The Role of FLIPP Explainers as a Tool to Assist the Visual Composition of Web Services for Enterprise Systems

Neil Adams, Simon Polovina, and Richard Hill

Faculty of Arts, Computing, Engineering & Sciences,
Sheffield Hallam University, Sheffield, United Kingdom
neil.g.adams@gmail.com,{s.polovina, r.hill}@shu.ac.uk

Abstract. This work considers process orchestration for software development on Service-Oriented Architectures (SOA). In this research the concept of FLIPP explainers are introduced to the web service community. The FLIPP explainer is a conceptual structure providing mass simplification to otherwise complex logic without the use of text or symbols. In this research the FLIPP explainer has been transformed into a workable tool to develop composite applications alongside SOAs as a direct alternative to some of the current products being offered by SAP and Oracle. Tests indicate that the tool has potential to assist in the development of applications but offers more real promise for the visualization of complex systems and processes. Also, the work highlights the fundamental issues that the FLIPP has for software development, including the initial complexity of designing the diagrams. Finally, guidelines for future enhancements and potential work are provided.

1 FLIPP Explainers and Service Composition

Part of SAP's strategy for improving the way enterprise applications are constructed requires a faster and more flexible development process [8]. With the major manufacturers now introducing Service-Oriented Architectures (SOA), tools are being provided that allow business analysts to create applications by dynamically composing web services in order to quickly produce systems.

It is only now that service-oriented architectures are a viable solution for enterprise computing that visual development has any real chance of succeeding due to the increased abstraction offered by the services as they are now semantically closer to the business than ever before[4]. SAP consider their NetWeaver suite to be the market leader in SOA systems and therefore Visual Composer (VC) to be the pioneering application for business analysts to use. This work aims to improve visual development through the use of FLIPP Explainers as a direct comparison the the SAP alternative.

FLIPP explainers are a logical method to provide massive simplification to complex systems[1]. Cox explains that the diagrams allow complex systems to be visualized "without language, symbols or formulae". He proposes that the diagrams achieve this by creating ‘scenarios’ which portray the unambiguous options to the user. Sowa [7] comments that each diagram provides the user with an acrylic and-or graph in rectangular blocks, nested together to form the system. Cox and Polovina [2] imply that by using the diagram frames, it is possible to represent ideas that are not easily conveyed through semantics alone.

The most convenient way to describe FLIPP Explainers is through example and Figure 1 shows a sample FLIPP diagram by Sowa[7]. Cox explains how to read the diagrams as “read down, don’t cross verticals”. From this example it is possible to analyse the characteristics of the FLIPP diagrams. Sowa[7] notes that the 11 boxes, marked A to K, are grouped vertically, by implicit (AND) symbols, and horizontally, by implicit (OR) symbols. Therefore, according to Sowa, figure 1b can be represented logically through the following equivalent formula:

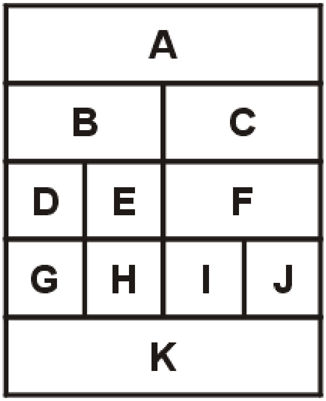


Fig. 1. A sample FLIPP diagram from Sowa[7]

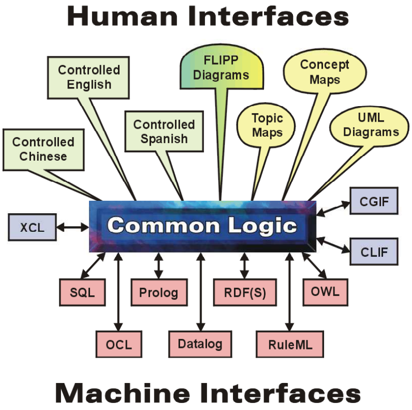
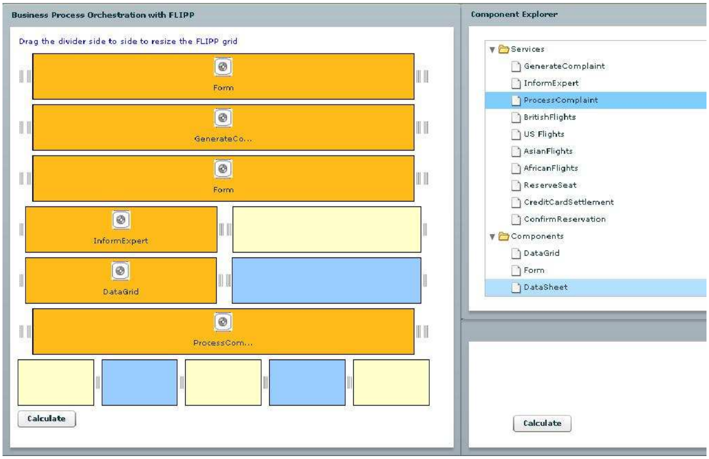


