Xerox Research Centre Europe
xrce
A Xerox Research & Technology Organization

Research Fact Sheet

Circus-DTE

Document Transformation Environment

*"Xerox's **Circus-DTE** programming technology for document transformation provides programmers with an attractive user-friendly language and helps software companies with a breakthrough language technology to be more competitive and to increase their market share."*

Abstract

The generalization of networking and the Internet standard over the past 10 to 15 years has opened up the modern business environment to vast amounts of information in the form of electronic documents. To realize the full potential of this wealth of information, it is important for organizations to be able to reorganize and combine documents in ways that will best suit their needs. This complex task represents an increasingly difficult challenge for document transformation software.

Document transformation technology is not new, and indeed a number of solutions already exist, but until now software engineers have basically had to choose between general purpose low-level languages and more specific, abstract high-level languages, each group fulfilling a particular need, but each with its drawbacks. Xerox's Circus-DTE technology bridges the gap between the two approaches, and in a sense offers the best of both worlds. Circus-DTE is specific enough to deal with today's complex transformation problems but also general enough to be able to adapt to new challenges tomorrow. As such, it represents a breakthrough in language technology and an opportunity for software companies to enhance their competitivity and increase their market share.

Introduction

With the rapid growth of networking and the Internet since the early 1990s, and the advent of widely accepted standards such as XML, the electronic document has become truly ubiquitous in today's business world. Multiple sources provide a wealth of electronic information and, thanks to electronic mail and the Internet, access to this information has never been easier.

The challenge facing information technology now is how to provide users with the means to organise the vast amounts of information available according to their specific needs. It is particularly important, for example, for users to be able to reorganise and combine documents from different sources and with different structures. This task requires sophisticated document transformation applications and, given the increasing variety and rapidly changing nature of electronic documents, it represents an ongoing challenge for software engineers. What is required is a tool which is capable of programming the complex document transformations needed today and yet flexible enough to deal with the requirements of tomorrow. The researchers at Xerox labs, honouring their long-term commitment to furthering the development of language processing technologies, have designed and built just such a tool, Circus-DTE. Circus-DTE provides programmers with an attractive and powerful language, while for software companies it represents an opportunity to take the lead in a rapidly developing segment of the IT market.

For software companies, having a breakthrough language technology is crucial. It allows them to shorten time-to-market, to control development costs and to economize on future software maintenance costs. For a given development budget, it means that they can increase the number of functions addressed by the software and thus its competitivity. It also reduces the business and technical risks for software companies, enabling them to face new programming challenges with greater confidence. This too increases their competitivity and provides them with the opportunity to increase their market share.

Business Problem

In designing programming languages, the abstraction level is a key factor. For example, Java is a general purpose language, and therefore handles general abstractions such as objects, exceptions and concurrency control. Using such a language to process structured documents, engineers can solve a wide range of problems, but to do so they have to develop complex algorithms from scratch or reuse complex libraries. A general purpose language does not provide assistance for such tasks because it does not embed the knowledge of the application domain.

At the other end of the spectrum, XSLT and XQuery, two transformation solutions from the W3C, are specialized in XML document transformation, and therefore propose more specific, abstract data and execution models. The language then becomes "sharper" and better adapted to solving a restricted class of problem. However, it is usually very cumbersome when used for processes which deviate even slightly from the original purpose.

Today, therefore, engineers generally have to choose between a very high-level approach or a very low-level approach typically based on a general purpose language extended with a tree manipulation toolkit (e.g Java and DOM). Very often, then, the choice does not provide a satisfactory solution.

Xerox Solution

Circus-DTE is a specialized programming language positioned between the high-level and low-level approaches. It addresses general structure transformation problems, relying on a cutting-edge type system which embeds knowledge about generic data structure modelling and handling. It can be used for all transformation applications that involve potentially complex information flows, including:

- Document content processing
- Internet publishing
- Data base to XML conversion
- format adapters
- transformation engines (static or dynamic)
- wrapping
- content analysis pre/post processing
- document filtering
- document rendering
- document customization and repurposing

Circus-DTE has been designed around the paradigm of structure transformation, particularly suited to language processing or the transformation of structured documents. Using this approach, a programmer solves document transformation problems through a succession of content processingdata base to XML transformation steps that together provide a safe route to a solution. When the transformation is especially complex, these intermediate steps are a natural way to break down the problem into its simpler, constituent parts. Each transformation step consumes and Technology produces data structures for which the structural properties can be stated and verified. This makes it easier to characterize the whole chain, thus enhancing rigor and predictability.

Although its design is based on modern theoretical approaches, Circus-DTE is targeted at engineers and is particularly easy to learn thanks to a few carefully chosen design principles:

- User-friendly but powerful type system: expression of structure schemes and static/dynamic verification.

- Minimal but expressive set of constructs: simplicity, legibility, flexibility and clarity.

- Powerful "structural" pattern matching: three fundamental operations combined in one operator.

- Composition operators: help in reducing time and memory overheads while at the same time simplifying component design and reuse.

