The Knowledge Projection System

Ann Christine Catlin

The Purdue Knowledge Projection Group for NSWC Crane
September 13, 2004
The Knowledge Projection System

Shore-side KPS

- Ship-shore synchronization
- Session transmit & viewing
- SME recommendation processing
- Data mining
- Knowledge discovery
- Knowledge query
- Maintenance streamlining, maintenance training, equipment engineering, management reporting, ...

XML Knowledge Base

Shipboard KPS

- Online Troubleshooting
- Session capture
- Session transmit & viewing
- Ship-shore synchronization
- Session analysis feedback
- Knowledge query
KPS Supporting Technologies

Oracle 9i
J2EE
oc4j
JSP
Applet
Java v1.4.1
JDBC
Oracle 9i
XML/XSL
Relational Tables & Java v1.4.1
PL/SQL

- Online Troubleshooting
- Fault Session capture
- Fault Session viewing
- Fault Session analysis feedback
  - Session transmit
  - SME recommendation processing
  - Knowledge query

Shipboard KPS

Shipboard XML Knowledge Base

Ship-shore Synchronization

Troubleshooting Session Transmit

Shore-side KPS

Shore-side XML Knowledge Base

- Fault Session viewing
- Fault Session analysis & mining
- Session transmit
- Knowledge query
- Maintenance streamlining, maintenance training, equipment engineering, management reporting…
KPS Infrastructure: Data & Code

KPS Application Server Oracle 9i AS J2EE oc4j for web-based client interface

JSP Applet Java

• JDBC connection
• XML message communication
• SQL data query from client to Oracle 9i database

Oracle 9i XML Knowledge Base

Java Stored Procedures
PL/SQL Functions and Scripts
Supporting Relational Tables
XSD, XML, XSL

• XML/XSL or HTML for client display
• XML message communication
• SQL data retrieval from Oracle 9i database to client

Code Layers
Data Layers
THE FOCUS IS ON THE FAULT SESSION!!
How does Purdue’s KPS FY04 support Online Troubleshooting?

- XML representations for troubleshooting procedures (Scenarios)
- XML representations for troubleshooting data (smartTables)
- Java code for procedure processing (ScenarioProcessor)
- Relational tables to support troubleshooting processing
- Web-based troubleshooting interface (web client)
- Client-database XML-based communication and routing
How does Purdue’s KPS FY04 support Online Troubleshooting?

- Action-based scenario processing for step-by-step guidance
- XSL transformations of XML for web-based display
- Action-based resource access (XML Document, smartImage, Link)
- Automatic session capture
- XML representations for captured sessions
- Action-based resource capture
Troubleshooting Infrastructure - 1

12 SUPPORT RELATIONAL TABLES
- SystemStart Scenario
- SystemNext Scenario
- NextEvent
- Load_Source SRA
- Eventlinks
- TableNotes
- EventAfter Notes
- Warnings
- GeneralNotes
- Special Procedures
- DatabaseCalls
- DatabaseCall Parameters

12 SUPPORT RELATIONAL TABLES
- SystemStart Scenario
- SystemNext Scenario
- NextEvent
- Load_Source SRA
- Eventlinks
- TableNotes
- EventAfter Notes
- Warnings
- GeneralNotes
- Special Procedures
- DatabaseCalls
- DatabaseCall Parameters

12 SUPPORT RELATIONAL TABLES
- SystemStart Scenario
- SystemNext Scenario
- NextEvent
- Load_Source SRA
- Eventlinks
- TableNotes
- EventAfter Notes
- Warnings
- GeneralNotes
- Special Procedures
- DatabaseCalls
- DatabaseCall Parameters

Linkage Infrastructure to External Content
- Document, Figure, Link, smartImage, smartTable XSDs for Troubleshooting Data Representation:
  - SLQ-32 HVS Document & Figure
  - SLQ-32 HVS Link
  - SLQ-32 HVS TDD smartImage
  - SLQ-32 HVS Parts
  - SLQ-32 HVS Swaps
  - SLQ-32 HVS Signals

