



Developing and Deploying an Integrated Geoenabled SOA Business Solution: A Case Study

Copyright © 2007 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100 USA.

The information contained in this document is subject to change without notice.

ESRI, the ESRI globe logo, ArcGIS, ArcCatalog, ArcInfo, ArcSDE, ArcIMS, ADF, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein may be trademarks or registered trademarks of their respective trademark owners.

Developing and Deploying an Integrated Geoenabled SOA Business Solution: A Case Study

An ESRI White Paper

Contents	Page
Introduction.....	1
Functional Scenario	1
Business Environment	1
Existing Capabilities	1
Desired Capabilities	2
Solution Description	3
Solution Overview	3
Solution Design.....	3
Service Definition and Discovery	3
County Registry Search for Service Discovery	4
Designing and Developing New Portfolio Services	6
Designing Services for Reuse	6
Designing the Geocode Service	6
Designing and Developing Composite Services.....	9
Designing and Developing Job Finder Web Client	13
Summary	15
Benefits Realized through Service Orientation.....	16

Developing and Deploying an Integrated Geoenabled SOA Business Solution: A Case Study

Introduction

Most organizations today have application portfolio needs that span multiple departments, geographies, business units, and heterogeneous hardware and software platforms. A flexible and responsive IT architecture is needed to meet the business goals of such organizations. In response, many organizations are adapting a service-oriented architecture (SOA) style of enterprise architecture in their technology infrastructure. The appeal of an SOA approach is in its promise as an IT enabler of flexible, responsive business capabilities.

In this case study, multiple disparate information systems and heritage data sources were integrated to deliver a flexible business solution using a service-oriented approach. Specifically, standards-based SOA-enabling technologies and design principles were used to integrate geographic information system (GIS) spatial analysis and visualization services, along with other business information systems, in a Web-based, service-oriented business application environment.

Functional Scenario

Business Environment

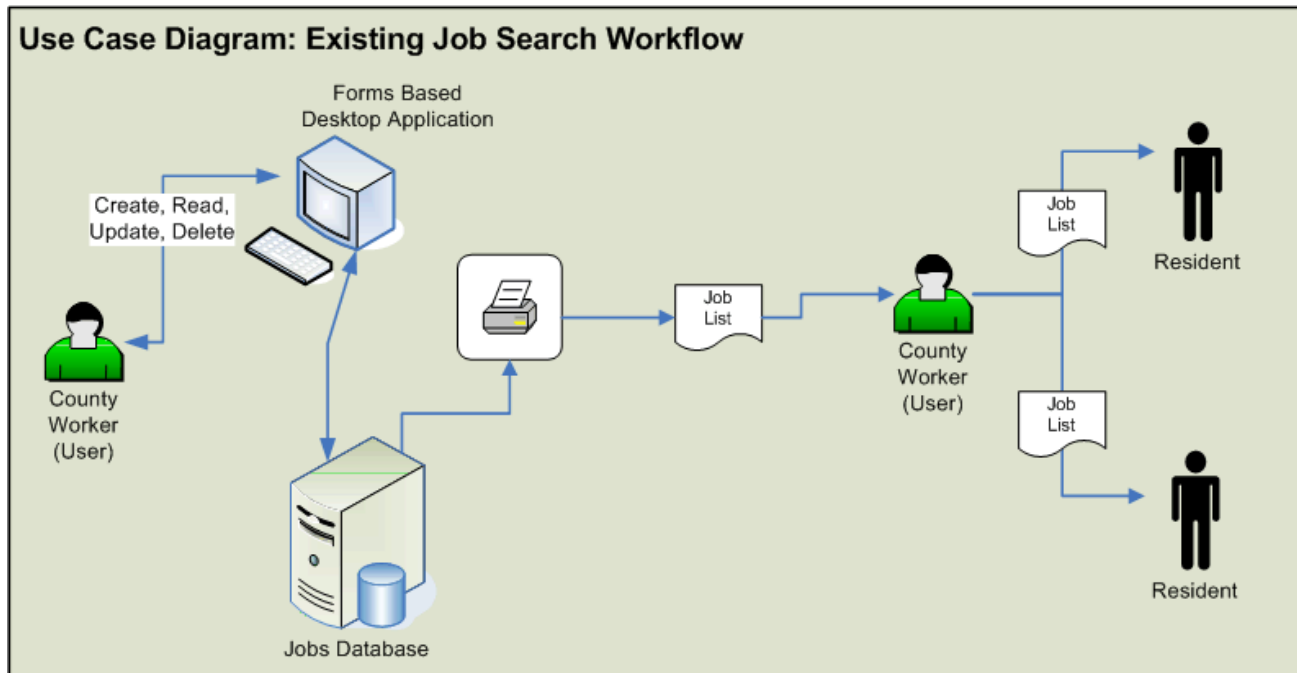
The business environment for this solution is a county government. Within the county, the Human Services Department is responsible for providing a number of resident-facing services as part of its mission. One such service is providing assistance in locating potential job opportunities for residents searching for employment. The county cooperates with local businesses for job placement services based on applicant interests, experience, and skills.

