

## Chapter 8

# Semantic Matching, Propagation and Transformation for Composition in Component- Based Systems

**Eric Bouillet**  
*IBM Research, USA*

**Mark Feblowitz**  
*IBM Research, USA*

**Zhen Liu**  
*IBM Research, USA*

**Anand Ranganathan**  
*IBM Research, USA*

**Anton Riabov**  
*IBM Research, USA*

### ABSTRACT

*Composition of software applications from component parts in response to high-level goals is a long-standing and highly challenging goal. We target the problem of composition in flow-based information processing systems and demonstrate how application composition and component development can be facilitated by the use of semantically described application metadata. The semantic metadata describe both the data flowing through each application and the processing performed in the associated application code. In this paper, we explore some of the key features of the semantic model, including the matching of outputs to input requirements, and the transformation and the propagation of semantic properties by components.*

Information analysts and decision-makers in many application domains face a daunting task. In order to quickly arrive at and continually update the business intelligence that informs their advice or decisions, they must collect data from many di-

verse sources, adapt and integrate that data and apply a variety of analytic models. For such users, the need for timely acquisition and analysis is not adequately addressed by either fixed, monolithic applications nor by queries against fixed-schema data stores. As new sources are discovered and new or evolving analyses are needed, new ap-

DOI: 10.4018/978-1-4666-0261-8.ch008

plication requirements trigger new development cycles, resulting in delayed decisions and requiring analysts to resort to manual acquisition and analysis in the interim.

Such circumstances have given rise to an important area in end-user driven computing that Cherbakov, et al, describe as *situational applications* (SAs) (Cherbakov, Bravery, & Pandya, 2007): "...situational applications describes applications built to address a particular situation, problem, or challenge. The development life cycle of these types of applications is quite different from the traditional IT-developed, SOA-based solution. SAs are usually built by casual programmers using short, iterative development life cycles that often are measured in days or weeks, not months or years."

How is it that the end user can assemble such applications? Would it be reasonable to expect that, to arrive at a desired outcome, they could identify the components needed (from a library of hundreds or thousands), know how to assemble them, and know how to prevent the seemingly sensible assembly of components into incorrect compositions? While developers can manually craft code modules and compose them into applications, how can the less knowledgeable application end users do so, without acquiring deep knowledge of the code, interface compatibility, the rules of assembly, etc.? Mashup tools such as Yahoo Pipes (Yahoo Pipes, 2007) and IBM Damia (Altinel et al, 2007) simplify the logistics of assembling situational applications, but they are limited to a relatively small and inextensible set of components. Such tools might encode interface compatibility rules, but they provide little visibility into the sometimes subtle semantics of the components' capabilities. We present a method of separately specifying *processing goals* for situational applications and the *functional capabilities* of the components to be used in addressing the goals. These specifications are used by composition tools that either

automatically provide a near-optimal solution or support the more knowledgeable user in the formulation of such a solution. We present a means of expressing processing goals using semantic expressions (graph expressions) built using terminology from the end users' problem domains. Descriptions of components' capabilities are similarly expressed. This approach is supported by SAWMILL, our semantics-based automatic flow composition middleware, as first introduced in (Bouillet et al., 2007).

The general problem of application composition is extremely challenging. To address a more tractable problem space we limit our focus to those applications described as *flow-based* information processing applications (Morrison, 1971). Flow-based applications are component assemblies arranged in a directed acyclic graph (*flow composition*) of black-box components connected by data flow links. Systems like Yahoo Pipes and IBM DAMIA support the creation of data mashups as flow-based applications. Stream processing systems like System S (Jain et al., 2006) also support flow-based applications that continuously process streaming data. Acyclic workflows in service-oriented systems can also be viewed as flow-based applications, where different services exchange messages either directly or through a coordinator service (e.g., a BPEL workflow).

In this article we describe composition of flow-based applications, describe the formal model underlying the semantic descriptions and describe how these descriptions support the composition of flow-based applications. In particular, we focus on key features of the semantic model that are particularly useful in facilitating composition. These include the matching of outputs of components to input requirements of other components, modeling the propagation of certain semantic properties by components from the inputs to the outputs, and modeling the transformation of other semantic properties by the components.

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/semantic-matching-propagation-transformation-composition/65126](http://www.igi-global.com/chapter/semantic-matching-propagation-transformation-composition/65126)

## Related Content

---

### Locally Recurrent Neural Networks and Their Applications

Todor D. Ganchev (2010). *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques* (pp. 195-222).

[www.irma-international.org/chapter/locally-recurrent-neural-networks-their/36986](http://www.irma-international.org/chapter/locally-recurrent-neural-networks-their/36986)

### Knowledge Discovery of Hospital Medical Technology Based on Partial Ordered Structure Diagrams

Dingju Zhu, Jianbin Tan, Guangbo Luo, Haoxiang Gu, Zhanhao Ye, Renfeng Deng, Keyi He, KaiLeung Yungand Andrew W. H. Ip (2023). *International Journal of Software Science and Computational Intelligence* (pp. 1-16).

[www.irma-international.org/article/knowledge-discovery-of-hospital-medical-technology-based-on-partial-ordered-structure-diagrams/320499](http://www.irma-international.org/article/knowledge-discovery-of-hospital-medical-technology-based-on-partial-ordered-structure-diagrams/320499)

### A Learning-based Neural Network Model for the Detection and Classification of SQL Injection Attacks

Naghmeh Moradpoor Sheykhkanloo (2020). *Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications* (pp. 450-475).

[www.irma-international.org/chapter/a-learning-based-neural-network-model-for-the-detection-and-classification-of-sql-injection-attacks/237886](http://www.irma-international.org/chapter/a-learning-based-neural-network-model-for-the-detection-and-classification-of-sql-injection-attacks/237886)

### Multi-Sensor Motion Fusion Using Deep Neural Network Learning

Xinyao Sun, Anup Basuand Irene Cheng (2020). *Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications* (pp. 568-586).

[www.irma-international.org/chapter/multi-sensor-motion-fusion-using-deep-neural-network-learning/237893](http://www.irma-international.org/chapter/multi-sensor-motion-fusion-using-deep-neural-network-learning/237893)

### Helicopter Motion Control Using a General Regression Neural Network

T. G.B. Amaral, M. M. Crisostomoand V. Fernao Pires (2003). *Computational Intelligence in Control* (pp. 41-68).

[www.irma-international.org/chapter/helicopter-motion-control-using-general/6845](http://www.irma-international.org/chapter/helicopter-motion-control-using-general/6845)