Architecture and Initial Development of a Knowledge-as-a-Service Activator for Computable Knowledge Objects for Health

Allen J FLYNN\(^{a,b}\), Peter BOISVERT\(^b\), Nate GITTLEN\(^b\), Colin GROSS\(^b\), Brad IOTT\(^a\), Carl LAGOZE\(^b\), George MENG\(^b\), and Charles P FRIEDMAN\(^{a,b,c}\)

\(^a\) School of Information, \(^b\) Medical School Department of Learning Health Sciences, and, \(^c\) School of Public Health at the University of Michigan

Abstract. The Knowledge Grid (KGrid) is a research and development program toward infrastructure capable of greatly decreasing latency between the publication of new biomedical knowledge and its widespread uptake into practice. KGrid comprises digital knowledge objects, an online Library to store them, and an Activator that uses them to provide Knowledge-as-a-Service (KaaS). KGrid’s Activator enables computable biomedical knowledge, held in knowledge objects, to be rapidly deployed at Internet-scale in cloud computing environments for improved health. Here we present the Activator, its system architecture and primary functions.

Keywords. Knowledge Objects, Digital Library, Activator, Knowledge Management, Knowledge as a Service, Learning Health System, eHealth

1. Introduction

This work addresses the latency between biomedical knowledge discovery and its widespread uptake into practice [1]. We believe fully computable (i.e., computer-actionable) biomedical knowledge is needed to achieve Learning Health System (LHSs) [2, 3]. Without it, inadequate health care outcomes will persist because slow dissemination of human-readable knowledge thwarts rapid, system-wide learning [1, 4].

In 1999, Cheah and Abidi discussed the growing significance of Knowledge Management (KM) in healthcare [5]. They described a 7-part KM framework [6], including a “Share/Disseminate” process to move previously stored knowledge throughout the health care enterprise [5]. Since that time, to help share and disseminate knowledge, cloud computing architectures have emerged that enable persistent, online Knowledge-as-a-Service (KaaS) [7]. KaaS combines computable knowledge and data in a cloud environment to generate advice on-demand [8]. However, before KaaS platforms can enable scalable sharing of routinely updated computable biomedical knowledge for LHSs, they must meet healthcare’s security, reliability, data format, and other requirements [9]. KaaS platforms that meet these requirements may help overcome longstanding problems with interoperability and computable knowledge sharing [9].

1 Allen J. Flynn, Department of Learning Health Sciences of the Medical School, University of Michigan, 300 North Ingalls Building, 1161C, Ann Arbor, Michigan, USA, 48109; E-mail: ajflynn@umich.edu
The Knowledge Grid (KGrid, www.kgrid.org), a computable biomedical knowledge platform under development at the University of Michigan, includes digital knowledge objects (KOs) that follow a published formal specification [10], a digital library within which to manage and steward KOs [11], and an Activator capable of being deployed in cloud environments to provide KO-based KaaS. The Activator is so named because it puts computable biomedical knowledge stored in Knowledge Objects to work via online webservices, thereby activating it. The Activator enables realization and scaling of biomedical KaaS. This paper describes the system architecture of the KGrid Activator, which has been realized over 9 months of software development using Agile methods.

2. Research Question

The research question motivating the initial work to design and develop KaaS capabilities for the Knowledge Grid -- in the form of its Activator -- is this: What constitutes a minimum set of necessary and sufficient logical-technical components that, when integrated, result in stable, scalable, rapidly deployable, secure, traceable, activated Knowledge Objects thereby enabling KGrid to provide performant, reliable on-demand knowledge services for health using cloud computing environments?

3. Materials and Methods

To create an encompassing architecture for the KGrid Activator, one that delineates a set of necessary logical-technical components and is extensible, we analyzed various cloud computing architectures [7-9] and studied the differences between a Remote Procedure Call (RPC), Remote Method Invocation (RMI), and the network-based RESTful architectural pattern for a stateless, scalable, generic networked interface with HTTP-based semantics (i.e., GET, PUT, POST, etc.) [12]. A logical-technical component is a single part of the system architecture that can be implemented by technical means.

The KGrid Activator is an independent component built with the Spring Framework for Java [13]. Using Spring’s Web module, and Maven for Java project build automation, we have built the Activator by combining a series of interdependent Java classes and object instances to bring about Knowledge Object based end-user KaaS capabilities. We have successfully deployed the Activator on Google Cloud and Amazon Web Services.

The KGrid Library works in conjunction with the Activator. The Library’s system architecture has been previously published [11]. Fedora, version 4.x, provides KGrid’s Library with a repository for native Resource Description Framework (RDF) linked data and binary files [14]. Via the Library’s application programming interfaces (APIs), it receives requests from the Activator for specific KOs, serializes KOs in JavaScript Object Notation (JSON), and delivers them to the Activator, which enables KaaS.

