS, Transfer Lines, LEP and the commonalities between them:
  - LHC controls applications software will be based on
    - Software technology choices and standard components made for SPS2001, CESAR, LASER
    - The infrastructure deployed for those projects
    - Aim for common solutions
- A clear AB/CO objective is to reduce diversity in the available CO solutions and deploy common services and components across all accelerators

High Level Controls, Architecture & Development

Conclusions - TT40 tests

- Projects progressing well, driven by realistic objectives based on the requirements for TT40 tests and other LHC major milestones
- SPS and TT40 will be used as test beds for our new controls infrastructure and software technology choices
- Need to clarify controls requirements for the next LHC major milestones for 2004 and beyond and start preparing for them

Acknowledgements

Many thanks to all people that contribute with slides, verbal input, or indirectly with their work (knowingly or not :-)

In particular:

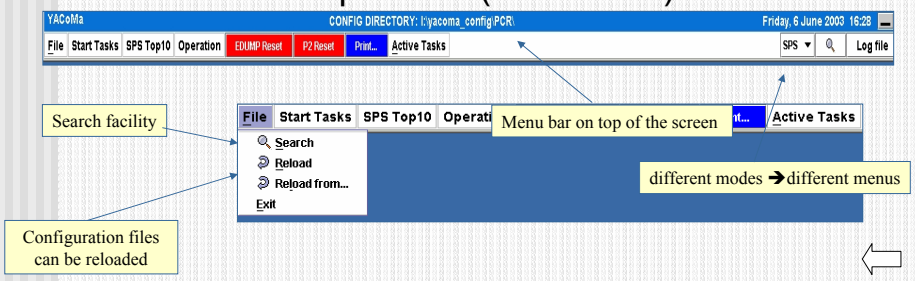
- Vito Baggiolini, Francesco Calderini, Pierre Charrue, Francois Chevrier, Greg Kruk, Mike Lamont, Robin Lauckner, Lionel Mestre, Vero Paris, Bruno Puccio, Eric Roux, Katarina Sigerud
- The pj. teams SPS2001, J2EE, GP, CESAR, JaWG, JAPC, YACoMa, SbS, …..

High Level Controls, Architecture & Development

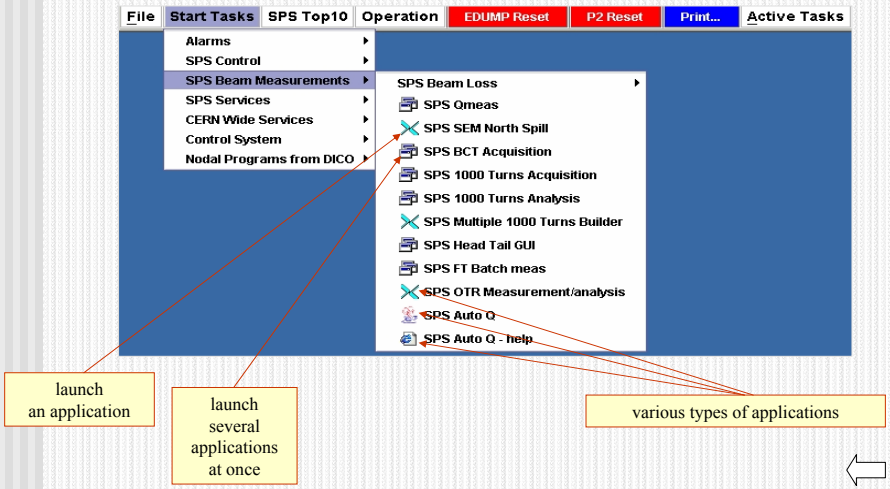
Q & A

YACoMa

- Launch and manage all types of applications:
  - X Motif, Java, Windows, PVSS
- Possible to define position and size of application windows
- Platform independent (Java + JNI)

