

A New Approach to Capturing the Behavioural Semantics of Large Scale Distributed Systems

Introduction

Businesses are defined by an interlinked contiguous collection of business systems each of which executes specific sets of business processes. Examples of business systems include financial management, customer relationship management, employee management, product procurement, and facilities management.

A business process is an assembly of interlinked, atomic activities each of which is represented by a service interaction. These interactions, or bindings, generate observable system behaviour. This behaviour is able to be captured and compiled into a continuously updating representation of the executing system's semantics.

This representation may be expressed using a standard graphical grammar notation, generating a machine readable ontology detailing the syntactic parameters of the system bindings.

Once generated the ontological representation may be presented as a system meta-model or used to provide binding specific micro-information.

Keywords: Business systems, business process, observable behaviour, continuously updating ontology

Problem

Extracting the system semantics of large scale distributed implementations is non-trivial. The conventional approach is to compile a static representation of historical activity by compiling event information from thousands of distributed log files.

In heterogeneous system, these files are invariably in numerous formats, including the XES, OCEL and CSV standards or may be in a non-standard format as is often the case with legacy implementations, such non-standard formats often failing to provide sufficient high quality, structured data to enable the system semantics to be extracted.

Regardless of the formats being employed the extracted samples from the log files need to be cleaned, formatted and standardised before an ontology can be generated. This is necessary as the extracted files will be found to have missing, duplicated or inaccurate information, with the native time-stamps made available by each of the sub-system's being in differing formats and varying levels of granularity.

Cleaning and standardising the extracted log files is a time consuming and resource hungry process, a problem compounded by the size, distribution and heterogeneity of large complex organisations.

Given the resource demands of extracting, cleaning and standardising the semantic information contained in the event log files, file sampling is used to contain resource demand.

Sampling generates a further level of complexity in the capture of the system semantics. The smaller the sample, less resource is required together with the increased likelihood of failing to capture the infrequently executing event paths; the larger the sample, the more costly the event log standardisation process with an increased likelihood of capturing, at least some of the infrequently executing sequences, within the framework that it is only possible to capture all executing sequences by capturing all services interactions.

HELIXsystem Process Assembler™

The Myrror Corporation's patent protected technology, the HELIXsystem Process Assembler™ ("Process Assembler") compiles a Digital Twin of any complex, heterogenous implementation. The Process Assembler is business system agnostic and may be deployed across any vertical.

The Process Assembler captures every service binding as these occur and assigns a standard format time-stamp to all observable system interactions. These interactions are sorted into their logical order and presented as an end-to-end system ontology.

This is achieved without changing the messages, installing agents on repositories, refactoring production, introducing a point of system failure or requiring any aspect of the production environment to be referred to QA.

The ontology generated by the Process Assembler continuously updates capturing implementation changes as these occur. This enables the Process Assembler to deliver the Digital Twin of the Organisation regardless of the complexity of the physical implementation.

This real-time representation of system state may be used to simulate and test proposed system changes for unintended consequences prior to such changes being introduced to production. Once such proposed changes are tested across the Process Assembler Digital Twin, the Process Assembler may be used to automatically generate the code required to introduce such changes into production.

The Process Assembler technology represents the convergence of the separate disciplines of Process Mining and Digital Twin Architectures enabling current state representations of run-time systems to be compiled into a continuously updating representation of the executing semantics.

The HELIXsystem Process Assembler™ is protected technology by US patent grant No 8,554,594 "Automated Process Assembler".