International Journal of IT-based Public Health Management
Vol.7, No.1 (2020), pp.23-30
http://dx.doi.org/10.21742/ijiphm.2020.7.1.04

An Overview of Healthcare Interoperability through NODE-RED Workflows

Jinan Fiaidhi[1*], Sabah Mohammed[2] and Sami Mohammed[3]

[1,2]*Lakehead University, Department of Computer Science, Canada*
[3]*University of Victoria, Department of Computer Science, Canada*
[1*]*jfiaidhi@lakeheadu.ca*, [2]*mohammed@lakeheadu.ca*, [3]*smohamme@uviv.ca*

Abstract

Electronic Healthcare Record (EHR) systems from aren't designed to meet the increasingly broad and complex enterprise wide analytics as well as to interact with other systems. As a result, clinicians have a hard time leveraging the information they need to improve patient care. This article overviews the authors efforts to prototype the new generations of clinical systems by incorporating clinical workflow frameworks like Node-Red and integrating the dynamic of this integration and the changes that may occur upon these clinical workflows using an extended JDL model.

Keywords: *Healthcare Interoperability, Clinical workflows, Node-red, JDL, IFTTT*

1. Introduction

As healthcare organizations move toward integrating into a patient's EHR more clinical data from devices and hospital information systems, it is important to design and implement an interoperability strategy based on an enterprise and cross enterprises versus department-by-department approach. Virtually all health systems that have invested heavily in these EHRs already have hit limitations around accessing the data they need for a comprehensive understanding of their patient populations and running advanced analytics to meet population health management (PHM) and value-based payment (VBP) goals. Actually, EHRs grew out of an ever-emerging vast array of independent computer systems and software that stored internal hospital processes. Many of these systems were built to address a specific problem and the thought of interoperability was virtually non-existent during the development process [1]. Electronic Healthcare Record (EHR) systems from aren't designed to meet the increasingly broad and complex enterprise wide analytics as well as to interact with other systems. As a result, clinicians have a hard time leveraging the information they need to improve patient care. Analysis of clinical data from core technologies in use is often the key to identifying gaps and inefficiencies. However, the pillar of interoperability is the stream of clinical workflows. To realize this potential, the data – and the actionable insights they hold – should be available in workflow formats that can be shared effortlessly, transparently and above all securely within and between hospital systems, in the home, or on the move. In this direction many researchers [2] recommended the new visual workflows like Node-Red to prototype the future interoperable FHIR healthcare record standard. Furthermore, Node-Red would need to have a mechanism to integrate the dynamic of the change in these workflows.

Article history:
Received (September 17, 2019), Review Result (November 20, 2019), Accepted (January 9, 2020)

Print ISSN: 2205-8508, eISSN: 2207-3965 IJIPHM

Several other researchers developed more dynamic models to deal with workflow interoperability based on fusion integration. In this direction there are four models that have been taken as a lingua franca for data fusion problems including:

- JDL model [3][4]
- DDF model [5]
- Omnibus model [6]
- Perceptual Reasoning model [7]

However, none of these models deal with multiple workflows integration as well as account for human processing dynamics. This paper proposes an extension to JDL Model where it can support different types of workflow integration and user interactions as intrinsic part of the clinical system for resilience business and continuity. This extension represents another architectural layer (seventh level) to the JDL model where the layout of the system is developed as a suite of workflows of small services, each running in its own process and communicating with lightweight mechanisms. The benefits of level 7 are many, ranging from an increase in development productivity, to better business-IT alignment, agility, scalability, and technology flexibility. [Figure 1] illustrates the extended JDL model that can manage the dynamic of the fusion data from the different workflows.

