Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2018-11-29

Deposited version:
Post-print

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Coutinho, C., Lopes, L., Viana, V., Pape, D., Klasen, G., von Halem, B....Stam, A. (2018). An open environment for development of manufacturing applications on vf-OS. In Martin Zelm; Frank-Walter Jaekel; Guy Doumeingts; Martin Wollschlaeger; (Ed.), Enterprise interoperability: smart services and business impact of enterprise interoperability. (pp. 107-114). Hoboken: Wiley.

Further information on publisher's website:
10.1002/9781119564034.ch13

Publisher's copyright statement:
This is the peer reviewed version of the following article: Coutinho, C., Lopes, L., Viana, V., Pape, D., Klasen, G., von Halem, B....Stam, A. (2018). An open environment for development of manufacturing applications on vf-OS. In Martin Zelm; Frank-Walter Jaekel; Guy Doumeingts; Martin Wollschlaeger; (Ed.), Enterprise interoperability: smart services and business impact of enterprise interoperability. (pp. 107-114). Hoboken: Wiley., which has been published in final form at https://dx.doi.org/10.1002/9781119564034.ch13. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

An Open Environment for Development of Manufacturing Applications on vf-OS

Carlos Coutinho[1], Luís Lopes[2], Vítor Viana[2], Danny Pape[3], Gerrit Klasen[3], Bastian von Halem[3], Óscar Garcia[4], Ludo Stellingwerff[5] and Andries Stam[5]

[1] Caixa Mágica Software/Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR-IUL, Lisbon, Portugal
[2] Caixa Mágica Software, S.A., Lisbon, Portugal
[3] Ascora GmbH, Ganderkesee, Germany
[4] Information Catalyst, Valencia, Spain
[5] Almende BV, Rotterdam, The Netherlands
carlos.coutinho@caixamagica.pt
luis.lopes@caixamagica.pt
vitor.viana@caixamagica.pt
pape@ascora.de
klasen@ascora.de
vonhalem@ascora.de
oscar.garcia@informationcatalyst.com
ludo@almende.org
andries@almende.org

Abstract. Developing a project aimed at providing an Operating System to a manufacturing system requires having a suite of tools that can specify, develop, build and distribute the applications that will be at the heart of this manufacturing system. These applications shall be able to receive stimuli from and to provide actuation over the manufacturing assets, combining this virtualised manufacturing system with the power of the novel ICT resources, enablers and state-of-the-art techniques for development and building of systems and applications that have been used in industry, which involve WYSIWYG editors with syntax highlighting, SDKs which provide a large set of artefacts and libraries including standard connectors and extension points, but also debuggers, tools to optimise the performance and make the best use of the latest technology. Additionally, innovative methodologies for development promote strong involvement of the development community and the developers' engagement. This paper describes a novel environment that was envisioned for this purpose, involved in the scope of the H2020 European Project vf-OS.

Keywords: Manufacturing SDK, Integrated Development Environment, Dependencies, Build.

1 Introduction

The upcoming of the new industrial revolution “Industrie 4.0” [1] presents a new paradigm for the development of manufacturing services, tightly aligned with the most advanced ICT innovations. At the very heart of this development are new techniques that pro-mote the collection of data from multiple sensors and sources of input information and the ability to infer knowledge from it (Internet of Things – IoT), and virtualisation of the manufacturing assets. The new smart digital manufacturing environments will allow the development of applications that are able to interact with the virtual factory assets, receiving inputs and actuating on the real factory systems. This will allow them to benefit from the advances in the methodologies and development of software applications that traditionally were not applicable to the manufacturing domain, with multipurposed agile dashboards and front-ends that deal with accessibility and that are more responsive and provide more valuable information. New methodologies which include agile techniques, the use of popular and proven development frameworks, the support of development communities that provide manpower for testing and for new ideas, suggestions and bug corrections. Additionally, the development of applications shall be more pleasant, with a set of tools like IDEs and editors with code syntax highlighting, debuggers, drag and drop features, integration with version control, issue trackers and other mechanisms that engage and motivate the developer into the creation of applications, a set of libraries accessible through the development SDK that will provide access to innovative features such as new enablers, dependencies analysis, dashboards and especially the connection to the internet and to a whole new level of ICT assets.

The developers will be able to produce “What You See Is What You Get” (WYSIWYG) applications that use front-end elements commonly used on web-based applications, based on the latest technologies and best-practices of the market.

The H2020 project vf-OS [2] proposes to create an operating system targeted to the manufacturing business. This naturally involves the development of mechanisms that interact with the real factory assets, enablers for connecting to internet and web features, and an environment that fosters the development of applications for the manufacturing business. The vf-OS development framework is named virtual factory Open Application development Kit (OAK) [3]. It is composed by a set of libraries, assets and tools which are described in the next subsections.

