Pegasus Architecture and Design Principles

Ming-Chien Shan
Hewlett-Packard Laboratories
shan@hplabs.hp.com

1 Introduction

To increase productivity, most enterprise operations demand an integrated information environment to facilitate the development of applications accessing multiple data sources effectively and efficiently. Typical applications include telephone service handling, computer integrated manufacturing processes, financial decision making, automated warehouses, and travel reservations.

Our goal is to provide an open and integrated information environment to support the database and service interoperability for such diverse applications in the 1990s. Based on the object-oriented technology, our prototype integrates not only database management systems (DB2, Oracle, Informix, Allbase, IMS, and OpenODB) but also multimedia data and legacy applications.

The talk will describe the overall architecture of Pegasus and the design principles of its major components.

2 The Pegasus Data Model and Language

Pegasus provides an object-oriented model as the framework for uniform interoperation of multiple data sources with different information management systems.

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0422...$1.50

The Pegasus model, which is based on the HP OpenODB's functional object model, contains three basic constructs: *types, functions and objects.*

Data definition and data manipulation in Pegasus are performed with the language HOSQL, which is an extension of OSQL, the DDL/DML of OpenODB. It is a functional and object-oriented language, which provides declarative statements to manipulate multiple, heterogeneous databases.

3 Pegasus Architecture

There are four major functional layers in the Pegasus information environment as depicted in Figure 1. Pegasus aims at providing the middleware service between individual information systems and end-users/applications. Therefore, Pegasus focuses on the devleopment of the middle two layers.

The Basic Integrated Information Service layer provdies such application neutral services as schema mapping and integration, distributed query processing, business operation flow management, directory management, and netwrok and external information systems access.

The Domain Specific Information Service layer deals with special needs for different application domains. This includes the information mining and query formulation capabilities for mobil computing devices application, and effective presentation and specific business modeling capabilities for interactive aplications, and data caching and versioning capabilities for transaction applications.

