

# Runtime Process Governance

## Formal Transactional Conformance with the Process Design

In loosely coupled systems, complex business process utilise large numbers of co-operating services across multiple execution paths, often over long periods of time. To ensure the real-time conformance of each transaction with the required business process, it is necessary to continuously monitor and compare the service interactions against a formal description of the overall process design.



## Background

Every business is defined by its processes. Efficient processing identifies the competitive difference between one business and its peers, and it is efficient processing that determines the the long term success of a business.

Over the years, most reasonably sized companies have automated their key processes by building customised applications. More recently these custom built applications have been modularised. Whilst this simplifies the application design, obtaining visibility over the service interactions can be opaque. As system and process complexity compounds, maintaining visibility over the service interactions becomes progressively more difficult until a point is reached where no single individual has a comprehensive understanding of the overall system or the overall process.

Replicating the distributed system in a QA testbed is seldom possible, and even if the system could be reproduced, using sanitized test messages to reproduce all the possible execution paths available to a transaction instance is unlikely to produce meaningful results. Eventually a point is reached where the only method available to comprehensively identify the service interactions, is to monitor the production environment.

System monitoring assists, but is only a partial solution. In order to ensure every transaction executes in accordance with its intended process, the service interactions need to be validated against a process design. Only by monitoring against a formal description of the required process, can assurance be provided that a transaction has executed as required.

## The HELIXsystem Process Governor

Runtime Process Governance is the continuous monitoring and comparison of the interactions between co-operating services against a formal description of the process design. Runtime Process Governance contains the risk of a transaction failure by ensuring every transaction either executes without variance to the required end-to-end process, or should variances arise, these are immediately identified.

The HELIXsystem Process Governor delivers Runtime Process Governance by the real-time, continuous validation of all service interactions against a machine readable process description. When variances arise, immediate remedial action is able to be effected, as the location of the variance is identified both in terms of the services impacted and the point in the process cycle where the variance occurred. This enables even the most complex transactions to be managed on an exceptions basis.

The HELIXsystem Process Governor delivers Runtime Process Governance without the introduction of proprietary protocols, recompiling or re-architecting the services or impacting overall system performance. The HELIXsystem Process Governor is a generally available capability, able to be deployed across virtually any heterogeneous, loosely coupled system.

The HELIXsystem Process Governor is comprised of four components:

- Service Validators
- Process Design<sup>1</sup>
- Transaction Information Service
- Dashboard

The location of these components is illustrated in the following diagram:

---

<sup>1</sup> The Process Design may also be described as the Global Description

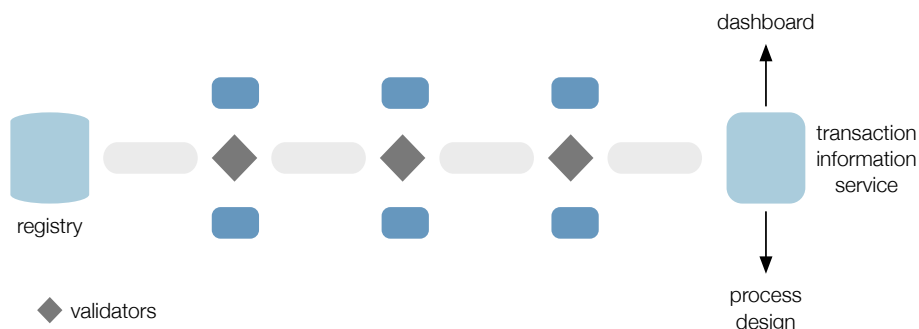


Diagram 1: Location of the HELIXsystem Process Governor components

## The Service Validators

The HELIXsystem Service Validators are service specific monitors. The validators are not embedded into the services, do not require any alteration to message structure and do not impact system performance. The function of the validators is to deliver an uninterrupted cycle of “monitor and compare” across every transaction instance.

The validators identify the inbound and outbound interactions for each of the services. These interactions are compared against the expected interactions as defined by the process design and variances identified. All variances are detected, including incorrectly sequenced exchanges, missing processing steps, missing interactions within a processing step, duplicated interactions, unexpected exchanges, and incorrect message content. By capturing the time stamp on each interaction, stalled transactions and system bottlenecks can also be immediately identified.

The validators provide granular information on the precise location of non-complaint events, both in terms of the services impacted and the sequencing position in the overall processing cycle. This enables precision-perfect remedial action to be taken, without the need to recompile the system behaviour from multiple, widely distributed log files. This contains the demands on the system administrators, and eliminates the labour intensive, often highly distracting “immediate action” drills associated with system outages.

## The Process Design

The HELIXsystem Process Design is derived from the originating business requirements. These requirements are modelled into a series of sequence diagrams, each representing an execution path available to a transaction instance. These diagrams detail the interactions required to complete the path, and their ordering. A declarative, standards based approach is then used to generate the endpoint behavioural descriptions of each of the services<sup>2</sup> engaged by the described path.

The sequence diagrams are aggregated to provide an end-to-end description of the process. The assembly of these execution paths represents an aggregation of the endpoint behavioural descriptions. This provides a comprehensive behavioural description of the overall process from a neutral or “global” perspective.

The resulting process design unambiguously describes how the services are able to communicate. The runtime interactions may now be validated against the process design to ensure the continuous conformance of the executing transactions.

---

<sup>2</sup> Details of how this is achieved is provided in the related whitepaper “SOA Best Practices”



## The Transaction Information Service

The HELIXsystem Transaction Information Service provides granular detail on the runtime behaviour of each transaction, and compiles an auditable record of the service interactions and the variances.

The Transaction Information Service delivers “snapshot” reports detailing overall system performance. These cover such areas as total transaction throughput, end-to-end system response times, individual service processing times, aggregated system messaging faults, load spikes, and system performance against benchmarks.

The Transaction Information Service may also be configured to provide “on-demand” reports, detailing the specific execution history of a transaction and the current state of a transaction. This latter capability is particularly useful when dealing with long running transactions as the current state of the transaction is able to be understood in the context of the overall process design.

## The Dashboard

The HELIXsystem Process Governor Dashboard is coupled to the Transaction Information Service. The dashboard provides visibility over all the processes executing across a single infrastructure, with exceptions being reported to a single consolidated view. This enables multiple processes to be managed concurrently.

The dashboard provides both hierarchical and tabular representations of the processes and delivers intuitive search and diagnostic capabilities. This reporting ability locates missing exchanges, unexpected messaging conditions, stalled transactions and service faults easily and with pinpoint accuracy.

## Summary

THE HELIXsystem Process Governor delivers Runtime Process Governance by ensuring the overall runtime behaviour of a business process executes in accordance with the “global” description of the originating requirements. This is achieved regardless of the complexity of the process, the number of co-operating services engaged, the number of execution paths involved, or the length of time required by the process to complete. The HELIXsystem Process Governor is a generally available capability, able to be deployed across virtually any heterogeneous loosely coupled environment.

The HELIXsystem Process Governor “monitor and compare” methodology is delivered without introducing proprietary protocols, recompiling or re-architecting the services or impacting overall system performance.