7 SCHEMAS
- More Than 200 XML Instances
- EventEventFeedback XSL
- SLQ-32 HVS Subtest 1 Specific/General Notes
- SLQ-32 HVS Subtest 1 Special Procedures
- SLQ-32 HVS Subtest 1
- SLQ-32 HVS Subtest 0 RunSDT

Document, Figure, Link, smartImage, smartTable XSDs for Troubleshooting Data Representation:
- SLQ-32 HVS Document & Figure
- SLQ-32 HVS Link
- SLQ-32 HVS TDD smartImage
- SLQ-32 HVS Parts
- SLQ-32 HVS Swaps
- SLQ-32 HVS Signals

Process XSD for Troubleshooting Procedure Representation, XSL for transformation
4 PL/SQL Functions

createSession
getSession
getSessionID
updateSession

SaveSession.java

1 Java Stored Procedure

Troubleshooting Infrastructure - 3

FaultSession XSD for Captured Fault Session Representation

Sample FaultSessions

1 SCHEMA
15 Sample XML Instances

SLQ-32 HVS Subtest 1 Special Procedures
Troubleshooting Data Flow

KPControl
ParseProcess
store FaultSession XML
saveSession
Scenario XML
smartTable XML
Relational Tables
getNextEvent
access
getFeedback
access
Relational Tables
genXML
genXML Event XSL
access
Event XSL
store FaultSession XML
access
Scenario XML
smartTable XML
Relational Tables
access
How does Purdue’s KPS FY04 support captured Fault Session viewing and analysis?

- XML representations for captured fault sessions (FaultSession)
- Web-based session viewer interface (web client)
- Web-based fault session browsing and selection
- XSL transformation for fault session viewing (SessionViewer XSL)
- JSP interface for fault session action analysis data
How does Purdue’s KPS FY04 support captured Fault Session viewing and analysis?

- Relational tables for optimal session mining representation
- XSL transformations for fault session mining preparation
- Ship asset representation
- Ship-Fault-Action based analysis and aggregation
- Action-based troubleshooting feedback
- Action-based session viewer feedback by Ship, Ship Class and Fleet
- Mining includes action statistics, diagnostic sequence analysis, action trigger analysis, part replacement analysis, part-fault history analysis
### Fault Session Infrastructure - 1

#### 4 Java Stored Procedures
- Miner.java
- getSession.java
- FStransform.java
- mergeSession.java

#### 12 Support Relational Tables
- Ship
- PartLog
- PartShip
- PartFaultShip
- Part
- CurrentNode
- FaultSession
- Action
- Action Parameter
- ActionLink
- EventNode
- NodeLink

#### 1 Schema 15 Sample XML Instances
- FSTransform1
- FSTransform2
- FSTransform3
- FSTransform4
- EventFeedback
- SessionViewer
- XSL
- SLQ-32
- HVS
- Subtest 1
- Sample FaultSessions

**FaultSession XSD** forCapturedFault Session Representation, **XSL** for transformation
Fault Session Infrastructure - 2

Web Client Interface & Communication:
- SessionViewer.jsp
- sessionList.jsp
- sessionView.jsp
- sessions.jsp
- navigate.jsp
- StoredProcedure Call.java
- CraneQuery.java

1 Java Program:
- SessionGenerator.java
  - Scenario Session
  - Event Node
  - Probability Compiler
  - allPossiblePaths

Fault Session Support:
- transform
- analyzeEvent
- updateAll
- retrieve
- merge
SME FAULT SESSION REPORT

Ship: DECATUR  Operator: Brian Townsend  Date: 2003.07.06
Start Time: 10:20:00  Total Time: 110 min.
Reason: Weekly Maintenance
Observations: ---

Action : 1
Action: Look Up Fault
Observations: ---
Date: 2003-07-06  Start Time: 10:20:00  Total Time: 20 min

Action : 2
Action: Swap 3A5A2 and 3A5A14
Observations: ---
Date: 2003-07-06  Start Time: 10:40:00  Total Time: 5 min