Knowledge Objects (KOs), each hold a computable biomedical knowledge payload along with service-interface specifications and descriptive metadata [10]. Payloads can include computer-actionable statistical prediction models and computable guidelines. As a specific example, we have created a KO that holds executable pharmacogenomic drug dosing knowledge. When activated, this KO provides genetic information to compute dosing advice. Using such executable payloads, the Activator provides persistent online KaaS to compute predictive scores and case-specific guideline-based advice on demand.
4. Results

4.1 Functional Requirements

The two basic KaaS functions are (1) to enable use of computable knowledge downstream by disseminating knowledge resources in digital formats “as is” and (2) to apply computable knowledge, as part of the service, by accepting instance data as input, combining those data with computable knowledge, performing an execution step, and returning a computed result as new information to answer a question [7-9]. The Activator can perform both of these KaaS functions. It can either share a KO payload directly with a requestor, for computation external to KGrid, or it can process instance data, using one or more KO payloads, to generate an answer for the requestor automatically.

In addition, in our work to design and develop KGrid’s Activator, we are evolving a specific model of computable biomedical knowledge activation to extend the two most basic functions of KaaS by including several other functions we believe are minimally necessary to deploy computable biomedical knowledge safely, effectively, and efficiently to improve health. These other activation-related functions are access control, knowledge object integrity checking, activation policy setting, activation policy enforcement, and knowledge object utilization logging.

A system architecture for a KaaS Activator is shown in Figure 1.

![Figure 1. System Architecture of a KaaS Activator for Computable Knowledge Objects for Health](image)

4.2 Payload Agnostic Activator Functions

KGrid’s Activator provides access controls for all users. It also provides several functions to Knowledge Management Administrators via APIs. These general functions
are not KO payload specific, but instead are payload agnostic Activator functions. These payload agnostic Activator functions are associated with the four APIs depicted to the left of the vertical dashed orange line in Figure 1.

The Shelf API enables users to load and unload knowledge objects (KOs) into and out of the Activator, respectively. The Activator maintains a persistent KO Shelf, which is currently implemented as a simple dedicated file folder.

The Policy API enables users to set, edit, remove, and apply policies to govern aspects of loading KOs and providing KaaS using them. For example, the Policy API can enforce a rule that only KOs from certain trusted sources can be loaded into the Activator.

The Logging API provides users with a mechanism to collect knowledge service utilization data. If logging is turned on within the Activator, a new data stream is implemented and log data are stored outside of the Activator in a separate data store.

The Adapter API enables sharing and use of KO payloads. An Adapter is a plug-in that enables the Activator to meet the two basic types of KaaS service requests. The Adapter API allows users to plug one or more Adapters into an Activator instance. Four specific Adapters are shown in Figure 1. The simplest one is the Share Payload Adapter that sends the payload of a KO out to an external system upon request. Local Execute Adapters are software language-specific. When they are plugged-in, various Local Execute Adapters enable the Activator to execute Python, JavaScript, and code in other languages. In a similar way, Proxy Execute Adapters engage external services to execute code in various languages. Finally, Workflow Adapters provide the Activator with a mechanism to orchestrate execution of the multiple KO payloads in repeatable ways.

One set of functions that are not controlled by an API are the internal Checking functions of the Activator. For safety and security, the Activator has the capabilities to check the integrity of any KO, using various fixity and checksum routines. The Activator can also check that incoming instance data are a match for any KO’s service interface.

4.3 Payload Specific Activator Functions

The Activator’s KO enabled, payload-specific KaaS capabilities are made available via its Activation API, which is directly supported by a series of logical-technical components that extend to the right of the dashed orange line in Figure 1. When a standard HTTP request (i.e., GET, PUT, POST, etc.) is received by the Activator’s RESTful Activation API, that request is handled by the Knowledge Object Service Model layer. If the needed KO is already on the Activator’s Shelf, and the needed Adapter to provide the requested service is already plugged into the Activator, then the Checking functions are performed and, if all checks are passed, the request is fulfilled. If the needed KO is not on the Shelf, then the Activator can automatically initiate a search for the KO at the KGrid Library to which it is connected (Three KOs are depicted on the Shelf in Figure 1 and many more can be added). The Activation API provides a set of error responses and troubleshooting support whenever a KaaS request cannot be fulfilled.

5. Discussion

We have described the logical technical components of the KGrid Activator and their core functions. Our work is informed by efforts to make and share complex compound digital objects in other domains but it specifically advances KaaS functions for LHSs.
6. Conclusion

Much better knowledge dissemination is needed to establish LHSs [1-3]. Cloud-based KaaS has the potential to improve knowledge dissemination for health. However, most commercial KaaS platforms have not been purpose-built for health care. They do not provide needed security, knowledge integrity checking, and technical policies to safely and effectively deploy computable biomedical knowledge [5,7]. A system architecture for a KaaS Activator with the potential to support KaaS for health has been reviewed. It exhibits capabilities to make KaaS a more feasible infrastructural solution for LHSs.

References