Key benefits of Circus-DTE

The key benefits of Circus-DTE may be summarised as follows:

1. Innovative and efficient

Circus-DTE is an innovative patented technology specifically designed to address the complex processing needs of today and ready to meet those of tomorrow. With a syntax and semantics as compact as those of higher-level languages, and as general as those of lower-level languages, the advanced design of Circus-DTE makes it scalable to deal with complexity over time, meaning a better return on overall development costs.

2. Elegant yet practical

Programmers enjoy using Circus-DTE not only because of its compactness and user-friendliness but also because of the outstanding efficiency of its elegant constructs. Circus-DTE is above all a creative language.

3. Universal

Circus-DTE embeds Unicode and XML processing facilities but can also support an unlimited set of formats, ranging from HTML to PDF and even binary format. Circus-DTE's original structural pattern matching is also applicable to all data types.

4. Early error detection through an effective type system

The Circus-DTE type system has demonstrated its capacity to detect programming errors at a very early stage, thus preventing many long and costly test cycles. It allows very precise modelling of data structure, including the use of recursive schemes and enumeration sets.

5. Rich execution model

Circus-DTE embeds new connectors for assembling transformations either sequentially or concurrently. To this end it features a coordination memory "à la Linda" with a compact set of coordination primitives.

6. Component orientation

Circus-DTE is designed around an original programming abstraction called PAM (Polymorphic abstract machine), a procedure having only one input parameter and one output parameter. PAM is ideally suited for programming small-grain transformation algorithms, which can then be combined through a rich set of connectors. Thanks to a patented composition calculus, Circus-DTE brings component reuse to a new level.

7. Testing facilities

The Circus-DTE compiler features verification clauses that are executed at compile time. These statements allow the building of robust code and make software maintenance easier.

Executive Summary

Circus-DTE provides an innovative, elegant and powerful technology capable of meeting not only the complex processing needs of document transformation today but also the challenges of tomorrow as requirements change. For programmers, Circus-DTE is easy to use, creative and efficient. For software companies, Circus-DTE provides an opportunity to increase the competitivity of their products, and their market share, by shortening time-to-market, controling development costs and reducing future software maintenance costs.

Technical Details

1. Overview

Circus-DTE is a mixture of functional, imperative and declarative programming styles. It is a type-safe compiled language with an embedded interpreter for compile-time evaluation of testing clauses. Circus-DTE incorporates structural matching operators that operate on all types of data. Matching operations involve type-checked filters which are combined up to arbitrary complex levels.

Circus-DTE also offers a *Linda*-like Coordination Memory. Such a model relies on a few basic synchronization primitives and an associative memory that together simplify complex synchronization schemes. The Linda experiment demonstrated that concurrent transformation processes remain highly reusable thanks to the indirection imposed by the model. Processes do not explicitly specify a recipient to exchange information.

2. Type system

Polymorphism: by subtyping and operators

Type safety (no execution errors due to typing issues)

Strong typing, static or dynamic

Explicit type operators: intersection, record overloading, non-discriminating union

Generic types: top, bottom, polymorphic enumeration sets

Primitive types: (unbounded) integers, floats, unicode strings, byte string, boolean (three-state logic handles **true**, **false**, **unknown**)

Constructed types: tuples, sequences, multisets, simple and extensible polymorphic records, associative mapping

References: read only and read/write references, no null pointer abstraction

Functors: lambda functions and composable procedural abstractions (PAM)

3. Syntax

Few syntactic features, but highly combinatorial

- one universal iterato
- one **for .. in ... do** construct
- one sequential and one concurrent combinator (**;** and **||**)
- **if** ... **then** ... **else**
- cascading rules
- switch
- one variable declarator **var** name:type
- assignment, member access, field access, reference constructor, dereferencing

4. Semantics

Formally defined through Structured Operational Semantics, including concurrent operators. Formal semantics brings clarity - and confidence - to the language. It also facilitates the task of establishing important run-time properties which are crucial in certain sensitive areas. And finally it enables engineers and other researchers to work on the technology using rigorous bases and common notations.

5. Back-end and Code generator

The Circus-DTE compiler generates Python byte code and an intermediate format. A reflexive API makes Circus-DTE particularly suitable for programming various code generators. A JVM code generator is included in the distribution package. It runs independently of the main compiler.

6. Portability

Circus-DTE runs natively on a Python2.2 platform and offers a cross compilation into the Java Virtual Machine byte code. This means that Circus-DTE runs on all major modern architectures and operating systems.

7. Libraries

Circus-DTE proposes four different data models for handling XML and HTML documents:

- Inclusion trees (compact, very well adapted to top-down recursive descent), reference trees (flexibility for navigating in nodes), reference trees with namespaces, DOM.
- Networking primitives: unified through Uniform Resource Identifiers (file, http, ftp, mail)
- Ready-to-use, stand-alone components: analysis and generation of document types (DTD, schemas). Used to translate known constraints on a document (e.g. DTD) into equivalent Circus-DTE types (generation of Circus-DTE source code).

For more information on the Circus-DTE research project please contact:

Xerox Research Centre Europe
6, chemin de Maupertuis
38240 Meylan
France

Christer.Fernstrom@xrce.xerox.com
http://www.xrce.xerox.com

Tel: +33 (0)4 76 61 50 50
Fax: +33 (0)4 76 64 50 99