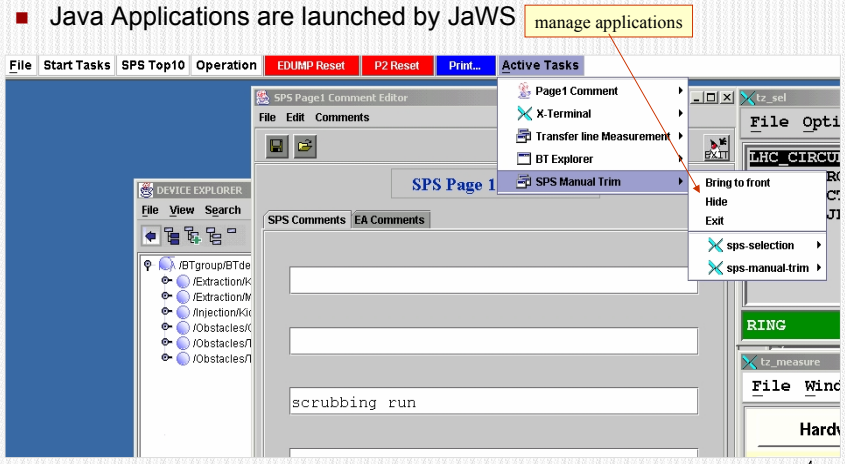


Application configurations in XML files



Manages applications

- Java Applications are launched by JaWS

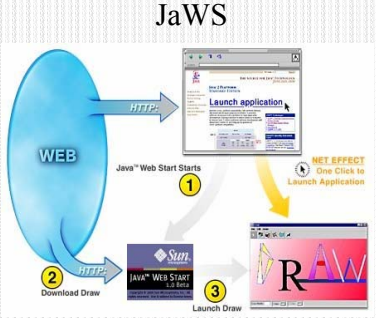


JaWS

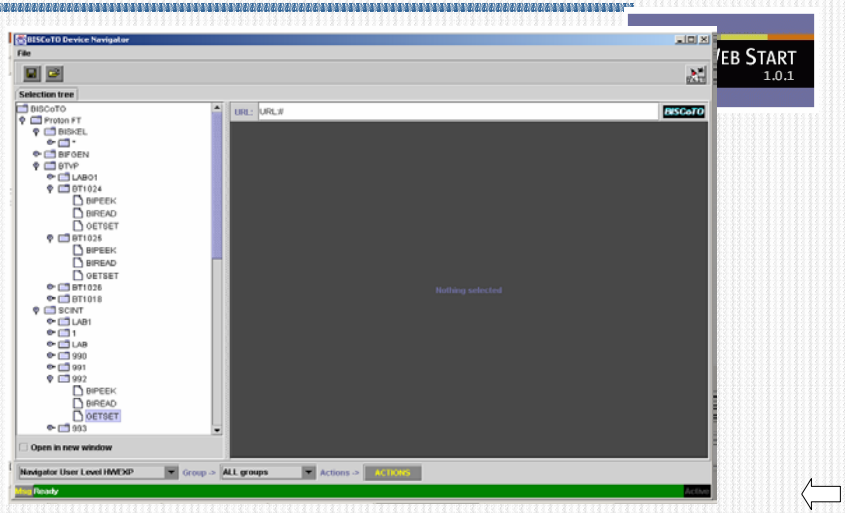
- It is provided by SUN free of charge
- Launches Java-technology-based applications directly from the Web
  - An operational web server is set up (slwww)
- Java application is distributed as a set of JAR files
- Special Java Network Launch Protocol (JNLP) file is created for each application