General Parts History

<table>
<thead>
<tr>
<th>Part SRA</th>
<th>Last Accessed</th>
<th>Last Replaced</th>
<th>Compr Replaced</th>
<th>Average Lifetime</th>
<th>Replaced Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>3A5A2</td>
<td>Sep 8, 2004 at 02:14:44 EST</td>
<td>Sep 3, 2004 at 01:10:29 EST</td>
<td>3</td>
<td>93.0</td>
<td>16.7%</td>
</tr>
<tr>
<td>3A5A9</td>
<td>Sep 8, 2004 at 02:10:43 EST</td>
<td>Sep 8, 2004 at 01:20:43 EST</td>
<td>3</td>
<td>2.5</td>
<td>16.7%</td>
</tr>
<tr>
<td>3A5A10</td>
<td>Sep 8, 2004 at 01:19:44 EST</td>
<td>Sep 3, 2004 at 15:33:45 EST</td>
<td>3</td>
<td>180.0</td>
<td>16.7%</td>
</tr>
<tr>
<td>3A5A11</td>
<td>Mar 10, 2004 at 11:40:39 EST</td>
<td>Mar 10, 2004 at 11:40:39 EST</td>
<td>2</td>
<td>0.0</td>
<td>11.1%</td>
</tr>
<tr>
<td>Tasking</td>
<td>Deliverables Specification</td>
<td>Deliverables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 1 High Performance Knowledge Base</td>
<td>Knowledge Base Infrastructure</td>
<td>KPS Infrastructure code successfully installed on the Crane Testbed August 27, 2004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design and implementation of entire data layer (xsd, xml, xsl, supporting relational tables, script) and knowledge base modules for troubleshooting and fault session capture. Design and implementation of client, server, communication components following a 3-tiered architecture.</td>
<td>System documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Power point presentations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Extraction of Shipboard System from Shore System</td>
<td>Design documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design of component for synchronization of shipboard and shore-side KP systems</td>
<td>Power point presentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Impact Assessment of Windows Platform</td>
<td>KPS code installed successfully on both Unix and Windows platforms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port of Unix platform KPS to Windows platform. Assessment of SQL Server.</td>
<td>SQL Server Lessons Learned documentation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dynamic Maintenance Infrastructure</td>
<td>Online Troubleshooting Infrastructure code successfully installed on the Crane Testbed August 27, 2004. Delivered system supports more than 4 faults.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Now known as online troubleshooting. Data, metadata, support data and component code for 4 SLQ-32 HVS faults. Encompasses xml, xsl, database java code and GUI client code. Includes sailor interaction window, flowchart and knowledge data feedback GUI.</td>
<td>System documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tasking</td>
<td>Deliverables Specification</td>
<td>Deliverables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 2 Data Mining</td>
<td>3M Database Analysis and Mining</td>
<td>Power point presentations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Covers analysis and mining of the 3M database data, including its use in analysis and mining within KPS.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Synthetic Session Generator</td>
<td>Session Generator code successfully installed on the Crane Testbed August 27, 2004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design and implementation of a session generator that build a KB layer for testing and validating the data mining.</td>
<td>Power point presentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infrastructure for offline Fault-Session Mining</td>
<td>Fault Session Mining Infrastructure code successfully installed on the Crane Testbed August 27, 2004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design and implementation of the underlying infrastructure to support analysis and mining of actions and fault-sessions. Includes data layer (xml, xsl, supporting data, scripts) and code for client and server to support capture, transformation, analysis, processing and feedback.</td>
<td>System documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session and data mining Viewers and GUIs</td>
<td>SessionViewer code successfully installed on the Crane Testbed August 27, 2004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design and implementation of web-based viewers and other graphical interfaces for the data mining component. This includes session viewers and mined data viewer.</td>
<td>Power point presentations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infrastructure for Offline Session-stream Mining</td>
<td>Design documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Now known as Troubleshooting Session. Design of the underlying infrastructure to the analysis and mining of the troubleshooting session.</td>
<td>This task has been modified. It now requires design for KPS support of Sailor-SME queueing/transmission/processing/viewing rather than mining.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Tasking

