Designing Workflow Systems Using Building Blocks

The purpose of this study is to show how design and engineering of workflow system using a building block approach can enhance flexibility and enables sharing of features without restricting functionality, performance, or sustainability. According to the excerpts, the building block approach employs a middleware design that is used in performing workflows. The fundamental properties of this architectural concept are self-sufficiency, interoperability, composability, and extensibility. The self-sufficient refers to the fact that software building block is independent of specification of other modules; interoperability defines how to using it in varied systems without modifying the building block itself; composability indicates a building block that is a well-defined module with interfaces that facilitate the communication between building blocks; and lastly, the extensibility is defined as the ability for extending functionalities as well as adding new features to the building block.

The RADICAL-Cybertools are layered modules that alternative composition of them can execute the workflows. The RADICAL-Cybertools are divided into four modules: RADICAL-SAGA, RADICAL-Pilot, RADICAL-WLMS, and EnTK. The RADICAL-SAGA deals with job submission and job requirements, handling and has all features of the building block approach. RADICAL-Pilot is responsible for managing the execution of workloads, supports self-sufficiency, interoperability, extensibility, and partially composability features. RADICAL-WLMS with use of execution strategy separates the planning and management of each workload execution. The last module that satisfied all properties of the building block approach, EnTK, enables the distributed execution of applications with specific computational patterns. All of these parts have been designed to use separately in the various systems. The innovation of building block approach is its design methodology.

The reading passage provides a survey to prove the integration of RADICAL-Cybertools with other ecosystems that needs the lowest alteration. Initially, it argues about domain-specific workflow (DSW) systems that employing the EnTK module of RADICAL-Cybertools. EnTK deals with four DSW systems: ExTASY, RepEx, HTBAC, and SeisFlows. These workflow systems has benefited from the use of EnTK by its execution patterns, replica-exchange pattern, the state-of-the-art abilities that lead to the minimalize development work and complexity, and its data flow management capabilities. Furthermore, the excerpt claims that integrating Swift with RADICAL-WLMS has a myriad of advantages. This integration allows Swift workflows executed concurrently and distributed on both HPC and HTC resources as well as the time to completion of workflows improved by RADICAL-WLMS capabilities without requiring extra functionalities. Also, RADICAL-Pilot as an independent subsystem aided the FireWorks distributed workflow system. This independent integration of RADICAL-Pilot will help the FireWorks by reducing the usage of HPC resources with the ability of late binding tasks to high-performance computing resources. Finally, the author delineates the Workload Management System, PanDA, without any alternation uses RADICAL-Pilot and -SAGA to enable pilot capabilities via new RADICAL-Cybertools, that is named the Next Generation Executer (NGE). In this integration, NGE performs like a resource queue for PanDA that resolves the challenge of executing numerous amount of small jobs.

This work focuses on the architecture design of building blocks that is still in progress and is not completed. Moreover, the building blocks approach is not immature. In addition, this work does not separate the ecosystems that building block approach has better advantages for them rather than others. Finally, the paper does not analyze the wider implications for the middleware ecosystem aimed at scientific computing. One purpose of this research is to begin a discussion on how the workflow community members can better manage, collaborate, and decrease the unnecessary efforts. The building block approach is a start on the road to examination and investigation of design principles and architectural patterns for workflow systems.