1.1 vf-OAK SDK

The vf-OAK Software Development Kit (SDK) is a centralised environment for the development of applications and, generically, for the centralised access of the vf-OS assets and functionalities. The SDK itself will not have a user interface *per se*, instead, it will be accessed as a set of APIs to access the main development resources. That way, the SDK will be able to provide to the vf-OAK Studio (see section 1.2) and to other Application/Development components the resources and services that they require.

The vf-OAK SDK comprises an extensive fully-documented API framework that provides to developers the means to easily generate applications (vApps) and services.

It also includes the development of a service API which offers methods to access functionalities of all technical components in the vf-OS environment that the service developers need, such as the easy integration of data from the Cloud-based information infrastructure and the storage of service-specific data within the Cloud.

The basis for the development of the SDK is to provide a component or framework of elements that foster and improve the development of solutions and vApps for vf-OS. The idea is to be able to help a developer to create vApps with this framework such that the SDK will provide what needs to be included in the vApp, and then other OAK components will be responsible for defining behaviours, process flows and other application definitions, in a user-friendly interface, based on Front-end customised elements, that aims to promote and foster the development.

Particularly speaking, the SDK is divided into several parts considering the components that are to be connected: The SDK core is responsible for handling and exporting the communication layer between the OAK Studio or other components for actions e.g., publish/subscribe, messages, or security. The SDK for vApps has a process interpreter which can transform the results from the Process Designer tool (XML files) to JavaScript language. And the SDK for the IOToolkit accesses the real manufacturing devices, sensors and other mechanisms via the vf-IO interfaces.

1.2 vf-OAK Studio

The vf-OAK Studio provides an Integrated Development Environment (IDE) with a Graphical User Interface (GUI) for users that want to develop vApps for the vf-OS platform. It also provides all the necessary tools and means for developing a vApp, and supports the application deployment on end user devices, and its publication in multiple Marketplaces. Besides including tools for code development, with editors that include features like Business Process Management Notation (BPMN), syntax highlighting and drag-and-drop, debugging and analysis of the developed code, the vf-OAK Studio includes the interfaces for Process Designer, Front-end Composer, and provides access to the vf-Store marketplace.

The vApp creation starts by setting the necessary configuration which includes dependencies, marketplaces and permissions that are going to be used by the vApp. Then the vf-OAK Studio provides a working space composed by a Process Designer (see section 1.3) and a list of services and processes available in the vf-Store marketplace which are specified during the project configuration.

The implementation of vApps will be based on the composition of pre-built blocks, where the developer can select an available process/service from the marketplace, and drag-and-drop it as a block to the Process Designer. Each block can be configured by the user. If a process or service needed by the user is not available in the list, the user can create and develop a new customised block. The logic for the implementation of this custom block can be done using or composing other blocks or code. When finished, the custom block can be saved it locally, so it can be used on the vApp, or deployed to one or more marketplaces, so that other developers can make use of it.

The vf-OAK Studio also includes a Front-end composer (see section 1.4) which manages the user interfaces used by the vApp. From this tool, the user can use the available templates, e.g., default login page, or create new ones. This User Interface (UI) composer also supports logic implementation used by the UI, e.g., filters the received data and shows the results or gets the inputs entered by the user.

1.3 Process Designer

vf-OS will use a Process Designer for complementing the development of the vApps. The Process Designer is responsible for allowing users to model multiple manufacturing workflows for orchestrating the various assets available within a collaborative framework. The tool will be an online workspace supporting a BPMN-like model and will be usable by vApps whenever process design and orchestration is appropriate.

The Process Designer will be a visual online reactive canvas, allowing the developer to pull in existing models from a library representing the virtualised manufacturing assets and all the elements which will be necessary for the correct development of a vApp. These elements can be any of the vf-OS Assets: Generic FIWARE Enablers (Application, Services, e.g., single-sign-on (SSO), but also Specification/Methodology), Manufacturing Enablers (Applications, Services and Specification/Methodology), vf-OS Services (e.g., Transformation), vf-OS Component (e.g., Process Engine [4]), and of course, other vApps.

Each asset may support additional properties that can be defined. The defined workflow can include other sub-workflows and be saved and versioned within the Data Storage as BPMN-model definitions. This will allow a developer to design an ad-hoc workflow that will model and shape the desired vApp. Examples of vApps could be the execution of a specific business process or the integration both intra-plant and cross-company data exchange processes. These services, which will be orchestrated at runtime, will be augmented with additional information needed to define the required processes.

The vf-OS Process Designer will be a web-based BPMN tool accessible through the vf-OS OAK. It will take advantage of cutting-edge web technologies. The foundations of the Process Designer will make use of an open source BPMN modeller framework such as BPMN-IO, which is a rendering toolkit and web modeller for BPMN. It allows easy creation of BPMN2.0 diagrams using a web based modelling library which can be extended to add the functionality for vf-OS.

