A Workflow Engine Server (WES) for the Design of Adaptive and Scalable Workflows

Andrzej Jasinski, Yuansong Qiao, John Keeney, Enda Fallon, Ronan Flynn

a.jasinski@research.ait.ie

Introduction

The generation of a workflow is an important management system that is widely applied in many different types of business. The traditional workflow approach, applied in specialised areas such as health or environmental control, is designed to manage one particular task flow; this means that it is not flexible or adaptive The traditional workflow approach does not lend itself to automatic updates or upgrades when required; this situation can lead to the replacement of a business's management system. This research proposes a new architecture for the design of workflows that are both adaptive and scalable. It is cantered on a novel workflow engine server (WES) that can process received jobs or requests and, based on a method chosen from a number of options, generate dynamic workflows and actions to be taken. The architecture redefines traditional workflow behaviour. Where previously a workflow was created for a particular job or task, the workflow will now be created by the job or task. The proposed solution eliminates the problems of adaptability and scalability associated with traditional workflow management. The proposed architecture can be applied to the design of workflows in both large-scale (5G networks) and small-scale (home IoT) environments.

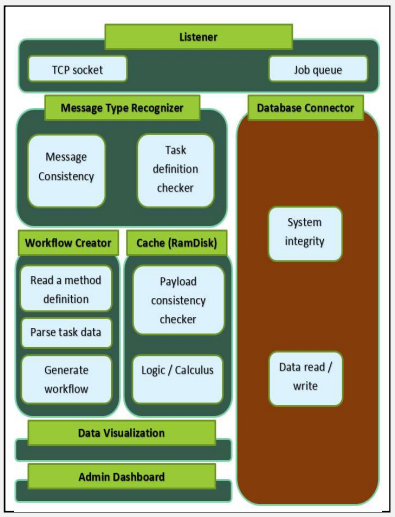
Workflow Engine Server Architecture

WES Attributes

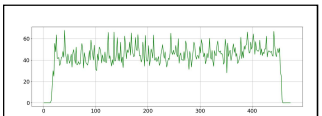
1. Multi-client behaviour
2. Connection security
3. Message Integrity
4. System Integrity
5. WES Methods

(The server uses a single or multi-method behaviour to build a workflow. The method is defined and stored in the WES database (MongoDB). The method name that the WES should use to generate a workflow must be coded in the message transmitted by the client (method descriptor) and be available on the client's request. Method types are pattern-based, template-based, ML-based, coding-based, combined and not-defined/not-known)

6. Self Adaptive Behaviour
7. Reusable Behaviour
8. Job independent and Update-ready



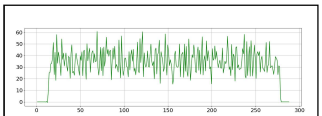
Architecture Evaluation



Multiple concurrent connections stored in the WES log file (x-axis represents the time in seconds, y-axis represents the number of simultaneous connections per second)

	PC-1	PC-2
Number of Connected Clients	50	50
Time for client synchronisation (seconds)	12	11.96
Total number of messages	12622	12070
Average number of messages (seconds)	21.87	20.14
Average response time (ms)	1097.08	1141.67
Minimum response time (ms)	26.78	60.41
Maximum response time (ms)	4034.9	4336.09
Number of connection errors	0	0

Connection measurements on client side for the two host computers PC-1 and PC-2, where each computer hosts 50 clients



WES message consistency and error detection based on log file (x-axis represents the time in seconds, y-axis represents the number of simultaneous connections per second)

	PC-1	PC-2
Number of Connected Clients	80	20
Time for client synchronisation (seconds)	12.79	11.35
Total number of messages	9425	3526
Average number of messages (seconds)	33.59	12.83
Average response time (ms)	811.12	370.26
Minimum response time (ms)	21.93	0.06
Maximum response time (ms)	3194.97	1738.88
Number of connection errors	0	3486

Connection measurements on client side for the two host computers PC-1 and PC-2 where each computer hosts 80 and 20 clients respectively, when authentication errors are introduced

Conclusion

This paper proposes a new smart approach for workflow management. The workflow engine server is a solution for both individual and large-scale customers. The performed tests showed the potential of block-oriented systems. The presented architecture is a semi-independent management system and assumes a management role automatically. The WES is able to dynamically generate a workflow based on particular stored methods. The WES can be run both in the public or private cloud, and on low-cost platforms like Raspberry-Pi. An open architecture allows flexibility for upgrading or modification without significantly impacting the entire system structure. The proposed WES architecture will be evaluated in the management of simple home systems (CCTV cameras or security alarms) and to more challenging configurations, such as 5G and the IoT.

This work was supported by the Irish Research Council under grant number EPSPG/2017/336

AIT Connect & Discover

2019