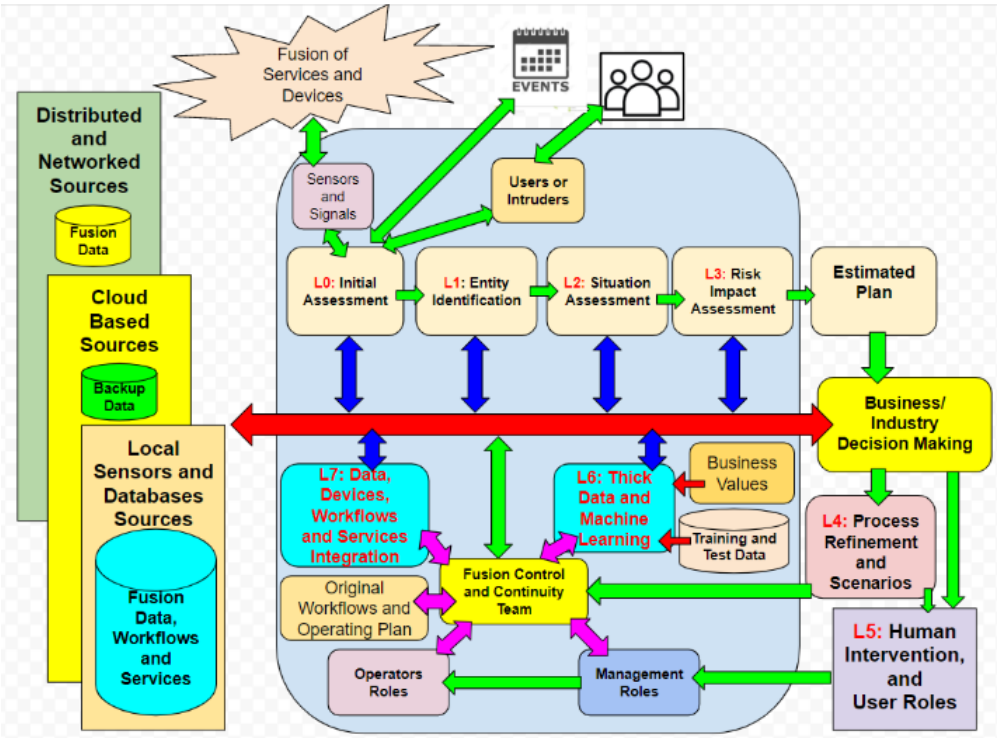

Figure 1. Extending the JDL model for dealing with clinical workflows dynamics

2. The case of using node-red for clinical workflows

In the early days, the Web was mainly used to publish information. Then, application servers were used to offer services to human customers. We now witness development of the

Services Web where the services can be accessed programmatically and application servers collaborate with each other, typically using simple protocols like NodeJS. The Services Web evolves out of the desire to perform transactions in an open and automated environment with ubiquitous services, rather than using other mechanisms such as EDI or manual processing. The Services Web environment typically exhibits the following characteristics [8]:

- Web Services are black-box components that encapsulate behavior.
- Web Services interact using NodeJS over HTTP thus providing wrappers for interoperation between technology specific components.
- Web Services can be discovered at run-time for dynamic binding.
- Several Web Services can be orchestrated to perform a series of functions in a workflow. Web Services are units of extremely low coupling; they communicate with each other with low dependency on the other party.

However, composing web services into a coherent application can be a tedious and error prone task when using traditional textual scripting languages like WSDL [9]. As an alternative, complex interactions patterns and data exchanges between different Web services can be effectively modeled using a visual language. In this paper we present the design of the ExtremeFlow composition ecosystem. ExtremeFlow uses visual composition that has been fully implemented in a development environment for Web service composition with usability features emphasizing rapid development and visual scalability. The ExtremeFlow environment provides an integrated toolkit to manage the whole lifecycle of a workflow composition in any business [Figure 2].