The toolbox inside the Process Designer will offer vf-OS manufacturing assets (Services) that are available from the vf-Store. From this repository, a process designer can then select and drag from the Toolbox onto the design canvas to begin modelling a process. It will contain all the standard BPMN modelling elements, such as Parallel or Exclusive Gateways, Processes, and many others needed to create a BPMN diagram.

1.4 Front-end

The Front-end Environment is part of the vf-OAK and builds a library of graphical elements that are optionally linked to simple functionalities. These elements are cross-platform compatible and they serve as a help for developers to quickly and safely create consistent looking vApps. The libraries from the Front-end Environment are integrated into the vf-OAK Studio, and can be used directly, by using the elements in the shape of classes. By using the Front-end Environment, the developers can choose between the following repositories:

- **UI Element Repository** contains single UI elements following the style guide of vf-OS that can be used during vApp development. Particular attention was paid to the corporate holistic layout of vf-OS for branding purposes;
- **UI Template Repository** provides compositions of UI elements from the UI Element Repository. Each template appears as a single UI element but is composed of several UI predefined elements. A lightweight example of a UI template is a login form, which is composed of a grid, textbox, password-textbox, and action buttons;
- **Behaviour Repository** contains several default behaviours for activities, which are related to UI interactions. Those behaviours are easy to integrate and applicable for e.g., forms, downloads, registration, and notifications;
- **Behaviour Template Repository** provides a composition of several behaviour templates. A single behaviour template consists of lined-up behaviours that are processed synchronously. Lined-up behaviours ease the use of handling events and support developers with additional default events;
- **Holistic Template Repository** provides holistic templates that unite UI templates and behaviour templates. Those templates follow the style guide and include logic to handle user interactions automatically.

Furthermore, in vf-OS, the Front-end Environment is used both in design and at runtime. In design time the developer can choose elements from the library and can integrate them in their vApps with help of the vf-Studio. But in runtime, demanded user interactions can call simple functionalities from public vf-OS interfaces, such as the analytics component, or the vf-SDK.

1.5 System Dashboard

The vf-OS System Dashboard is a runtime, central dashboard and task manager, aimed at monitoring, warning, configuring, and adapting system resources. As such, it provides information on the runtime behaviour of the vf-Apps, useful for both the end user and the developers of the apps.

The primary component of the dashboard is a web-based user interface, in which the status of all user accessible vf-OS components and deployed assets can be monitored and controlled.

Behind this UI, a controller service is provided which aims to collect the required information for the dashboard and which contacts the execution services for the starting and stopping of vf-Apps and their dependencies.

A notification service is also part of the dashboard, allowing the user (and/or developer) to be informed of off-nominal issues, problems or other triggers. The Notification Engine stores and manages notifications, and applies several external messaging mechanisms (e.g., email, messaging, notification icon) to warn users of such issues.

1.6 Developers Engagement Hub

To complement the actions of development, creation, debugging and integration, the modern methodologies for supporting the development advocate that as good or better than having a great development team is to have an active and engaged community of users, partners, customers and developers that contribute to the success of the projects.

Hence, the team proposed the inclusion of a centralised hub which has the purpose of gathering and promoting the contact, interaction, collaboration and discussion of interested stakeholders. These communities often help with their requirements, their suggestions and ideas, but also greatly with their own development or testing of the developed vApps. This is the largely successful philosophy around Open Source, but to be able to do this in a correct way, these communities need to be correctly biased and engaged to the developing project. The vf-OAK suite therefore includes a platform that has as sole purpose the capitalisation of the developing project, providing mechanisms that promote a clear and straightforward description of what is being developed, and its surrounding concepts, ideas, methodologies, requirements, and as much more information as possible.

The vf-OS Developers Engagement Hub will promote the dissemination of the developed code and its evolution using source code control (e.g., Git repositories), and tools which will mostly deal with dissemination and communication (collaboration), which may include wikis, templates, tutorials, multimedia, forums, mechanisms for capturing and describing new features and bugs (issue trackers), management tools, and many others which may support the development and make it simpler like business and continuous integration, connections to larger ERP and CRM tools, statistics and other tools which may both facilitate the development, its documentation and dissemination.

2 Conclusion

The purpose of the H2020 vf-OS European Project is to provide manufacturing businesses with an open Operating System that will allow them to – similarly to what happens in a regular computer – create, develop, build and load applications that span and cover the whole manufacturing operations and processes, enabling the capture of the business knowledge and promoting its analysis and potential to achieve greater value.

The authors wish to acknowledge the support of the European Commission through the funding of the H2020 vf-OS project (GA 723710) for their support, interaction and contribution in the development of the research that is presented on this paper.

References

1. Industrie 4.0 Homepage, http://www.plattform-i40.de/I40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html, last accessed 2018/01/10.
2. vf-OS Homepage, http://vf-os.eu, last accessed 2018/01/10.
3. vf-OS Consortium, D2.1 Global Architecture Definition (2017).
4. http://158.42.105.151/mediawiki/index.php/Process_Engine, last accessed 2018/01/10.