Xerox Research Centre Europe

xrce

A Xerox Research & Technology Organization

# fact-sheet

# STITCH :
## Coordination Language Facility

## Overview

STITCH is an advanced integration tool that enables programmers to integrate incompatible applications to provide new services on a global basis. STITCH combines, within a single application, software that runs on different platforms written in different programming languages and communication protocols.

STITCH allows organisations to develop necessary business applications faster than ever before by co-ordinating existing elements, such as legacy applications and web services as well as facilitating the integration of application-specific packages such as e-commerce, HR or finance.

STITCH benefits from the CLF (Coordination Language facility) project and is the mature outcome of more than five years of in-house research and development at the Grenoble Lab of XRCE (Xerox Research Centre Europe). CLF-based applications have already been successfully deployed within the Xerox Corporation around the themes of distributed document and knowledge management, and new applications are under development.

STITCH is the preferred choice of environment for building highly distributed (inter) enterprise applications with strong demands for reliability, scalability and flexibility. These characteristics are usually found in application domains such as web services, document management, workflow processing and e-commerce. STITCH greatly simplifies the task of building such applications by providing two principal capabilities: a scripting capability that provides a very high-level language for building applications from components, and an encapsulation capability that makes it easy to reuse existing applications and services by allowing them to act as STITCH components and thereby be controlled through CLF scripting.

## Two outstanding capabilities

### High-level scripting language

The CLF scripting language is used to define application logic and to build the necessary "glue" between software components. It allows complex interactions between distributed software components to be expressed in a simple and compact way. The resulting coordination scripts, interpreted by a patented runtime mechanism, can be dynamically modified for an unprecedented level of flexibility, tailorability and ease-of-use. CLF scripting pilots the notion of resource-based programming using a rule-based approach. This leads to drastic reductions in size and efforts.

### Lightweight encapsulation framework

Building STITCH components is greatly simplified by the straightforward *STITCH component model* that defines in very clear terms how a well-behaved component should interact with its environment. Encapsula-

tion of existing web services and applications is easy and can be done with zero or very minor changes to the source code thanks to the STITCH encapsulation framework. This framework comprises a minimal set of libraries and programming rules that allows developers to promote components to first-class STITCH citizens. From this point they may be manipulated using the CLF scripting language with all the associated benefits.

## Pre-packed solutions for open distributed systems

### Transaction-awareness

A major feature of STITCH is that it is transaction-aware. Transaction processing is the software technology that makes distributed computing reliable. It guarantees the synchronization of modification to data across different components, helps manage concurrent and competing accesses to resources from many users, and provides basic capabilities to achieve fault tolerance. Transaction processing is also a very natural and simple model for end users of information systems. The price to pay for this is usually increased complexity for developers and system integrators, making transactional systems expensive to buy, operate and modify. As a result, such systems are felt to be inflexible and "heavy". One aim of STITCH is to provide the power of transaction processing in combination with the flexibility and simplicity of modern object-oriented scripting.

### Heterogeneity

It is widely acknowledged that tomorrow's Internet-enabled applications and web services will have to deal with heterogeneity at various levels. From communication protocols to operating systems, from integration platforms to data representations, every layer involved in a large-scale software application comes in different, often incompatible, types and flavours. Still, software entities need to interoperate, and interoperation between heterogeneous parts involves ad-hoc programming efforts. To completely eliminate such ad-hoc development is almost certainly an unattainable goal, yet significantly reducing the amount is by itself a challenging objective. STITCH has been designed with heterogeneity in mind, and supports, within a single application, components built on different platforms, with different programming languages and using different communication protocols.

### Application control

Issues related to application deployment and monitoring are very relevant to distributed settings. While the aim is that software components be widely distributed, the facilities provided to system administrators need to be fully available at the network nodes from where the applications are managed. Providing the means to easily configure, start, stop and replace the various components of a distributed application from a single node of a network is a hard task, in particular if the network spans across several administrative domains. The STITCH deployment tools provide system administrators with the necessary means to do this in a controlled manner. Furthermore, powerful monitoring tools make it possible to inspect the internal states or to trace state changes of the various components at execution time, in order to check the correctness of their behaviour.

