Chapter 5
Runtime Integration Capability for Distributed Model Driven Applications

Jon Davis
Curtin University, Australia

ABSTRACT
Geographically distributed organizations face unique challenges to effectively implement shared information services across the enterprise. Traditional solutions require options such as establishing large centralized application and database servers, which simplifies some data integration issues but involves higher associated centralization risks with potential scalability limitations, or establishing multiple decentralized application servers optionally arranged in hierarchical hubs, requiring significant customization and data migration functions to be developed, reducing the level of risk but incurring additional expenditure on data integration and transfer. Our ongoing development of a distributed temporal metadata framework for Enterprise Information Systems (EIS) applications seeks to overcome these issues with the application logic model supporting the capability for direct integration with similar distributed application instances to readily provide: data replication, transfer, and transformations; centralized authorization and distribution of core identity data; sharing and deployment of modified logic model elements; and workflow integration between application instances.

INTRODUCTION
Large and geographically distributed organizations need to resolve unique challenges to effectively implement Enterprise Information System (EIS) scale shared information services across their entire organization. Multiple architecture options exist with great variability on the ultimate scalability, expense, functionality, and risk of the final solution.

A common approach is to establish very large centralized application and database servers with appropriate wide area network links to all remote business units to centrally service all users across the organization from a single virtual instance of the corporate applications. This option may simplify some data integration issues by centralizing the availability of all corporate data but requires additional costs to ensure adequate network accessibility is maintained. There are higher associated
centralization risks with this option due to the need for high network availability and failover with the potential for scalability concerns of the underlying applications to ensure they can service an expanding organization.

A potentially less risky option is to establish multiple de-centralized application and database servers to service individual or clusters of business units. Depending on the geographical user base and optionally arranged in hierarchical hubs serviced with appropriate wide area network links, each application instance provides services to all local regional users. This option typically requires significant customization and data migration functions to be developed in order to transfer the required transactions data or transaction summaries between the separate application and database instance tiers that simulate a suitable logical or functional corporate organizational hierarchy, to satisfy the information requirements of both corporate management and of each business unit. The overall level of operational risk is reduced from the previous centralized option but at the cost of incurring additional expenditure on architecture environment, and customizing data integration and transfer functions to implement the information transfers.

A common additional problem to both scenarios is that it is rare that the out-of-the-box functionality of an EIS application will satisfy all the functional requirements of an organization’s business units, requiring additional customization and maintenance expense. The cost and availability of customizations can be variable but often extensive for either architecture option.

While the technologies for application development, layer separation and deployment, data transfer and network accessibility continue to change dramatically bringing continually improving functionality and access capability, these scenarios for core application (de)centralization options and their associated risks and costs have not been fundamentally altered in terms of their basic complexity or requirements. Emerging cloud based technologies can alter the technical landscape in terms of scalability and cost of ownership but are currently primarily leveraging platform and hardware efficiencies rather than solving any underlying data and application integration issues, although multi-tenanted SaaS options are evolving as maturing candidates.

Model Driven Architecture (MDA) options or more specifically what we term as Meta-Data based Enterprise Information Systems (MDEIS), as a variant of MDA, can directly provide the capability to substantially simplify the hierarchical or indeed any organizational topology of structured information, data and application logic integration requirements for widespread geographically de-centralized organizations. We rely on the Distributed Components (DC) within the MDEIS application model and a supporting framework to seamlessly provide advanced integration services such as: data replication, transfer and transformations; centralized authorization and distribution of core identity data; sharing and deployment of modified logic model elements; and workflow integration between application instances.

How do we define MDEIS applications? Firstly, we consider the class of EIS applications that we summarize as visual and interactive applications that prompt for the entry of appropriate transaction data and user events from the application users, use rules based workflow sequences and actions and utilize database transactions in a (relational) database environment to complete the actions (Davis, 2004). They are typically structurally repetitive and tend to be a technically simpler subset of possible software applications. They generally consist of EIS and Enterprise Resource Planning (ERP) style applications such as; logistics, human resource, payroll, project costing, accounting, customer relationship management and other general database applications. The collective application design requirements are stored and available in a suitable meta-model structure and supported by an execution framework that will allow the EIS application models to be executed automatically.