Fig. 2. Relationships of various languages to common logic, Sowa[7]

$$A \wedge ((B \wedge ((D \wedge G) \wedge (E \wedge H))) \wedge (C \wedge F \wedge (I \wedge J))) \wedge K \tag{1}$$

In English the above diagram can be read as follows: *“Start with A, if you proceed to B as opposed to C then use D followed by G or E then followed by H. If you follow A with C then use F followed by either I or J. Regardless of route, finally finish with K.”*

It is by this reasoning that it seems that web services can be modelled through these diagrams in order to improve user understanding and also to provide a logical framework for semi-automated decision making. Although the concept of FLIPP diagrams in relation to user interfaces has not been investigated yet, Sowa[7] has researched them in their original context. Figure 2 shows Sowa’s positioning of FLIPP diagrams in relation to common logic and other structures related to the semantic web. By mapping a FLIPP diagram to formats such as controlled English, processing can occur upon the logic by existing languages. What is attractive about the FLIPP explainer in relation to this work is the impact that it has upon data visualization. FLIPP diagrams immediately convey complex information to people without the need to learn symbols or notation.



Drag the divider side to side to resize the FLIPP grid

Fig. 3. Sample application modelled inside the FLIPP interface.

2 Discussion

The results of this research have been collated from a number of business and process analysts working at British Airways. In order to ascertain the results, the users were subjected to a range of case studies and asked for opinions regarding the usability and relevance of the interface for their current jobs. These results are outlined in the section below.

When comparing FLIPP explainers with SAP Visual Composer some of the claims made by Cox regarding FLIPP explainer appear to be subjective. Analyst 1 found that lines were easier to decipher than the box system whereas Analyst 2, although understanding the methodology, failed to see a real advantage at this stage. However, Analyst 3 found that reading the processes was much easier on the FLIPP interface than on VC. Therefore it appears that there are two distinct phases of using the FLIPP diagram in process orchestration; software composition, creating a FLIPP, and software modification, reading from a FLIPP.

Cox's[1] main claim, regarding FLIPP's simplicity of reading, is certainly shared by Analysts 2 and 3. Both users felt that it is simple to track processes through the FLIPP grid yet they both struggled to produce the diagrams initially. Parush et al.'s [5] work note that the use of graphic stimuli in a structured interface help to create a 'holistic' view that users can interpret at a glance. This effect was enhanced significantly amongst inexperienced users which is also evident in the FLIPP testing analysis. Analyst 3 had little knowledge of the are prior to experimentation and afterwards understood the purpose of both the diagrams and the systems perfectly.

However, Analyst 1 had plenty of knowledge of reading diagrams from UML and other experience but failed to see the benefit of the holistic view. However, when it came to developing the interfaces all the users had at least some initial problems. There appear to be two obvious explanations for this trend. Firstly it could be down to the lack of a finalised interface that caused the user frustration with simple problems such as limitations with the colours or deleting items. However it is more likely that it is harder to plan a system in one's head prior to expressing it on a FLIPP interface. The development process of a FLIPP explainer is not covered in detail in Cox's material yet it is a fundamental part of the process. Therefore with this part of the process proving challenging it may deter other users as well.

A common criticism of the interface in the results is the inability for the user to create a series of loops or to use iteration to reach a particular goal. Analyst 3's suggestion of using bolder borders, seems like a particularly interesting idea, especially when compared to the idea of having linking arrows, thus effectively detracting much of the simplification from the diagram. However even this leads to some questions regarding usability as Cox[1] champions the FLIPP diagrams to reduce the complexity of standard arduous textual descriptions. Other systems, such as SAP's Visual Composer, have the luxury of being able to simple connect an arrow back to the start of the loop which is activated when a guard condition is met. Interestingly in the SAP system it is not possible to mark the guard condition on the line thus limiting the visualization that is available.

Analyst 2 raised the point that between cells in the process there is no method for identifying the condition that is met to trigger a certain path through the system. Inside the SAP interface there is generally either a description on a connecting line or the inputs/outputs to be connected are clearly shown. However, inside the FLIPP interface at present this information isn't available which causes uncertainty. Pautusso and Alonso[6] conclude that a visual approach is the natural complement to service selection only when the correct amount of information is provided in the visualization.