Existing Capabilities

Workers in the Human Services Department currently interact with a heritage, or legacy, client/server database system to store and retrieve information about available jobs within the county. As depicted in the use case diagram (figure 1), county workers interact with the county jobs database through a desktop forms-based client application. They were satisfied with the desktop application as a means to enter and manage job information.

The output from the legacy database system was a simple text list of available jobs with associated job information. Workers, however, found this output format unintuitive, hard to use, and not supportive of their efforts in locating the best potential job opportunity for a given candidate. Workers indicated that they often spent more time than they felt should be necessary when working with the jobs list because its utility was so limited.

Figure 1
Existing Job Search Workflow



Desired Capabilities

Workers expressed a desire to use a common Web browser to create job listings for residents. In addition to the information from the jobs database, workers desired the ability to create a map display of the returned job locations as well as the location of the job seeker's home address. This capability would improve the system by providing a visualization of the job location in relationship to the job seeker's residence, enabling location to factor into the employment search and decision.

During interviews to gather system requirements, users were very enthusiastic about adding location information to the solution and continued defining location-based requirements. They wanted the system to generate transportation routes between the job seeker's residence and selected job opportunities. They also wanted travel routing algorithms to consider whether or not the job candidate would drive a personal vehicle to and from the job location or if he or she would take public transportation.

Finally, users expressed a need to include the location of day care facilities as another factor in the identification of available jobs. When assisting residents in locating employment opportunities, county workers often were told that job seekers wanted child care in close proximity to the job location. The Human Services Department management team then defined a business rule stating that for those job candidates who want day care facilities located close by their work site the county would provide job leads that have a licensed day care facility within two miles of the work location.

Solution Description

Solution Overview

With desired capabilities in hand, the system requirements were framed within a service-oriented design, development, and implementation strategy. The county chief information officer (CIO) established architecture guidance and policy that defined SOA as the architectural approach. Recognizing the advantages of the SOA approach, the Human Services Department set out to instantiate the new Job Finder application in a manner that aligned with the tenets and principles of SOA. Thus, the new Job Finder service-oriented business application was designed, developed, and deployed in accordance with the CIO's architectural policy and guidance.

Solution Design

Based on the requirements-gathering process and an examination of the existing and desired workflow elements of the planned Job Finder business process, the service-oriented solutions team identified four main tasks for this effort as follows:

- Service definition and discovery
- Designing and developing new portfolio services
- Designing and developing composite services
- Designing and developing Job Finder Web client

The remainder of this case study illustrates the approach and methods used to design, develop, and deploy the Job Finder service-oriented business application using ArcGIS® Server and IBM® WebSphere® Process Server and leveraging heritage business information systems.

Service Definition and Discovery

The project team created an initial list of business functions to deliver the desired capabilities and workflow. Business process storyboards were developed and presented to county workers and department managers to ensure all project stakeholders shared a common vision of the solution. With a high-level definition of the business and technical requirements, along with a shared vision of the solution by all stakeholders, the design team initiated a review of the county centralized service registry to discover published business services available across the organization.

The ability to discover and subsequently reuse applicable services across multiple business systems and processes is one of the underpinnings of SOA. The creation and use of the county government service registry is a key component of the CIO's SOA governance program. Participating business, technology, and other service partners publish and describe the business services they offer in a centralized registry so that services are effectively managed, discovered, and used. A central registry framework promotes service manageability, discovery, and reusability through a standards-based publishing and discovery model. The ability to publish, discover, and reuse existing services reduces the time required to locate, understand, and implement available capabilities and is a key advantage to employing a service-oriented approach. Reusability also reduces development time, development-to-deployment cycles, and software maintenance efforts.