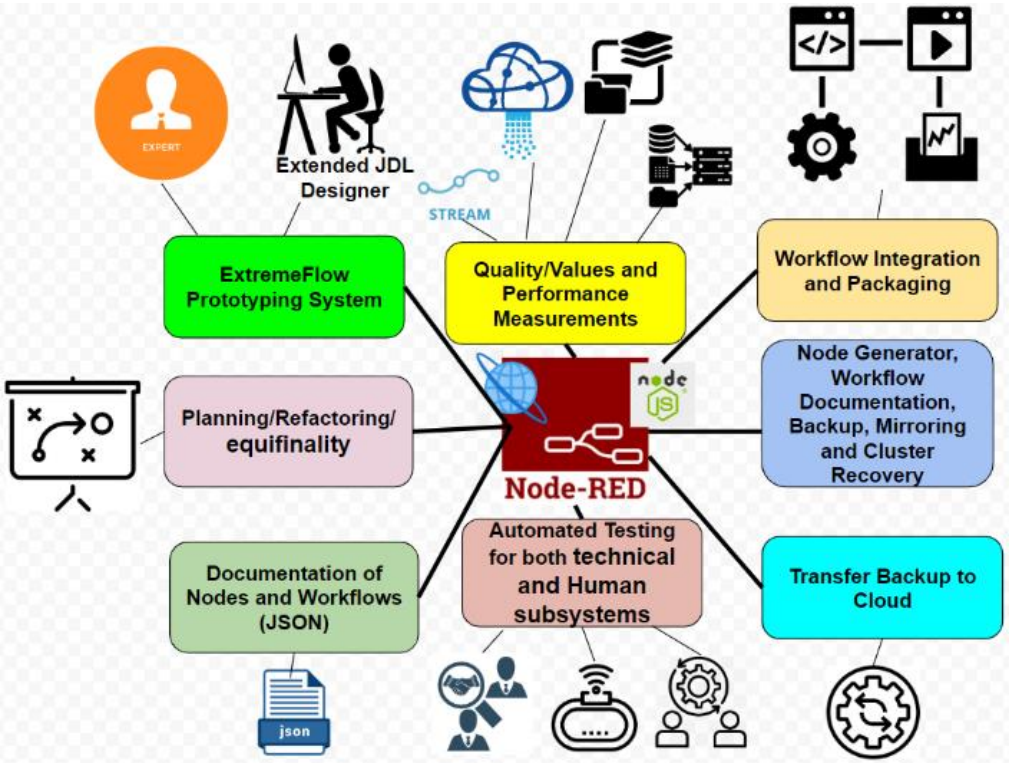

Figure 2. The extremeflow model of using node-red

In order to manage and fuse these sensor and services feeds, an architecture is required that integrates the services in a loosely coupled way to support decentralized discovery and execution. This loosely coupled nature ensures that existing services and resources can be quickly set up to be discovered and take part in query responses without having to be re-written from the ground up. For this reason, the ExtremeFlow needs to have component which is the Node-Red IFTTT Broker. This new component provides library methods to allow services to send and receive IFTTT Triggers and Actions is a web-based service to create chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest. Figure 3 illustrate the usage of the Node-Red IFTTT broker in developing and composing workflows in the healthcare paradigm.