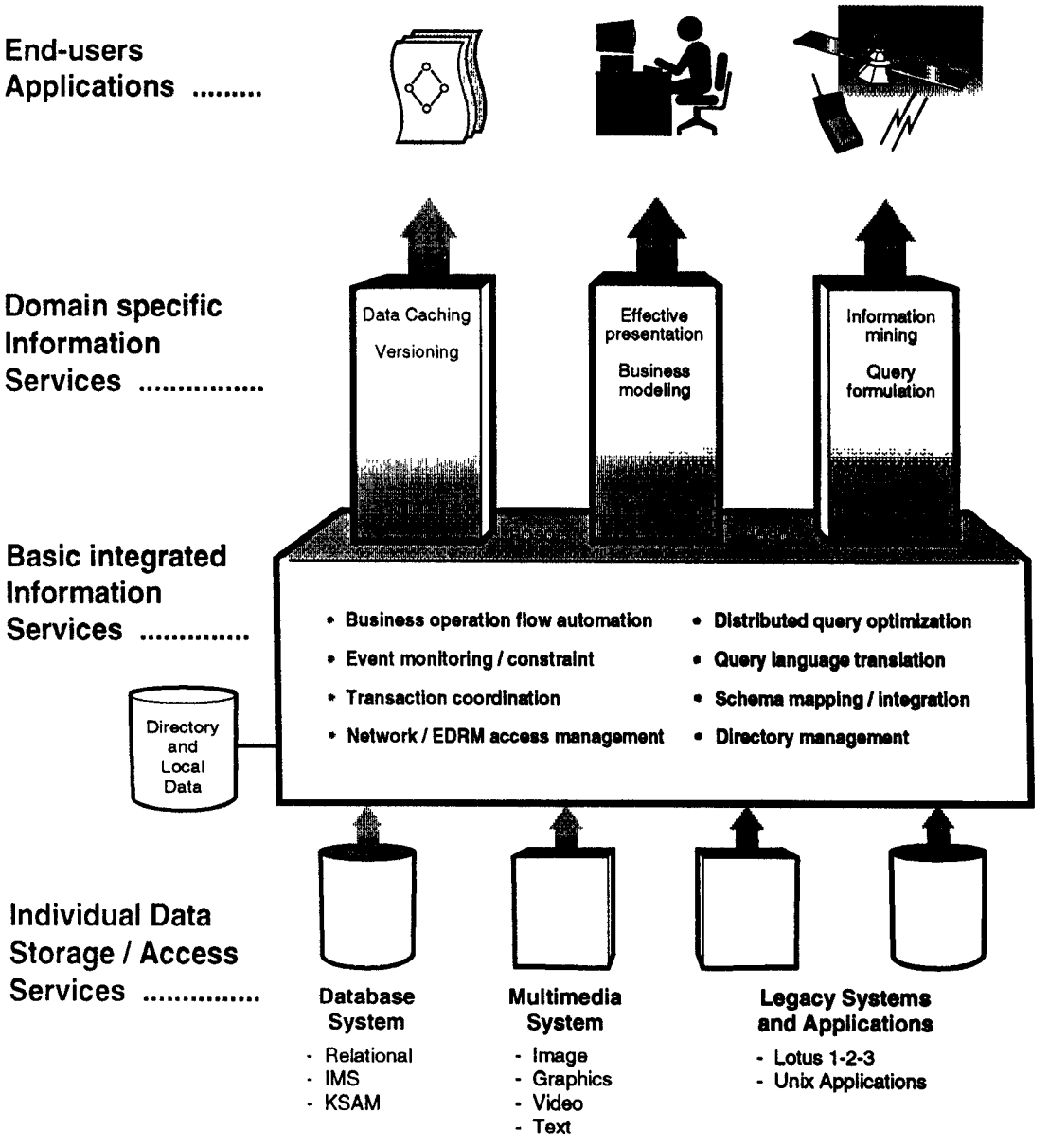

Figure 1: Pegasus Functional Architecture

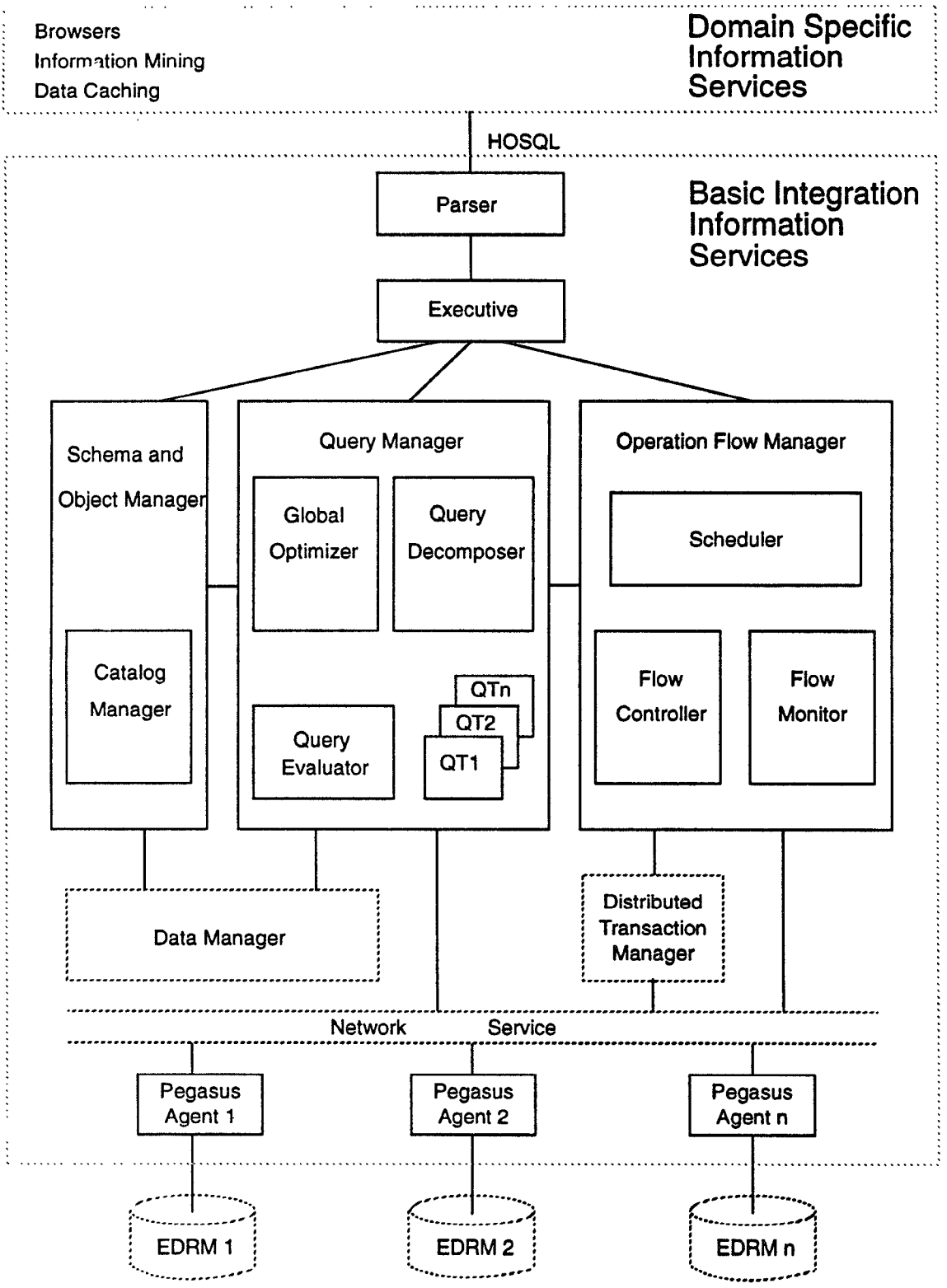

Figure 2: Pegasus Component Architecture

Figure 2 illustrates the major components of the Pegasus system. A request is submitted to Pegasus in the form of an HOSQL statement. After parsing, the Executive module routes it to the appropriate managers: the Schema and Object Manager, the Query Manager, and the Operation Flow Manager.

The Schema and Object Manager implements data definition operations and catalog management, including schema importation and integration. The Query Manager generates efficient execution plans and coordinates the execution of the plans. It consists of the Global Optimizer, Query Decomposer, Query Evaluator, and a number of Query Translators, one for each kind of External Data Resource Manager (EDRM). The Operation Flow Manager schedules and controls the flow of processes supporting complex work. It also coordidates the interaction with a distributed transaction manager, e.g., Encina, to support conventional transactions.

Meta-data and local user data are kept in a local Data Manager, which can be any standard relational system, e.g., HP/Allbase and Informix, or even a low level storage manager like the RSS component of System R.

The interaction with various EDRMs is implemented via a number of Pegasus Agents. A Pegasus Agent is a process that runs in the same machine as an EDRM. Its role is to perform the needed functions on behalf of Pegasus at the external site. It provides the services of EDRM invocation, network communication, data routing and buffering, data format exchage, and flow management related tasks.

Currently, we are focused on the development of a single Pegasus server. We plan to extend Pegasus itself into a distributed system. We are also investigating the relationship and integration of the OMG/CORBA technology and Pegasus technology to support the needs of information/service access in many diverse distributed computing enviroments.

4 Design Principles

In this section, we highlight the design principles of Pegasus major function areas:

- Information unification/integration.

  We separate the schema importation from schema integration so that (i) we can focus on one issue at a time, and (ii) we can support different user needs without overcharging them.

- Query processing.

  A single unified architecture/mechanism is developed to support access to diverse information systems, including multimedia and applications. Based on the engineering approach, a synthetic database and a set of queries are designed to calibrate foreign information systems for cost estimation. The Pegasus internal data structures are carefully designed to support extensibility.

- Operation flow management.

  We recognize that a different level of control/management of tasks is needed on top of traditional transaction management to support most enterprise applications. We focus on business operation flow automation. A cost-based operation dispatching approach is explored.

- Database tools.

  We focus on the ease-of-use rather than complete functionality. The ability to generate default schema mapping and integration as well as accepting user's hints for complicated importation and integration is emphasized.