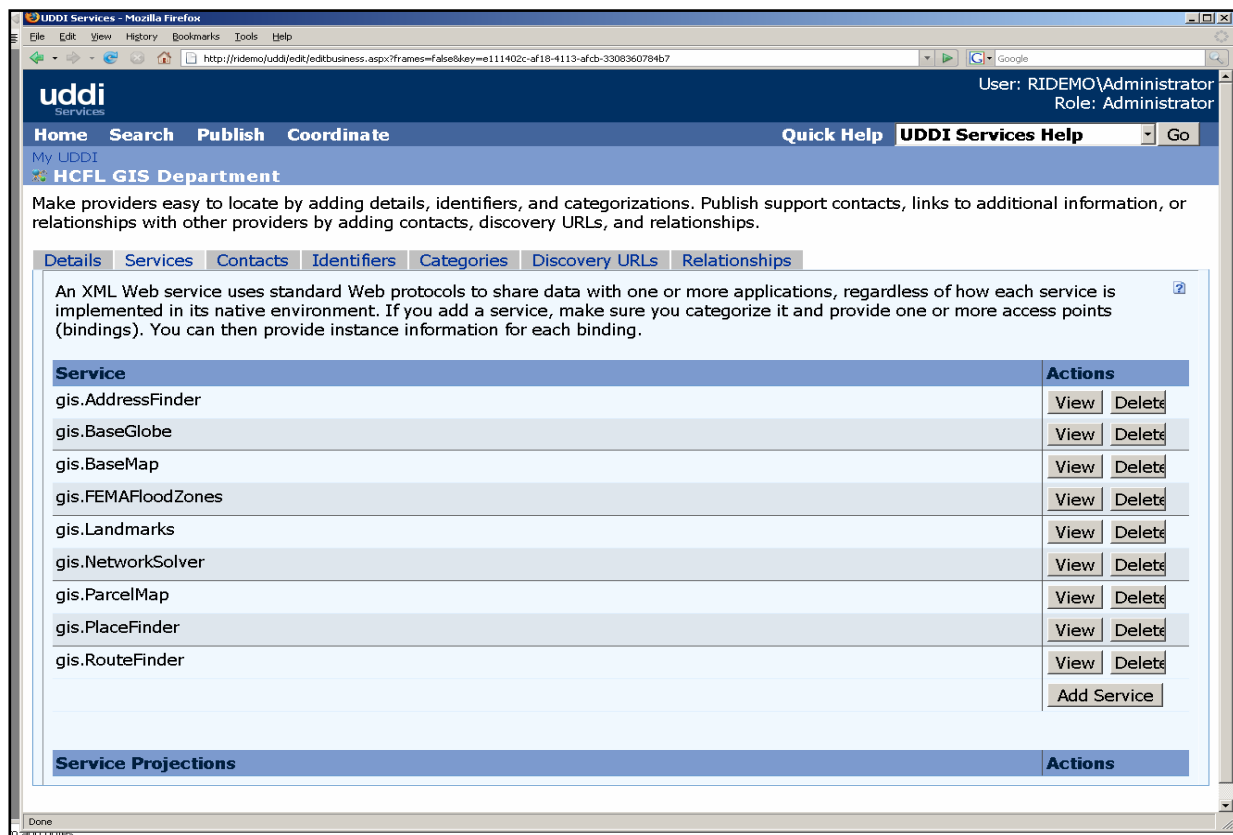
The county's service registry employs Web Services Description Language (WSDL) to publish, describe, and discover services within a Universal Description, Discovery, and Integration (UDDI) online registry. Although SOA business systems can be built without the use of a centralized registry, such an approach undermines the value of SOA. A

service registry, and governance policies that mandate its use by both providers and consumers, increases service use, reuse, and SOA value.

County Registry Search for Service Discovery

Using the county UDDI registry, the development team discovered a set of common GIS and business services applicable to the needs of the Job Finder application. Figure 2 provides a view of the county's UDDI service registry. Depicted in this view are GIS services published and maintained by the county GIS Department. These services are discoverable in the registry because they are published and are being used by other business applications within the county enterprise application portfolio.

Figure 2
UDDI Online Service Registry



In the Job Finder business scenario, the development team can reuse the published services discovered in the centralized UDDI service registry and shown in table 1.