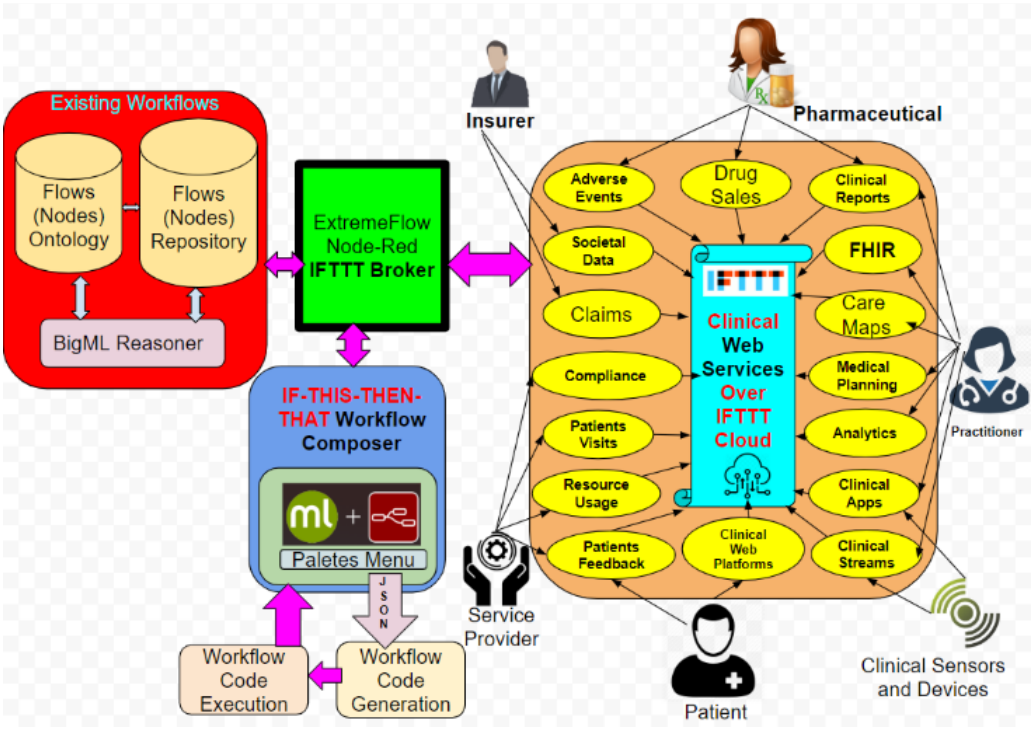

Figure 3. Implementing the extremeflow using the ifttt broker for ehealth workflow applications

The composition process using the IFTTT broker begins with the Flow Repository, where Web services can be imported as reusable components. Any user can search the repository with the help of ontology and a reasoner, select a set of existing services and drag them into the workflow composer. Then, the new workflow can be modified by adding new nodes from the palate's menu or from the web services published over the cloud. This operation is partially automatic, since the editor can bind parameters with matching names. To get an overview over the order of execution of the tasks and add additional constraints, the user may view and edit the workflow anytime. Once all of the services have been connected the new process may be compiled and uploaded to an ExtremeFlow runtime environment for execution. The user may interact with a running process or its tasks, and abort, pause, continue, and restart them at will. More than one copy of a process may be run concurrently. Once a process has completed its execution, the user may access the content of all parameters as well as measurements about the execution time of each task, until the process is explicitly

deleted from the system. The IFTTT workflow composer uses a flexible domain specific language (DSL) to solve a range of problems in the eHealth application domain. Even though a domain specific language is limited in its usage outside the domain, it can be used to write efficient code to solve problems in the target domain [10][11]. There has been a range of DSLs (e.g. Tasker, IFTTT) for heterogeneous systems like those for IoT, to enhance the programmability or simplify the process of wiring different components of an IoT system [12]. This type of DSLs allows users to create workflows with "triggers" and "actions." The "IF THIS" keyword is used to identify the triggers and "THEN THAT" is to identify the actions part. The trigger is activated by changes that occur within other web services such as FITBIT steps or an email from Gmail. Upon activation you may assign an action like pushing a reminder notification about your achieved steps [Figure 4].

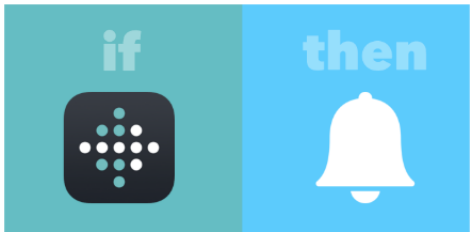

Figure 4. Pushing a reminder notification about achieved steps

And you may chain this part of the workflow with another part like if a notification from Gmail arrived then you will add it to your TODO list [Figure 5].

Figure 5. Adding to the TODO list

The automations are accomplished via Node-Red recipe with IFTTT integration - which are sort of like macros that connect multiple apps and devices to run automated tasks. The recipe employs the following concepts:

- Triggers are the "this" part of the recipe. They are the items that trigger the action. For example, from an RSS feed, you can receive a notification based on a keyword or phrase.
- Actions are the "that" part of the recipe. They are the output that results from the input of the trigger.
- Recipes are the predicates made from Triggers and Actions. For example, if you like a picture on Instagram (trigger), an ExtremeFlow app can send the photo to your Dropbox account (action).
- Ingredients are basic data available from a trigger - from the email trigger, for example; subject, body, attachment, received date, and sender's address.