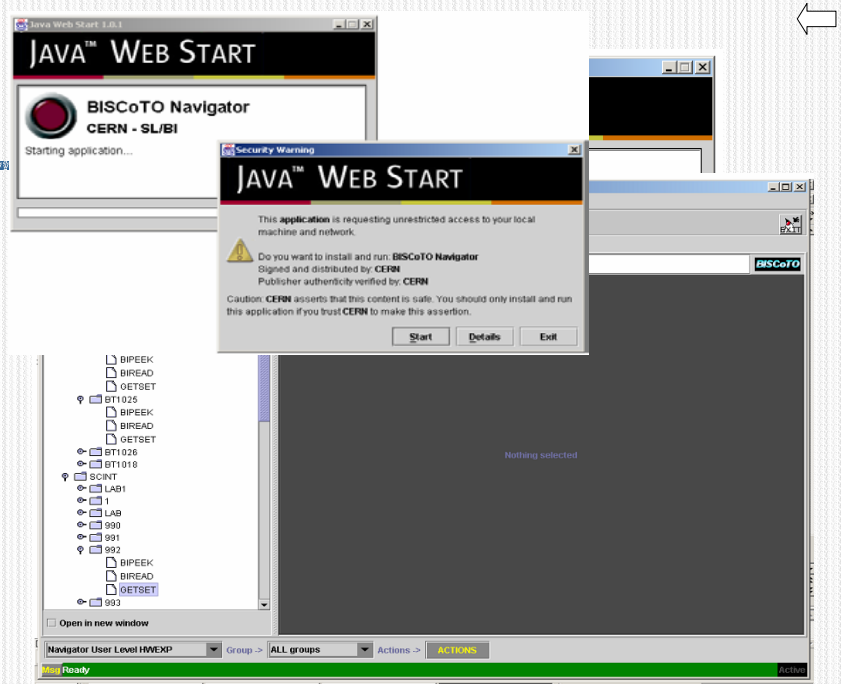
JaWS- How it works

- User clicks on a download link, the link instructs the browser to invoke JaWS technology.
- JaWS technology queries the Web to determine if all the resources needed for the application are already downloaded. If they are, and the most recent version of the application is present, the application will be launched immediately (step 3).
- If the resources are not present or an update is available, JaWS downloads the needed resources. Thus, the initial download and subsequent updates of an application happens transparently.



Application Launching & Authentication





High Level Development

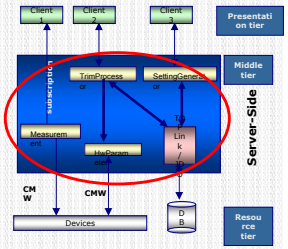
Software Production Process

- Based on Rational Unified Process
  - Analysis & Design phase
    - OOA/D process (Unified Software Development Process (USDP)
    - Rational Rose Tool & TogetherJ (Windows)
  - Software Implementation
    - Java language
    - IDEs
      - JDeveloper (Oracle) : Most recommended
      - NetBeans & Eclipse : Used & Tolerated
    - JDK
    - Development build process (more here..)

High Level Architecture - Examples

SPS2001 Business logic

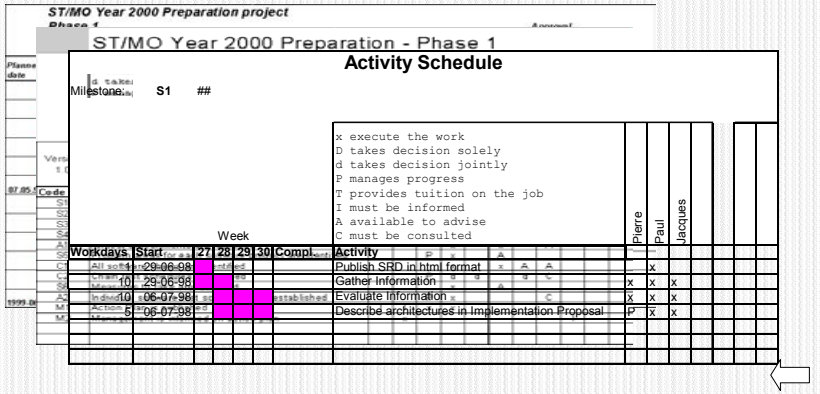
- SPS
  - Machine and cycle configuration
- Parameter maintenance
  - Settings generation, Parameter definition, Settings management, trim facilities
- Exploitation
  - Sequence change, virtual devices
- Measurements
- Settings Generation
- Interlock Handler