Table 1
Published Reusable Business Services

Service Name	Owner	Description
gis.BaseMap	GIS Department	This service provides a county basemap consisting of features and objects from the county's enterprise GIS system. The county's ArcGIS Server publishes this service to multiple business applications.
gis.NetworkSolver	GIS Department	<p>In the Job Finder solution, this service provides the basemap imagery and feature information.</p> <p>This service is a service area algorithm that uses county street centerline information from the enterprise GIS system and calculates optimum route information between two or more points. The service considers travel restrictions, obstacles, public transportation nodes and routes, and other travel time/distance factors. The county's ArcGIS Server publishes this service to multiple business applications.</p>
crm.CitizenLocator	Human Services Department	<p>In the Job Finder solution, this service provides travel distance, travel time, and detailed vehicle or public transportation route information between the job seeker's residence and the job location and includes a travel stop at a day care facility if required by the job seeker.</p> <p>This service provides the registered residential address based on the county's tax system. The service accepts a resident's county identification number as input and returns the residential street address on file.</p>
crm.JobsList	Human Services Department	<p>In the Job Finder solution, this service receives the resident's unique tax identification number and returns a well-formed address record for the person's place of residence.</p> <p>This service provides a listing of available jobs within the county. The listings include all relevant job-related information. Jobs are defined by category. These categories provide an integer value that is passed to the service to filter the jobs based on a given job category.</p> <p>The county jobs database is a legacy system that is not natively Web services enabled. A WebSphere ODBC adapter is used to expose the results of a job search as a standards-based Web service.</p> <p>In the Job Finder solution, this service provides the information pertaining to available jobs including job location information.</p>

Service Name	Owner	Description
crm.BusinessFinder	Human Services Department	<p>This service provides the registered address of licensed businesses in the county. The service accepts a number of identifying attributes, such as business name, business type, and business address, to create a list of businesses and their addresses based on the information provided in the service invocation message.</p> <p>In the Job Finder solution, this service provides a list of registered day care providers in the county based on a business type database query.</p>

Designing and Developing New Portfolio Services

After discovering services in the registry that were deemed reusable for the Job Finder solution, the design team identified one additional business function needed in the Job Finder solution: the need for a service that performs a geocode operation on the residential address information returned from the crm.CitizenLocator service. A description of this new service is provided in table 2.

Table 2
Description of New GIS Business Service

Service Name	Owner	Description
gis.Geocode	GIS Department	<p>This common service is used to send an address to the server and return a geographic location—such as a point representing a house.</p> <p>This geocoding service can be used to match a single address, an array of addresses, or a list of addresses contained in a file.</p> <p>In the Job Finder solution, this service transforms resident address information into geographic x- and y-values for mapping.</p>

Designing Services for Reuse

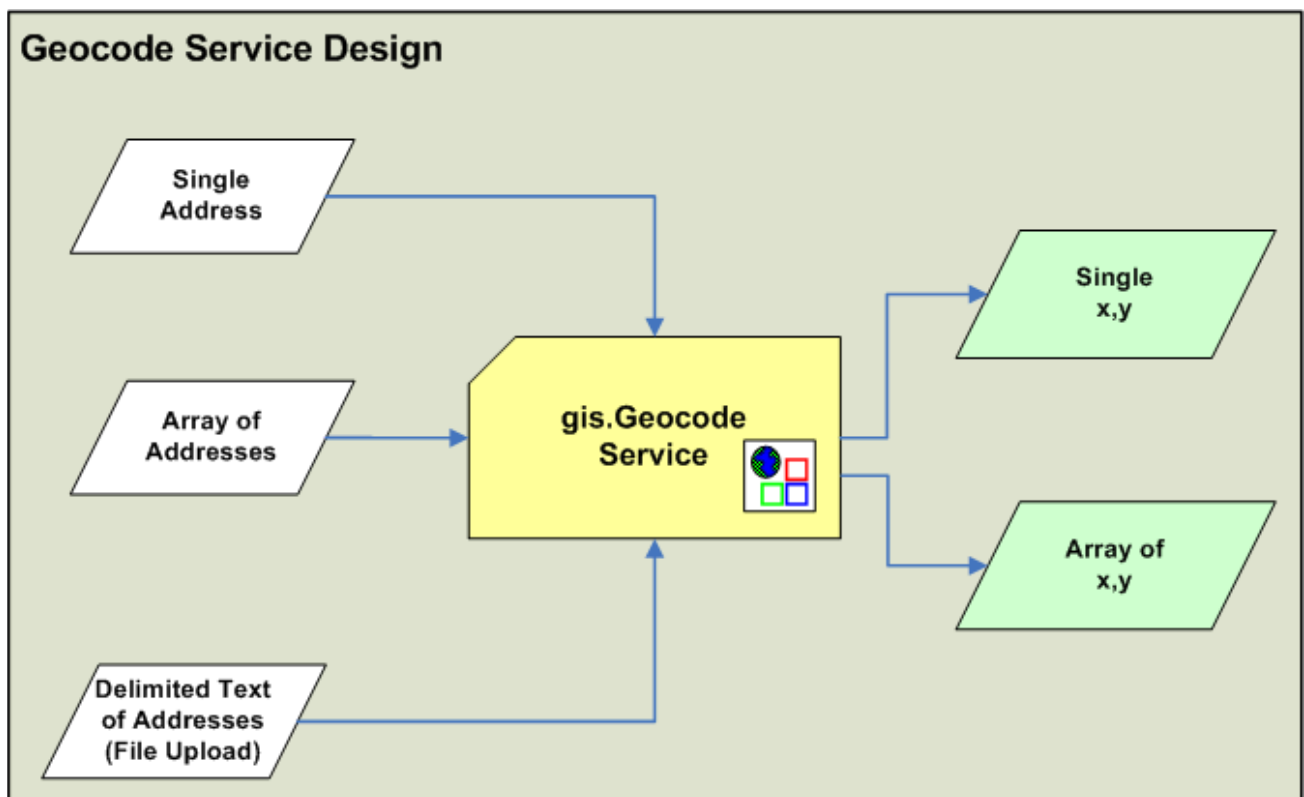
Enterprise technology services should be flexible, meeting current business needs and providing a foundation to meet new requirements. Designing flexible services enables the enterprise architecture to remain agile and accessible to business change and business opportunity. Service flexibility achieves the service-oriented design objective of service reuse, which is the most prevalent way that SOA adopters have recognized the SOA value proposition of faster development cycles, reduced development costs, and reduced software bugs.

