

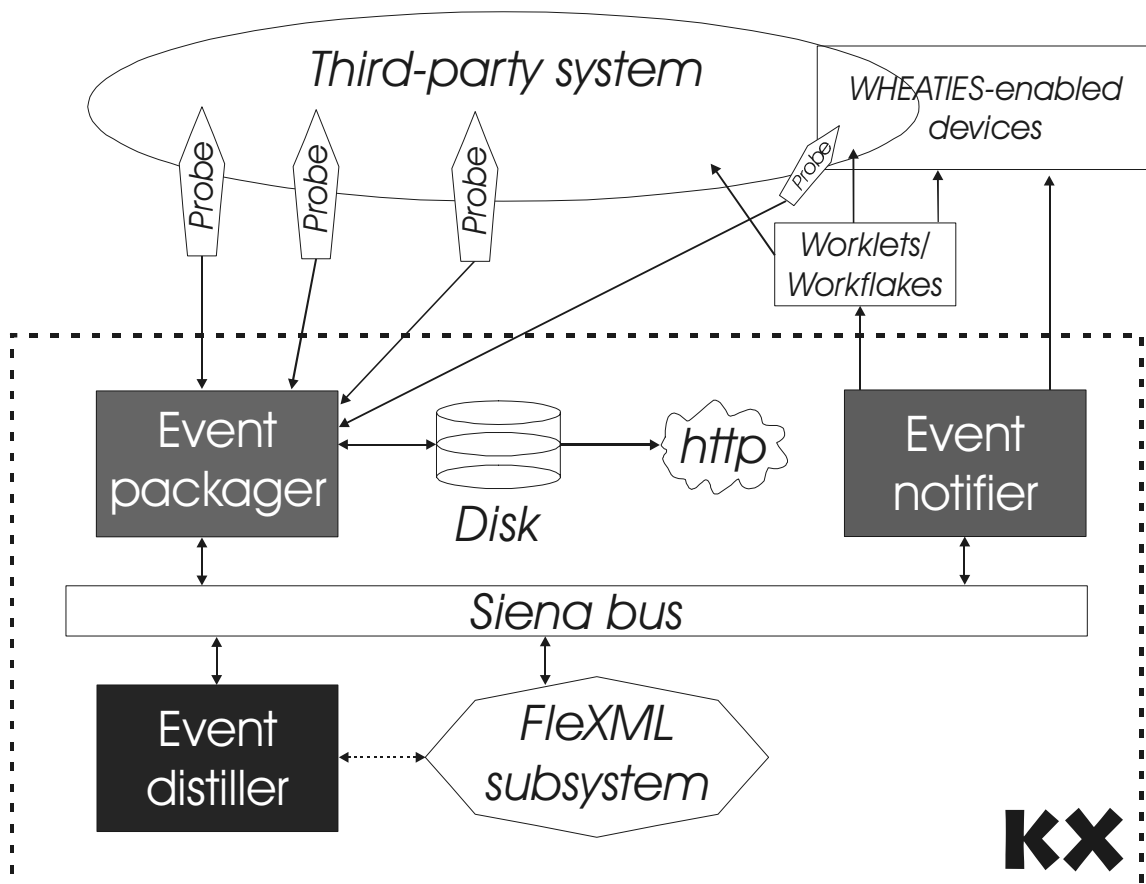
Introduction

The *XML-based Universal Event Service (XUES)* consists of three main services that collectively provide the necessary event handling for the Kinesthetics eXtreme (KX) real-time monitoring architecture. XUES is responsible for introducing events into the system from probes (via the *Event Packager*, or EP); recognizing significant events and complex temporal event patterns (via the *Event Distiller*, or ED); and dispatching meta-events, worklets mobile agents (see separate flyer) or other gauge-indicated material (via the *Event Notifier*, or EN).

XUES as a whole is a powerful and flexible run-time monitoring framework, which can be teamed with (re)configuration facilities such as **PSU**'s worklets, to support dynamic system adaptation and assembly.

System Description

The XUES services interoperate with other KX components (and third-party systems, via XML/FleXML and/or Siena) to provide a complete framework for gauging founded on event filtering and aggregation.



The **Event Packager** inserts monitoring capabilities into third-party systems via any of the variety of DASADA probe technologies. Probes' outgoing raw data streams are transformed into Siena-compliant events. EP also spools the event stream to disk in an intelligent fashion, which can then be presented using an http server to interested parties - e.g., "latecomer" gauge subsystems that may have been restarted or added to the monitoring system later on, applications that need to handle particularly large events such as streaming video that are unsuitable for the Siena bus, or for off-line data mining.

The **Event Distiller** subscribes to events potentially matching portions of any of its gauge rules - with possible assistance from the FleXML schema discovery, schema fragment composition, and incremental tag processing subsystem, recognizes complex event patterns, and emits the corresponding meta-event(s). Multiple EDs dispersed across the Siena bus automatically chain through event/meta-event publish/subscribe relationships to maintain particularly sophisticated gauges. ED is capable of recognizing temporal constraints among events from the same or multiple sources, and can even notify if a specified event or event pattern has *not* occurred within a given temporal restriction.

Finally, the **Event Notifier** acts as the converse of the packager: it packages meta-events and payload data into forms useful for external systems, whether they are directly input to the third-party system(s) being monitored by probes, TRIKX (TRansitional Interface to KX) portal-like frontends, or WHEATIES (Wimpy Handheld-Accessible Tracking, Identification and Error correction System) handheld devices. EN will soon interact with Workflakes to instantiate and coordinate Worklets in their guise as "gaugents" - to complete the KX feedback loop through process-aware system reconfiguration and repair.

Features

Broad interoperability. All XUES components natively support Siena, XML, and FleXML, and can easily be converted to other event formats. ED's gauge rule engine is replaceable, e.g., with Acme gauge Java objects. Event Notifiers can be built to match a target system's custom adaptation facilities.

On-the-fly runtime rule engine. XUES is designed from the ground up to support run-time monitoring systems (such as KX, but other approaches as well). Additionally, flexible spooling by the Event Packager allows for new components to "catch-up" with past events: EP is also generally useful for logging and data mining (e.g., for intrusion detection). The Event Distiller supports dynamic run-time changes to its gauge rule base (i.e., addition, deletion or modification of existing rules), to focus on specific distributed computing protocols, functionality problems or performance issues.

Lightweight, flexible, and embeddable. XUES services are designed to be integrated into existing systems with ease. They can be attached to another event bus besides Siena (in fact they originated on **PSU's** groupspace controller bus), or can be directly embedded within the target system; the components may be used separately or together in tandem.

General Specifications

All XUES components are written using 100% Pure Java 2, but can interoperate with C++ probes and event notification recipients. XUES has been tested on Sun Solaris, Linux, and Windows NT/2000 systems, with (limited) events additionally to/from PocketPC and PalmOS.

Availability

<http://www.psl.cs.columbia.edu/software/download>.

Copyright 2001, Trustees of Columbia University in the City of New York. All Rights Reserved.