Abstract
As the demands on hospital infection control teams increase, it becomes less efficient for them to use paper-based surveillance methods. The existing electronic infection control surveillance system at our largest facility was not designed to support a multi-hospital model. Our goal was to redesign the application using generic, open source technologies, and make it flexible enough to support the infection control surveillance needs of the entire enterprise.

Keywords: Microbiology, Infection Control, XML

Introduction
GermWatcher™ was developed in 1993 as an electronic alternative to manually reviewing microbiology cultures at Barnes-Jewish Hospital (BJH). Designed as a single facility application, it was unable to accommodate the needs of other hospitals in the BJC network that were using less efficient paper-based microbiology surveillance methods. Late in 2003 we began redesigning GermWatcher™ with an emphasis on flexibility and scalability within the enterprise.

Methods
Data Acquisition: Microbiology data are complex, with multiple tests, organisms and susceptibility data encompassed in a single result. In addition, hospitals within BJC HealthCare use different lab systems, and finding a standardized way to extract the information was a challenge. Using HL7 messaging was not practical, because some lab systems were not HL7 compliant and would have required expensive additional interfaces. We arrived at a more cost-effective approach using XML messages created by leveraging the reporting capabilities of the individual lab systems. A listener on a queue (IBM MQSeries) triggers GermWatcher™ to begin processing whenever a new message arrives, resulting in real-time data capture.

Standard Codes: As an enterprise application, it was important to ensure that codes from each facility were translated to a standardized set of codes within GermWatcher™. This provides the foundation for shared rules and the ability to compare data seamlessly across the enterprise. Organisms, specimen sites and antibiotics were all mapped to SNOMED concepts.

Rule Engine: The rule engine was implemented using open-source technologies. The JSR-94 Java Rule Engine API provided the framework, and allowed us to group rules into customized rule sets for each facility. We used the reaction rule described by the Rule Markup Initiative as the basis for our XML rule schema. It was modified slightly to allow for reusable code fragments and calls to custom methods that were too specific to be included as generic operators. The rules and rule set definitions are stored in a relational database (Sybase), which simplifies maintenance.

User Interface: A web interface was developed using Java Server Faces. This eliminated the need to support a client-based application at multiple facilities. Users are authenticated via an LDAP server, and a firewall prevents unauthorized access from outside the network. The interface organizes the cultures by the rule classification, which helps the infection control specialists streamline their workflow. It also provides ad-hoc query capabilities to retrieve information from the repository.

Discussion/Conclusion
Development of the enterprise ready GermWatcher™ application was completed in December 2004. It is currently deployed as a pilot at BJH and is scheduled to replace the current system at the end of March 2005. Expansion to two additional BJC facilities by the fall of 2005 is planned, and more facilities are being evaluated for 2006 deployment.

References