Designing the Geocode Service

In the Job Finder solution, the design team needed to define a simple geocoding service that accepts a well-formed residential address and converts the address into geographic (x and y) location information that can be used by Web and desktop mapping applications.

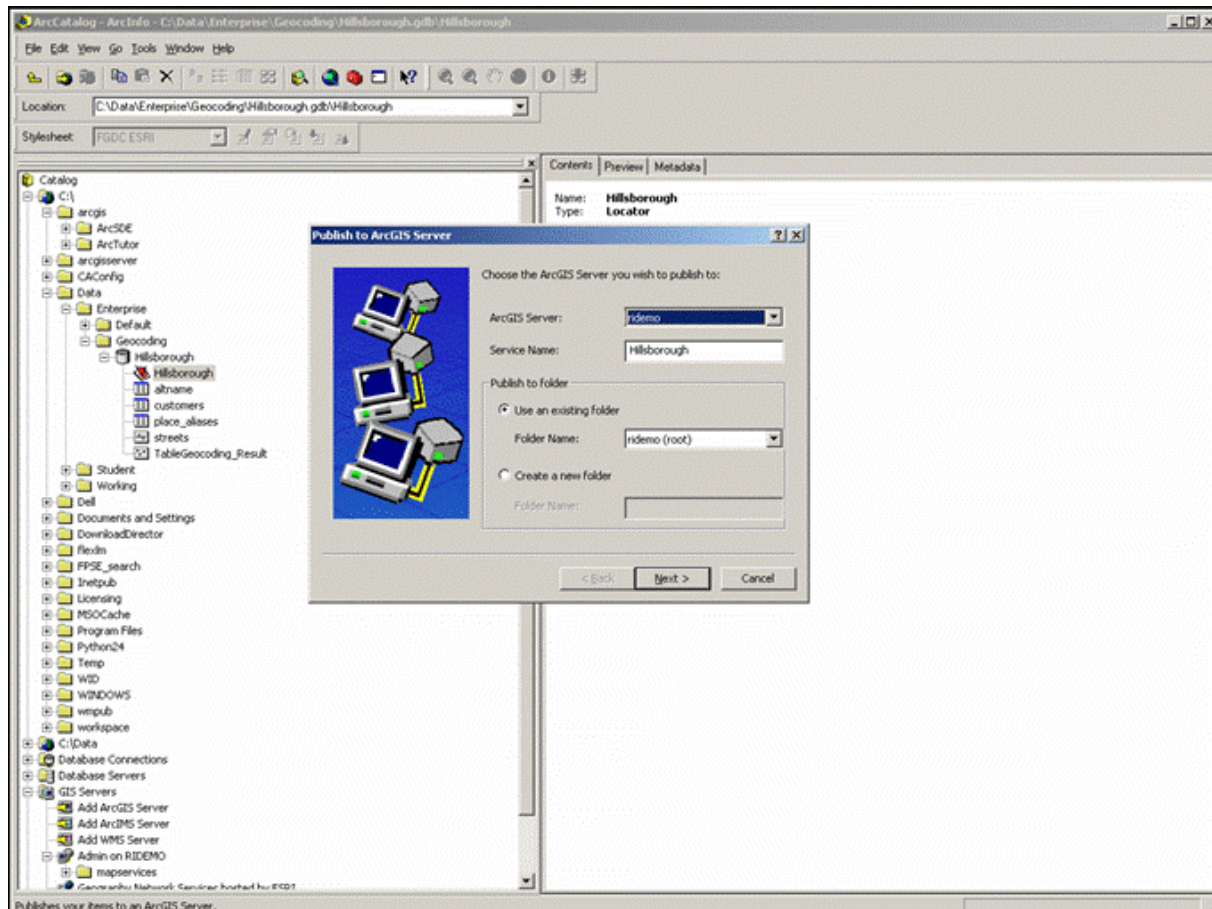
Adhering to the SOA design principle of service reuse, the development team considered other possible business uses of this type of information among different business units and across the organization. The team initiated the service design process by brainstorming different ways geocoding information is or might be used countywide in current and future solutions. The team quickly realized that it should expand the usability of this service by extending it to perform batch processing on addresses contained in a file or passed to the service as an array of addresses. The service logical design view for the county geocoding Web service is depicted in figure 3.

