Applied Enterprise Architecture: Part 1 — Application Implementation. Organizations should apply the EWTA to the application development effort. This will increase enterprise application and infrastructure asset leverage, lower TCO, and provide quick response to business process changes.

META Trend: Increasingly, an essential design point for an EWTA will be logical consistency and synchronization among various component architectures, and a consistent set of "architecture patterns" will emerge. By 2000, Global 2000 companies will explicitly recognize that optimizing the overall EWTA — thereby reducing integration complexity — is more important than optimizing the performance of the component architectures.

META Trend: By 2001, most Global 2000 companies will use a "software factory" model to implement new application systems (though this will not mature until 2003), requiring developers to move from a "craftsman" approach to a culture of assembly and reuse. These applications will be component-based, message-enabled, and event-driven, using an n-tier design that will leverage enterprises' capacity-on-demand capability.

Many of our clients have a documented enterprise-wide technical architecture (EWTA) strategy and are trying to determine how to apply the strategy to the tactical activities involved in implementing business applications. Enterprise architecture teams must get involved in the pre/post-implementation design review processes of application projects. In the early stages of architecture maturity, this will involve more education than review, but eventually, architecture assurance steps will be adopted as part of the design review process. Architects will begin to handle more consultative duties as they work with application project teams to fit the application designs to the architecture strategy and to identify new requirements that the architecture must accommodate.

META Group projects that fewer than 15% of Global 2000 companies currently have formal architecture assurance steps documented and implemented within formal design review processes. Due to Y2K demands, we see this figure only doubling by the end of 2001, but by 2003, 50% of the Global 2000 will have formal architecture assurance in place (see EAS Delta 16, 28 Jul 1998). The goal of applying the EWTA to application team efforts is twofold. First, the EWTA presents application teams, as well as other groups performing IT-related initiatives, with a set of principles from which to take guidance in making design decisions that will be consistent with other decisions being made within the IT organization. Second, the EWTA must influence the mindset and approach of application developers and integrators to stress the importance of reuse and facilitation of change.

Conceptual Architecture. The first META Trend above describes the need for a conceptual architecture that provides logical consistency and synchronization among various component architectures. The goal of the conceptual architecture is to define a set of guiding principles to be applied to IT decision making so that those decisions made individually are consistent with other decisions. For example, decisions made by the application and database teams regarding the partitioning of applications and data have a profound impact on the number, types, and sizes of servers needed to accommodate the partitioning. In addition, there is an impact on the network, middleware, and systems management areas in pursuing a highly partitioned application and data strategy. If the network, platform, middleware, and systems management decisions are not made with an understanding of the partitioning strategy, the result will be an infrastructure that does not support highly partitioned databases and applications. The conceptual architecture, which is created before the component architectures are completed,

must be understood and adhered to by the rest of the IT organization. This META Trend also mentions the emergence of "architectural patterns." META Group's Adaptive Infrastructure Strategies service has introduced seven common infrastructure patterns (see Figure 1 in Addendum). The infrastructure patterns are based on the business demand that is being satisfied. Business patterns must be identified and matched with the appropriate infrastructure pattern. The architecture team must engage application teams to identify where a pattern match may occur and help the application team design its implementation within the matching infrastructure pattern (see Figure 2 in Addendum).

Reusable Components. The second META Trend above refers to another best practice of adaptive enterprise architectures — reusable component-based development. This best practice has the greatest benefits to a development organization, though it is dependent on other best practices such as message-based interfaces and an adaptive infrastructure that provides capacity on demand (see META Practice, Volume 1, Number 3, May 1997). To realize the benefits of reusable components, the scope of analysis must be broader than a specific application domain. Enterprise architects' scope is naturally broader, so they must facilitate the process of identifying reuse opportunities. The first step in recognizing these benefits is to change the way applications are developed to make them more conducive to reusing previously coded and tested function components. Enterprise architects must facilitate the process of establishing consistent standards for defining message formats and event boundaries, and linkages between and within application domains. This process needs input from several communities of interest within an IT organization, including application, database, and infrastructure professionals. This requires architects to interact with the major application project teams to identify business process overlaps, determine potential reuse opportunities, and promote standard message formats for multiple applications to use.