<table>
<thead>
<tr>
<th>Tasking</th>
<th>Deliverables Specification</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 3 Query Processing and Knowledge Retrieval</td>
<td>Query Processing Infrastructure</td>
<td>SessionViewer with Knowledge Projection viewing code successfully installed on the Crane Testbed August 27, 2004</td>
</tr>
<tr>
<td></td>
<td>Design of the underlying infrastructure to support the query, search and retrieval of data from the Knowledge Base.</td>
<td>Online Troubleshooting with Knowledge Projection feedback code successfully installed on the Crane Testbed August 27, 2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Online Troubleshooting with linkage infrastructure code to access external technical content successfully installed on the Crane Testbed August 27, 2004</td>
</tr>
<tr>
<td>Document</td>
<td>Date</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Session Generator</td>
<td>02.15.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>KPS Architecture</td>
<td>02.03.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>Fault Session Mining</td>
<td>03.20.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>3M Text Mining</td>
<td>05.19.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>KPS Functional</td>
<td>02.03.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>Ship Shore Synchronization</td>
<td>06.24.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>Lessons Learned SQL Server</td>
<td>01.05.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>FY04 Purdue KPS Fault Coverage</td>
<td>08.30.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>KPS External Content Linkage</td>
<td>06.30.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>KPS Technical Content</td>
<td>06.30.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>TSS Pre-Design Document</td>
<td>08.05.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>TSS Event Block Design Document</td>
<td>08.05.2004</td>
<td>PDF</td>
</tr>
<tr>
<td>Purdue KPS 2004</td>
<td>09.10.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>Purdue KPS Troubleshooting</td>
<td>09.10.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>Purdue KPS Fault Session Mining</td>
<td>09.10.2004</td>
<td>PowerPoint</td>
</tr>
<tr>
<td>The Knowledge Projection Vision</td>
<td>09.10.2004</td>
<td>PowerPoint</td>
</tr>
</tbody>
</table>
# FY04 Baseline Document Schedule

<table>
<thead>
<tr>
<th>Document</th>
<th>Date</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>KPS Installation Guide&lt;br&gt;<code>Unix and Windows Platforms</code></td>
<td>10.08.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>KPS Requirements Specification Update</td>
<td>10.08.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>KPS Design Specification Update&lt;br&gt;<code>Database Design, Component Design and Data Flow Specification for the FY04 KPS</code></td>
<td>10.08.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>KPS Functional Specification Update</td>
<td>10.15.2004</td>
<td>WORD</td>
</tr>
<tr>
<td>KPS System Architecture</td>
<td>10.29.2004</td>
<td>WORD PowerPoint</td>
</tr>
</tbody>
</table>
Purdue KPS Team

**Principal Investigators**
- Ann Christine Catlin
- Chris Clifton
- Ahmed Elmagarmid
- Arif Ghafoor
- Sunil Prabhakar

**Research Staff**
- Mirette Marzouk
- Mourad Ouzzani

**Graduate Students**
- Mohamed Ali
- Rafae Bhattir
- Jason Catlin
- Mohamed Elfeky
- Hazem Elmeleegy
- Hicham Elmongui
- Mohamed El Tabakh
- Thanaa Ghanem
- Ammar Massar
- Ercan Mehmet Mnergiz
- Omar Alrawi
- Javed Siddique
- Yicheng Tu
- Yuni Xia
Purdue’s FY04 BAA promised to deliver 4 SLQ-32 HVS faults. How many faults does KPS Online Troubleshooting support?

A major change in the functionality, structure and goals for the KPS “troubleshooting session” was made at the end of FY04.

- Identify the impact of this change to the KPS design deliverable.

- Describe the impact of this change on the existing infrastructure for FaultSession capture, viewing, analysis and processing.

- Can Purdue present a design for the “troubleshooting session” as it is now defined?

- Any other questions?