alphaServices – an Experiment in Developing Enterprise Applications

Colin G Harrison and Edith H Stern, IBM T J Watson Research Center, Yorktown Heights, NY colinh@us.ibm.com and estern@us.ibm.com

The geographic placement of application function is determined by our attempts to optimize the costs of factors such as computation, storage, communication bandwidth, management and development. Forty years ago this meant that application integration was achieved at the register level and since that time a certain fraction of the increases in performance of the technical factors has been spent in progressively "looser" methods of integration and applications have become increasingly geographically dispersed. For some time storage costs have had the fastest rate of descent; this makes approaches such as replication and caching attractive. It now appears that communication costs are about to drop rapidly, at least in North America, while development and management costs are steady or rising. This suggests an opportunity to experiment with novel approaches to application architecture and development, which focus on minimizing development and management costs, rather than computation or communication costs.

We are beginning an experiment in which we hope to achieve software reuse and reduced management costs by delivering application functionality as a network service. We call this experiment alphaServices. Examples of alphaServices could be ZIP code verification, language translation, DTD translation, content watermarking, optimization, text and data mining services. We can readily identify hundreds of application functions that can be delivered by this mechanism. The approach is particularly relevant to application functionality that can be modeled by a transaction: that is, a bounded set of messages between the application and the functional component which does not require an extended session (and is probably stateless). Large computations which can be done "off-line" (meaning that the application calling the function does not require an immediate response) are also candidates; these include large-scale optimization problems and similar commercial computations that we describe generally as Deep Computing.

We propose a new class of service providers – Transaction Service Providers – who make available chunks of application function through the Internet via open, standard interfaces (ftp, http, XML, SSL . . .), rather than by incorporating the functionality directly into the application itself. Transaction Service Providers are similar – possibly identical – to Application Service Providers. They host application functionality that is delivered as a network service to licensed users with certain Service Level Agreements. The difference here is that Application

Service Providers expect to provide relatively small numbers of broad, horizontal applications such as Enterprise Resource Planning or accounting. Whereas a Transaction Service Provider will provide very large numbers of chunks of functionality, which are integrated into a remote platform to create some overall application.

There are several advantages to such an approach:

- ♦ This approach enables the functionality to be developed or physically integrated into the application to be reduced. Application functionality that is used only rarely does not need to be included in the application. The application is simplified down to the specific, non-generic functions required to implement the business model. The compute resource required to execute the application is reduced. The smaller or simplified compute platform reduces the Total Cost of Ownership in terms of hardware and management costs.
- ♦ This approach enables software developers with expertise in a specific piece of application function to find a new channel to market by having their work hosted by one or more at Transaction Service Providers. For the software developer, this channel to market is both cheaper and faster than traditional product marketing. New functionality and version upgrades can be deployed across the customer base in days - at most - rather than the months of a typical product/application development cycle. This could create an entirely new industry for niche developers, who can deliver a valuable but narrow function that would be hard to sell alone.
- ♦ This approach creates a new set of open API standards, which are independent of processor and operating system. If there is sufficient functionality in a transaction process based solely on the exchange of XML-documents, then these standards may also be independent of programming language. One or more industry standards organizations could standardize such API’s and offer templates for implementation. This approach may dovetail with other work to establish directories as the basis for finding and using network services.

Many organizations – enterprises, industrial research and development labs, universities - have large numbers of such chunks of functionality, whose value is greatly under realized because of the effort required to turn a useful hack into a successful product. The approach proposed here would enable enterprise applications to “script” together these functional units across the Internet as easily as we pipe together Unix programs.

IBM has for several years provided a popular Web site called alphaWorks www.alphaworks.ibm.com where IBM-developed software is posted for free download. We propose to operate an experimental service platform where application functionality based on code created within IBM Research is made freely available on the Internet – hence alphaServices. There will be no implied

Service Level Agreement! We are beginning by analyzing commercial application platforms to determine what application functions are susceptible to this approach. We will also harvest our own crop of useful hacks that fit this model. We will then develop pilot enterprise applications in collaboration with selected customers as part of our “First-of-a-Kind” program of research in the marketplace to determine whether this approach to application development and operation is feasible and worthwhile.