3. Conclusions

We tested our ExtremeFlow composer on a Google Pixel 3 running Android 9. In every application domain, interoperability between various services comprises the integration of computation, software, networking, and physical processes. Consequently, interoperability models required an increasing support for hybrid and heterogeneous models, networking, services and time synchronization. To assist developers and users alike in designing such systems, we have developed a prototype for composing workflows based on incorporating an IFTTT Broker (IF-THIS-THEN-THAT) Domain-Specific Language (DSL) that comes with fully-automated Node-Red and IFTTT tool support. Our ExtremeFlow composer include support for: (i) interactive Node-Red model description with input validation; (ii) the computation of possible operation modes of subsystems and parts; and, (iii) checking the adherence to requirements for various design alternatives and finding the near optimal designs given these requirements. Moreover, the generated workflow models provide visualizations throughout the toolchain which help design engineers to better understand the implications of design decisions and communicate them to stakeholders. The ExtremeFlow composer has been applied to the healthcare domain. We are currently experimenting to extend the ExtremeComposer to provide smart analytic capabilities through integrating the Node-Red with the BigML services.

References

[1] Pugh Carla M., "Electronic health records, physician workflows and system change: defining a pathway to better healthcare," Annals of translational medicine, vol.7, no.1, **(2019)**

[2] David Hay, "FHIR prototyping with Node-RED (Part 1 and Part 2)," Available Online: https://fhirblog.com/2019/01/07/fhir-prototyping-with-node-red/, January 7, **(2019)**

[3] Alan Steinberg, Christopher Bowman and Franklin White, "Revisions to the JDL data fusion model," Proc. SPIE, vol.3719, pp.430-441, Sensor Fusion: Architectures, Algorithms, and Applications III, Belur V. Dasarathy; Ed. March, **(1999)**

[4] James Llinas, Christopher Bowman, Galina Rogova, and Alan Steinberg, "Revisiting the JDL data fusion model II," December 2004. Available Online: https://pdfs.semanticscholar.org/b183/008f9d63252bd4cc4438cf0bf744bc8e995d.pdf

[5] B. Dasarathy, "Decision fusion strategies in multisensor environments," IEEE Transactions on Systems, Man, and Cybernetics, vol.21, no.5, pp.1140-1154, **(1991)** DOI: 10.1109/21.120065

[6] M. Bedworth and J. Obrien, "The omnibus model: A new model of data fusion?" AES Magazine, Available Online: https://pdfs.semanticscholar.org/7595/1a01b0c603774ed99e57d9b427ce55741020.pdf, April, **(2000)**

[7] Kadar I., "Perceptual reasoning in adaptive fusion processing," SPIE, Available Online: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/4729/0000/Perceptual-reasoning-in-adaptive-fusion-processing/10.1117/12.477619.short, **(2002)**

[8] Ganesarajah Dinesh and Emil Lupu. "Workflow-based composition of web-services: a business model or a programming paradigm?" In Proceedings Sixth International Enterprise Distributed Object Computing, IEEE, pp.273-284, **(2002)** DOI: 10.1109/EDOC.2002.1137716

[9] Pautasso Cesare and Gustavo Alonso, "Visual composition of web services," In IEEE Symposium on Human Centric Computing Languages and Environments, pp.92-99, **(2003)** DOI: 10.1109/HCC.2003.1260208

[10] van den Berg Freek, Vahid Garousi, Bedir Tekinerdogan, and Boudewijn R. Haverkort, "Designing cyber-physical systems with aDSL: A domain-specific language and tool support," In 2018 13th Annual Conference on System of Systems Engineering (SoSE), pp.225-232, **(2018)** DOI: 10.1109/SYSOSE.2018.8428770

[11] Nägele Thomas and Jozef Hooman, “Rapid construction of co-simulations of cyber-physical systems in HLA using a DSL,” In 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp.247-251, **(2017)** DOI: 10.1109/SEAA.2017.29

[12] Quirk Chris, Raymond Mooney, and Michel Galley, “Language to code: Learning semantic parsers for if-this-then-that recipes,” In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol.1, pp.878-888, **(2015)**

This page is empty by intention.