Architecture Assurance Steps for Applications. The architecture team must implement the following to ensure architecture is applied to application development and integration efforts:

- Become knowledgeable of the application projects underway and planned. META Group recommends engaging the enterprise program management office for this task (see EAS Delta 17, 28 Jul 1998).
- Educate application teams on the content of the EWTA and implications to their work. This should be done initially via presentations and face-to-face meetings, giving application professionals the opportunity to ask questions.
- Identify the points in the formal design review processes in which to interject technical architecture reviews. For most application projects, this will include a high-level review when the project is chartered and another review when the project's technical direction has been finalized by the project team. The goal of the first review is to ensure the initial technical direction is compatible with the EWTA and to identify potential reuse opportunities. The goal of the second review is to identify issues or deviations proposed by the application project team. The application project team is responsible for business justification of the deviation.
- Determine whether or not the EWTA needs to be extended to satisfy an application project's proposed changes or to isolate the change as an exception.
- Provide ongoing consultation to application projects that have issues or exceptions being raised with the EWTA.

Bottom Line: To apply adaptive enterprise architecture to application implementations, formal architecture assurance steps must be crafted into an organization's application design review process. This interaction must ensure compliance with the principles and standards of the architecture, but also increase the identification of reuse opportunities across the enterprise.

Figure 1 — Seven Common Infrastructure Patterns	
Pattern Name	**Pattern Characteristics**
Host	Host or nondistributed applications typically operate almost entirely within a single system, perhaps across multiple address spaces, but work in these cases is synchronized by operating system, not network, services. Typically, host applications run under IBM's OS/390, but large numbers of Unix users and some NT users have been forced into deploying those technologies to support this pattern.
Two-Tier Client/Server	Two-tier C/S applications typically locate most application elements on a "fat client," using the database server as a powerful file system. This pattern is most appropriate for application environments that require maximum use of powerful client-side tools.
N-Tier Client/Server	The n-tier pattern offers the greatest distribution of processing function in the transactional environment. Typically, application function is deployed on multiple individual servers, while database function is allocated according to the degree of data synchronization required across supported application modules (database and server networks typically are LAN-based); clients tend to be "thin." Behavior, performance, and cost are all very sensitive to proper placement of data and proper use of synchronizing middleware.
Hub and Spoke	The hub-and-spoke pattern is designed to minimize the costs of complex data movement. It features a centralized "hub" connected to multiple "spokes," each filling local subject-oriented marts where analysis or Web data can be cached. This pattern reduces the number of interfaces that must be supported from [N(sources) * M(targets)] to the much more manageable [N(sources) + M(targets)]. Its application is limited to reading transactions, for the most part.
Enterprise Office	Enterprise office applications are fat-client applications that reside almost entirely on the client device. Typically, enterprise office applications require the largest client-device footprint. The objective is to reduce network I/O.
Remote Access	Remote access provides for autonomous, disconnected processing of enterprise applications involving minimal data synchronization. Typically, remote access applications do not allow edits, only adds. In situations where editing is necessary, data partitioning is a critical design requirement.
Electronic Commerce	Ownership of processing resources (e.g., desktop, network, even server) can be minimal. Massive scaling issues are frequent, particularly in consumer markets.
	Source: META Group

Figure 2 — Infrastructure Pattern Matching

Infrastructure pattern matching is a method used by architects and infrastructure development teams to match business needs with a common infrastructure pattern or combinations of patterns previously identified and defined. An infrastructure pattern defines the network, server, middleware, operating system, and database management system offerings provided by the appropriate standard interface services. These interface services provide a standard method of interacting with the infrastructure elements deployed throughout an enterprise to promote and provide reuse of those elements.

Source: META Group