

# A Database-Supported Workbench for Information Fusion: InFuse\*

Oliver Dunemann, Ingolf Geist, Roland Jesse, Kai-Uwe Sattler, and Andreas  
Stephanik

Department of Computer Science, University of Magdeburg,  
P.O. Box 4120, D-39016 Magdeburg, Germany  
fusion@cs.uni-magdeburg.de

## 1 Introduction

*Information Fusion* is the process of integration and interpretation of heterogeneous data in order to gain new information of higher quality [3]. A successful support for this task requires a tight coupling of different integration and analysis tools: accessing heterogeneous data sources, their integration, preparation and transformation, analysis of syntactic, semantic and temporal structures as well as their visualisation. The INFUSE framework relies on database techniques with the goal to meet these requirements.

The proposed demonstration studies *Comparative Genomics* as one Information Fusion scenario. Gene information from different, heterogeneous sequence databases are used by several operators to analyse the function of unknown gene sequences within the demonstration.

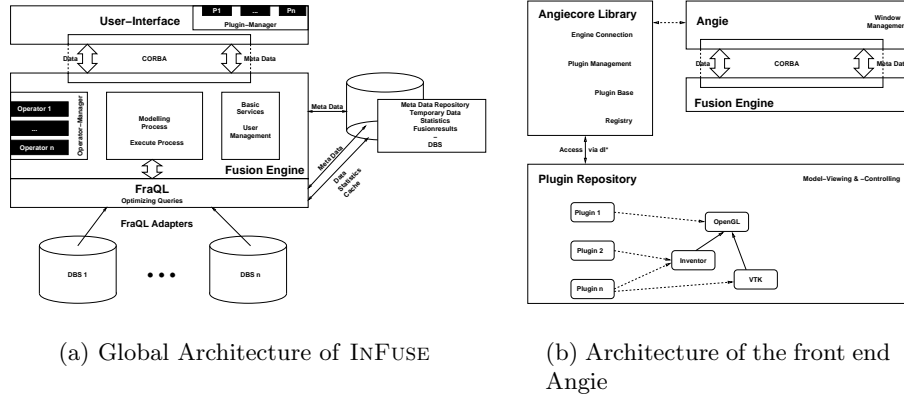
## 2 The InFuse System

The INFUSE system is designed as a database centered and component based middleware system to efficiently support the Information Fusion requirements described above. As shown in Figure 1(a) the global architecture of INFUSE consists of three main tiers: the *fusion engine* for process and meta data management, the FRAQL *query processor* for data management of heterogeneous sources and a *front end* for interactive graphical data analysis and exploration.

The *fusion engine* represents the central part of the system and is responsible for different tasks. Because a fusion process consists of several dependent steps, the fusion engine manages the definition and persistence of processes and controls their execution. Process definitions as well as the states of running processes are stored in a meta data repository. Special information gathering operators, which can be thought of as a kind of stored procedures, like data mining or machine learning algorithms, are implemented in the workbench. Besides these main features, the *fusion engine* provides additional basic services and a CORBA based API to connect to different front ends.

---

\* This work is supported by DFG (German Research Council) grant FOR 345.



(a) Global Architecture of INFUSE

(b) Architecture of the front end Angie

**Fig. 1.** The Architecture of the workbench

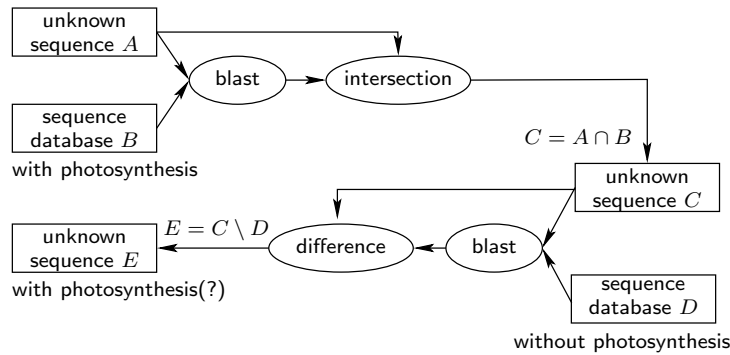
Supporting the data analysis techniques on heterogeneous data sources, the *fusion engine* relies on the features of the query processor FRAQL. A global object-relational view on heterogeneous data is offered by the query processor's adapter architecture. Several extensions in the query language provide integration and data preparation mechanisms [4].

A set of plugins is used by the front end *Angie* to support different means of visual information representation. As part of the system's visualisation network these form individual pipelines. In a visualisation pipeline, data of not necessarily geometric structure is mapped onto available display variables. These variables include but are not limited to colour, shape, and transparency. The set of mapping functions spans from glyph construction and volume rendering via splatting to the use of motion as an individual display dimension [2] for the purpose of representing temporal data behaviour.

### 3 The Scenario to be demonstrated

*Comparative Genomics* is a technique that compares gene sequences from different organisms or cells in order to determine or isolate specific domains of sequences (subsequences) using known information about other sequences. The analysis results of these comparisons are used to derive information about functions of the analyzed sequences. The process of Comparative Genomics is executed in an iterative and interactive manner.

Figure 2 illustrates a scenario of Comparative Genomics, where a set of unknown sequences *A* is analyzed for sequences coding photosynthesis. The sources *B* and *D* have to be compared with the input sequences using the *blast* operator [1]. The operators *Intersection* and *Difference* determine a set of common or excluded subsequences.



**Fig. 2.** Example scenario

In the proposed demonstration following points are intended to be shown in order to support Comparative Genomics:

**Transformation and Preparation:** Several data sources needed by gene analysis have to be accessed and prepared for different operators. Thus, the schemas and the data have to be transformed into the required structure.

**Database Techniques:** Several operators are implemented in a database-centric manner. These operators utilise the proposed query language support for data preparation and analysis.

**Interaction support:** The front end *Angie* provides techniques for guiding users through selection of analysis and visualisation methods. Thereby, the system ensures control of method combination as well as fast reply to query results. For this purpose, access to supported meta data management is provided.

**Visualisation:** The mapping of not inherent geometric data onto glyphs is shown as well as the appliance of motion for representing temporal behaviour.

## References

1. S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.
2. Roland Jesse and Thomas Strothotte. Motion Enhanced Visualization in Support of Information Fusion. In Hamid R. Arabnia, editor, *Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2001)*, pages 492–497. CSREA Press, June 2001.
3. K. Sattler and G. Saake. Supporting Information Fusion with Federated Database Technologies. In S. Conrad, W. Hasselbring, and G. Saake, editors, *Proc. 2nd Int. Workshop on Engineering Federated Information Systems, EFIS'99, Kühlungsborn, Germany, May 5–7, 1999*, pages 179–184. infix-Verlag, Sankt Augustin, 1999.
4. K.-U. Sattler, S. Conrad, and G. Saake. Adding Conflict Resolution Features to a Query Language for Database Federations. *Australian Journal of Information Systems*, 8(1):116–125, 2000.