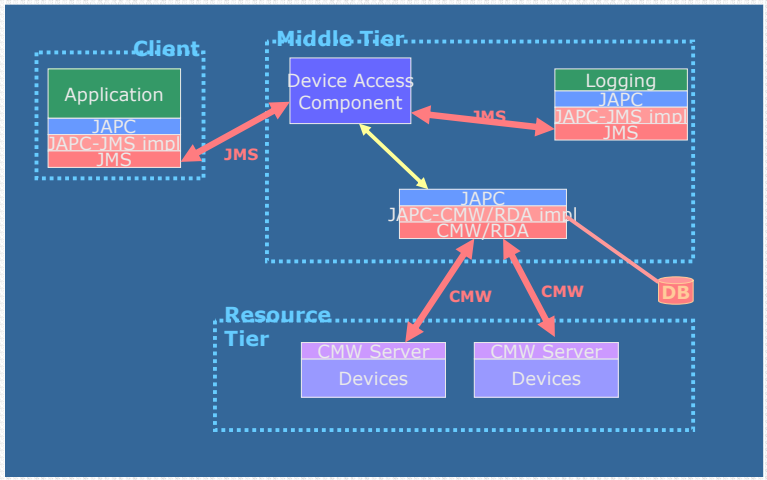
High Level Development

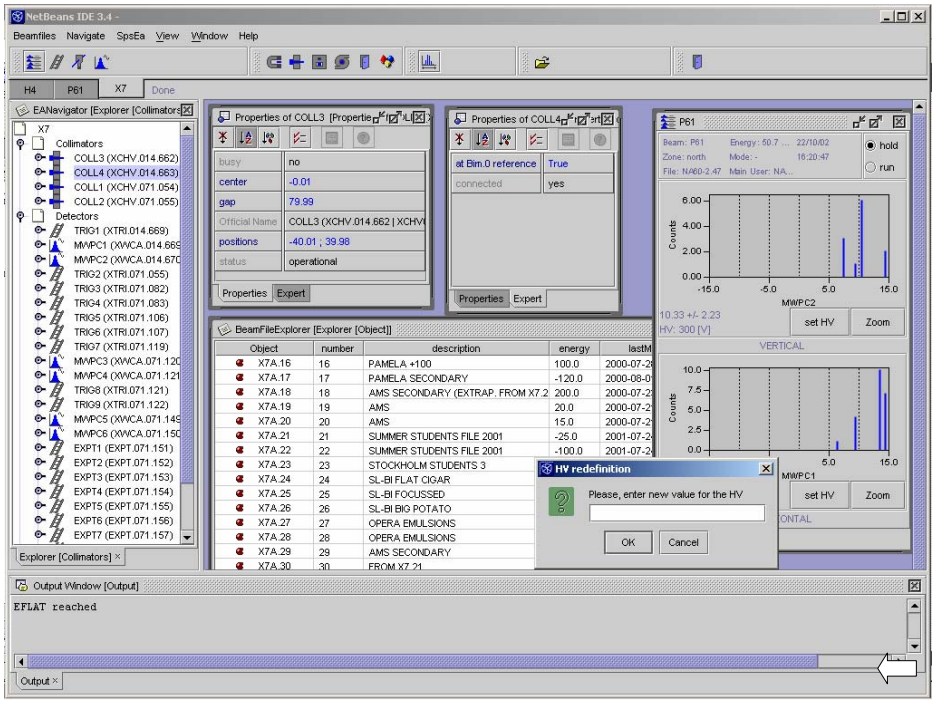
Software Tools

- GDPM

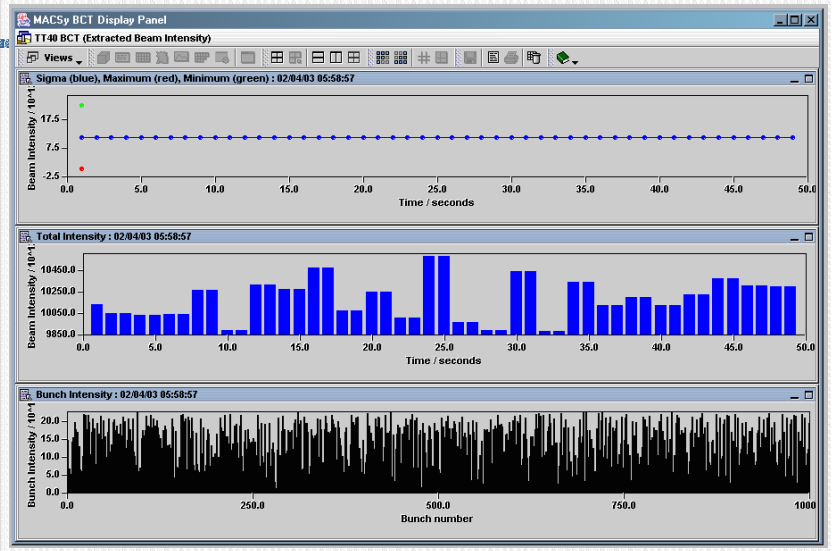


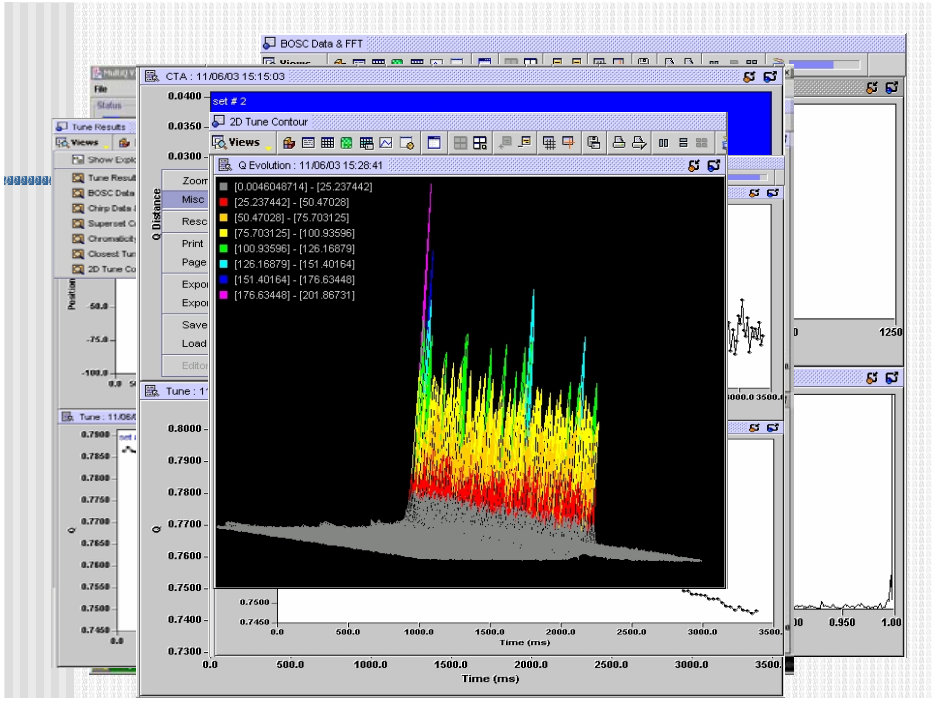
JAPC 3-tiers Architecture

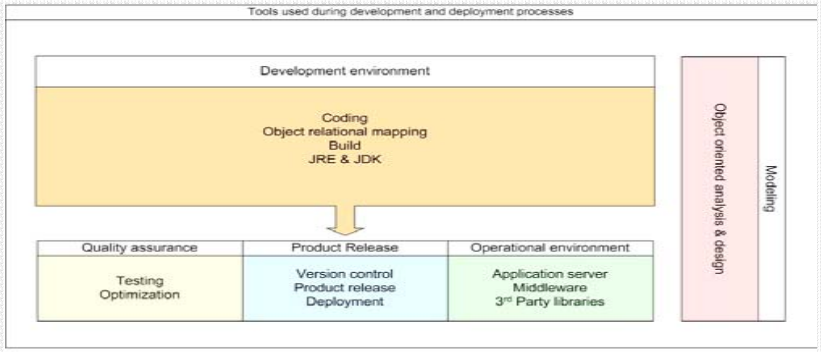




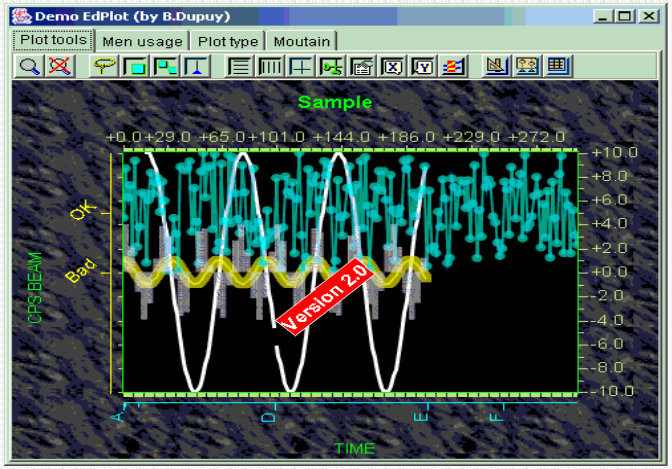
JDVE





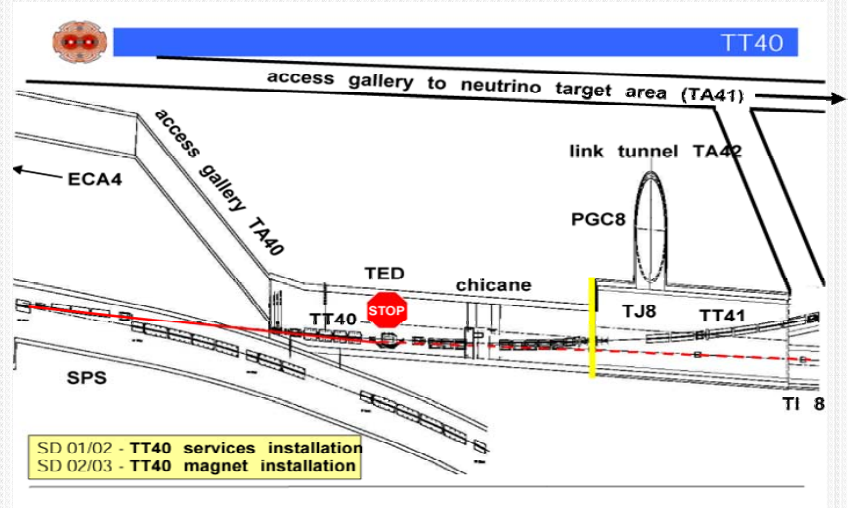


EdPlot



TT40 line

LHC Major Milestones



TT40 Tests

LHC Major Milestones

Two days MDs: 8th Sept. and 1st Oct. Objectives:

- Verify correct functioning of all equipment with beam
  - Bumpers, MKE, MSE, TT40 magnets, BI, Interlock system
- Extract beam from SPS into first part of TT40 onto TED
  - pilot beam (5×10^9 p)
- Verify and optimize trajectory and settings
- Measure acceptance of extraction channel
- Check trajectory correction
- Test of extraction interlock system
- Measure reproducibility of trajectory
- Double batch extraction (?)
- Effect of MKE kicker ripple (?)

Other Milestones

LHC Major Milestones

- QRL reception tests – Nov 2003
  - Logging & Alarms for Cryo and Vacuum systems
- TI8 commissioning with beam - Sep/Oct 2004 (4x24h)
  - 2.7 Km of beamline
  - Requirement are not clear - Use SPS2001 software?
- Sector 7-8 inj. test with beam - Apr. 2006 (> 7 days)
  - Requirement are not clear - Use SPS2001 software?
- Full LHC Machine
  - LHC controls applications software will be based on:
    - Software technology choices and standard components made for SPS2001, CESAR, LASER
    - The infrastructure deployed for those projects