### Example Applications

**Yaka** is an application that provides users with automatic document notification and delivery from multiple heterogeneous sources, such as document management systems, file systems, web services, etc.. Topics are defined in YAKA, such as CRM, e-commerce etc. and users subscribe to the topics they are interested in. Whenever a new document is published on a given topic, Yaka looks up the list of subscribed users, notifies each one of them of the availability of the document and, upon request, delivers it according to personal preferences. Notification is provided by email along with an automatically generated summary of the document so that the reader can more easily identify its specific interest. Delivery of the full document can be by fax or e-mail, but also by printing to a local printer or even by an automatic insertion of the document into a local, personal document repository. Specific STITCH features in this application are its ability to manage heterogeneous sources and delivery media, the ability to detect and react to new documents, and the flexible definition of document transformations and handling thanks to the scripting capability. The compactness of CLF scripting is illustrated by the fact that the core integration and business logic of YAKA is expressed in less than twenty CLF rules.

**X-folders** is a more sophisticated electronic version of the traditional office circulation envelope. It allows users across distributed virtual organizations to flexibly and effectively collaborate and share documents. By implementing the concept of Electronic Circulation Folders (ECF) as a "virtual" storage space for documents, users store and access documents in document repositories, with all handling and housekeeping necessary to manage the collaborative use transparently handled in the background. When an ECF is created users can attach a simple routing process that defines its corresponding workflow. Workflows are lightweight and may be modified at any stage by the user currently in control of the ECF. Reliability is assured through the STITCH capability of migrating documents across multiple heterogeneous repositories and firewalls, while preserving access control. The library of reusable components and easy definition of workflow descriptions through CLF scripts make this application particularly effective. The flexibility that lets users dynamically modify workflow processes comes from the dynamic nature of how STITCH manages and executes scripts. Other applications have been implemented in the area of multi-party negotiation and business-to-business e-commerce.

## Positioning

STITCH shares a number of common capabilities with object-based middleware, in particular CORBA, J2EE and Jini. However, its focus is different in that it is aimed towards the coordination between the applications and IT infrastructures of various businesses, rather then being an application development platform. As a consequence it differs from other middleware in some significant aspects. STITCH provides a high-level rule-based scripting language, designed purely for the purpose of systems integration. This language gives access to several core functions, without the need to address them explicitly. Therefore by hiding their intrinsic complexity STITCH makes the building of applications faster and less error prone.

Core functions include:

- Rapid application building through the coordination of autonomous applications
- Guaranteed loose coupling through dynamic matching of service requests and offers
- Exploration of multiple offers in parallel
- Transactional system semantics

These features can be implemented to either bridge groups of components integrated by other technologies (e.g. CORBA and Jini) or integrate the components themselves while they continue to function in their original environment. This is a unique capability which enables the creation of an unlimited number of end user services.

## In brief

STITCH was designed from the very beginning to tackle many of the difficult problems encountered by system integrators and developers of large-scale distributed applications and web services. It provides the means to:

- Ensure interoperability across heterogeneous environments;
- Leverage the power of transactions while abstracting from their complexity;
- Control distributed applications through a specific set of tools.

It manages to do this, while providing a very simple and straightforward model to programmers and system integrators. An important goal has been to put all the features of STITCH in the hands of non-experts of distributed systems. This means that system integrators dramatically improve their efficiency in producing quality applications.

## Technical data

The STITCH infrastructure consists of:

- A high-level scripting language
- A distributed object computing runtime
- Encapsulation libraries and APIs
- A sample of STITCH -compliant reusable components
- Deployment and monitoring tools
- Comprehensive documentation, including a hands-on tutorial

The STITCH implementation is available for the following platforms: Solaris, Linux, and Windows NT. It is based on open standards, is self-contained and does not rely on or require the installation of any vendor-specific technology. Encapsulation libraries are available for Java and Python, and may also be used to encapsulate components written in other languages, through any of the following open communication protocols: HTTP, IIOP, RMI.

## Research

STITCH is the result of years of research, and has been widely presented and discussed in the communities of distributed and coordinated systems research. Nearly 20 articles have been published in international journals and conferences between 1994 and 2001. The efficient implementation of distributed transactional scripting with the CLF scripting language has been patented.