In order to read the FLIPP explainers the creator has derived a top to bottom methodology as standard. However it was commented on by Analysts 2 and 3 that processes are traditionally expressed from left to right and Analyst 1 automatically began developing in the left hand corner. Analyst 2, in particular, found that the processes should be expressed this way to promote usability amongst differing sets of people. Many of the results taken from the analysts appear to represent the FLIPP interface in a bad light. However the analysts didn't report that the interface contained any fundamental problems. All the analysts agreed that the interface was easy to read from and many of the points mentioned above are purely improvements that would be necessary to use the interface for on a full scale implementation.

One major problem that has become apparent with the FLIPP explainer when used in this context is related to the connection of inputs and outputs. Each of the analysts liked the simple method of connection that the interface utilised by combining the inputs via a single screen. However the problems become apparent in the modification of the grid layout. When the grid is moved with the web services attached it can effectively disconnect any connections that have been made when the web services were placed in the grid.

3 Conclusions

The principle contribution is that the FLIPP explainer appears to have a future role in the composition of web services, assuming that some of the initial limitations can be resolved. In response to the original question, of whether it is possible to portray more information through a simpler visualization, the results indicate that it is possible to convey more information through a FLIPP based interface than through the current implementations. This is substantiated as follows:

– Although the results suggest that it is easier to read data from FLIPP diagrams, it is not clear whether it is a tangible effort or time saving approach for the user;
– Being able to compile FLIPP diagrams quickly and accurately appears to be an ability that requires some degree of skill;
– The benefit of process visualization via FLIPP diagrams also depends on the person involved to a certain extent. However it appears to be less important than when creating systems.

It is apparent that the FLIPP explainers original claim by Cox[1] that users prefer complex logic to be portrayed without ‘language, symbols or formula’ is partially true when used for developmental purposes. Users have a desire to feel empowered by the application and not restricted or frustrated with functional limitations. Therefore the challenge for implementing a successful FLIPP interface is to understand the balance between usability and simplicity. Building a FLIPP interface true to Cox’s guidelines [1] would provide a logically superior product which would excel at demonstrating systems to people unaware of the method. However, as noted by Halipern and Tarr[3], too much simplification restricts the purpose of the interface, particularly for experienced users, thus negating any advantages that might have been achieved.

One of the limitations of the FLIPP interface is that it still fails to provide an adequate method in assisting users to select the correct services for their applications. In its current form it serves simply as a tool in the armory of the Visual Composer user, and would best be used as an optional construct dependent upon the developer’s preferences.

However, the tool still has a much greater potential when assisting information visualization. Ultimately, this is the strength of the FLIPP explainer, and by implementing a system where the grid is used to assist in modifying previous composite applications, or for identifying useful components from other processes, its potential will be maximised.

4 Future Work

This work represents the first stages of a topic that could potentially lead in many disparate directions. Firstly, the immediate future would be to create an interface with sufficient functionality that it could be used inside an SOA environment to provide genuine 'like for like' tests. This, followed by more detailed analysis of user behaviour in the environment would provide a detailed insight into the real benefits of FLIPP diagrams for a variety of users. Another possible use of the FLIPP diagram is to combine the interface with other Common Logic tools. This would enable users to quickly and easily search for a process that would meet their exact needs whilst communicating the relevant output via the FLIPP interface. Sowa[7] describes a link between FLIPP Explainers and Common Logic, thus suggesting that this approach has potential. Since it is possible to represent the FLIPP data inside common logic notation, it would therefore be feasible to expose the composed service for placement within other conceptual structures. Future work involving FLIPP Explainers is to be focused around improving service composition and visualizing the processes for future modifications.

References

1. Cox, D. (2005) Explanation by Pattern Means Massive Simplification (an E-book), [online] Last accessed 20th June 2007 at http://www.flipp-explainers.org/
2. Cox, D. and Polovina, S (2007) Helping System Users to Be Smarter by Representing Logic in Transaction Frame Diagrams , In Proceedings: of15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, July 22-27, 2007.
3. Hailpern and Tarr (2006) Model-driven development: The good, the bad, and the ugly, In IBM Systems Journal, Vol 45, No 3.
4. Norton, D (2007) SOA is a Catalyst for Model-Driven Development, Gartner Research
5. Parush, A., Hod, A. and Shtub, A. (2006) Impact of visualization type and contextual factors on performance with enterprise resource planning systems, Computers & Industrial Engineering, 52, 133-142.
6. Pautasso, C. and Alonso, G. (2003) Visual Composition of Web Services, In: Proceedings of the 2003 IEEE Symposia on Human Centric Computing Languages and Environments, Auckland, New Zealand, October 2003.
7. Sowa, J. F. (2007) FLIPP Diagrams, [online] Last Accessed 20th November 2007 at: http://www.jfsowa.com/logic/flipp.htm
8. Woods, D & Mattern, T. (2006) Enterprise SOA: Designing IT for business innovation, O'Reilly.