A Bag of Wiki Stories, Attribute Oriented Programming with XDoclet, Rx ++ and GS collections.

Dr Bheemaiah, Anil Kumar, A.B Seattle W.A 98125
miyawaki@yopmail.com

Abstract:

The Wiki Story an essential documentation as javadoc is added to GS collections as a mutable data structure, added to collections like the Bag, leading to self modifiable programs with attribute oriented programming inspired by work in LISP and AIML, and the Self language of BotLibre. In this paper we integrate Open FaaS to Java 8 and RxJava, for Green Coding and the generalization to reusable components in remote functions on the edge or the cloud.

Keywords: GS collections, Eclipse collections, XDoclet, Code Generators, Open FaaS, Bayou Framework.

What:

The GS Collections has a new data structure candidate, the Wiki Story WS, WS inherits from the User Stories of the Agile process, with documentation embedded in it as comments, amenable directly to OOPS programming. A WS data structure is defined by the **this** property and reflects a uniform xml, JSON and html5 DOM structure. X Doclet is introduced as attribute oriented programming with a set of attributes , both for Beans, Streams, and Rx Operators and data structures.

How:

Attributes define Rx and Rx ++ programming with code generators from XDoclet 2 library.

Custom objects allow for the integration of Rx Stream objects, both sensor streams, event streams, kinesis streams and dynamoDB streams and many more streams. OpenFaaS is also integrated by a query based function integration as remote method or cloud based method integration with attributes, called green coding similar to the method, queryCodeGenerator()(Bheemaiah, n.d.)Bayou but extended to FaaS, services as attributes. (“How to Use Bayou – Bayou: Program Synthesis Powered by Bayesian Machine Learning” n.d.)

Why:

We have added wiki’s to the user stories provided as agile, attributes are added allowing for a query based tool for FaaS and XDoclet based code generation analogous to the neural sketch learning of Bayou. Code generation as amplification is now so fashionable that gangster like coders can also contribute really well generated code, an evolution of compiler backend code.

Applications:

Uniform high quality code, optimized to score high on Sonar, Code Generators for amplification and Green Coding.

Introduction.

XDoclet allows for the addition of code generation through modular code generation, through the generation of objects with data structures and integration of legacy code, through the open FaaS framework for uniform cloud deployment over a range of cloud providers, such as Digital Ocean, Oracle, AWS and Google.

This extends the Bayou API integration to the addition of both microservices, stateful and stateless and the integration of reusability in stateful Green Coding with functions as a service.

FaaS integrates older legacy code with custom JSON defined triggers, embedded in the Java objects defined in attributes and allows for green coding. It also provides for IaC, integrating a vast spectrum of stand-alone edge function based computing to universal UI based cloud solutions. FaaS offers both a CLI and a UI for integration.

FaaS is the natural superset of API integration for Rx and Rx++ programming, inspired by Netflix.(Netflix Technology Blog 2013).

Problem Definition.

Background.

<original-contribution>

Formal Definitions:

Definition 1.0: A **Bag** is a **Collection** which contains elements that are unordered, and may contain duplicate entries. It adds a protocol for adding, removing, and determining the number of occurrences for an item. This package contains 3 interfaces: **Bag** - contains the common API for Mutable and Immutable **Bag**.(“Website” n.d.)

Definition 1.1: We add the wiki story or WS to collections with a DOM model to it in Rx++, the DOM model is defined as an object with a this reference, and analogous to the self language of BotLibre is also equivalent to xml and to JSON.

We can thus define a Bag Of Wiki Stories. A wiki story is central to attribute oriented programming. The bag of stories, Bag <WS> is a mutable data structure allowing for the “self” modification of the DOM, a new genre of self programming software with attribute based code generators.

Definition 1.2: Attribute Oriented Programming, defines a set of meta-data attributes or tags as

comments, to define a template based approach to code generation using XDoclet 2, extended to new java bean categories as custom beans and to queryCodeGenerators, defining a Bayou ++ framework, extending the API framework to a FaaS based framework.

Factoid 1.0:

A FaaS system has a library of predefined functions with a container based deployment on cloud or on edge, including a k based deployment on gitpod.io or on a raspberryPi like IOT solution.

This includes templates, (openfaas n.d.)

Factoid 1.1:

Adding API stateful/stateless microservice, to FaaS, (“Introducing Stateless Microservices for OpenFaaS” 2018)

Factoid 1.2:

Adding Legacy code, with green coding to FaaS.(“Introducing the Template Store for OpenFaaS” 2018)

Adding to the sample functions repository.(openfaas n.d.)

(openfaas n.d.)urls for Green Coding with the queryCodeGenerator(), are added with the legacy template, from cloud storage from zip files for JSON/yaml based function descriptions and deployment on to a cluster.

</original-contribution>

Discussion.

The generator pattern is a design pattern for template based code generation as popularized with attribute oriented programming, this extends the present code generators both available in GS collections and in the IaC frameworks.

We have thus described an Rx and Rx++ framework for a generator pattern integration, to RxJava and Java 8 as a super set of the Bayou tool.

Future Work.

The AIML framework can be extended to both the CAS and the generator patterns, as an extension to its traditional NLP role. In future work we use the self language of Bot Libre model of state and pattern-templating in the

frameworks of Rx programming with the generator pattern.

`collections › api › bag › package-summary.`

References.

Bheemaiah, Anil Kumar. n.d. “Automation with the Taskoid Framework: A Case Study I.” <https://doi.org/10.35543/osf.io/w64qd>.

“How to Use Bayou – Bayou: Program Synthesis Powered by Bayesian Machine Learning.” n.d. Accessed December 14, 2019. <https://info.askbayou.com/how-to-use-bayou/>.

“Introducing Stateless Microservices for OpenFaaS.” 2018. OpenFaaS - Serverless Functions Made Simple. September 5, 2018. <https://www.openfaas.com/blog/stateless-microservices/>.

“Introducing the Template Store for OpenFaaS.” 2018. OpenFaaS - Serverless Functions Made Simple. November 28, 2018. <https://www.openfaas.com/blog/template-store/>.

Netflix Technology Blog. 2013. “Reactive Programming in the Netflix API with RxJava.” Medium. Netflix TechBlog. February 4, 2013. <https://medium.com/netflix-techblog/reactive-programming-in-the-netflix-api-with-rxjava-7811c3a1496a>.

openfaas. n.d. “Openfaas/faas.” GitHub. Accessed December 15, 2019a. <https://github.com/openfaas/faas>.

———. n.d. “Openfaas/templates.” GitHub. Accessed December 15, 2019b. <https://github.com/openfaas/templates>.

“Website.” n.d. Accessed December 14, 2019. <https://www.eclipse.org › eclipse ›>