Figure 3
Geocode Service Design



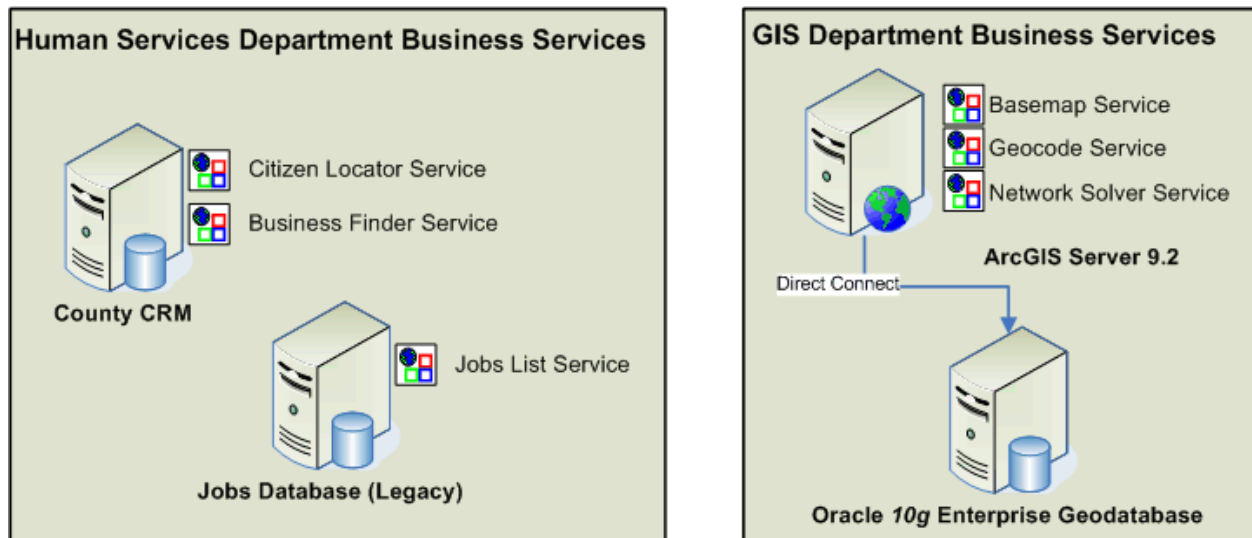
With the geocoding service designed, GIS analysts created a standards-based geocoding service using the geospatial tradecraft tools and functionality of the ArcGIS technology platform. Once the county geocoding service was instantiated according to the service-oriented design, GIS server administrators used the Web-based administration tools in ArcGIS Server to publish the geocoding service to the enterprise architecture as a standards-compliant GIS Web service (figure 4). After being published to the architecture, the GIS Web service was discoverable in the county UDDI online registry and was prepared for use in the Job Finder solution as well as in all other current and future county solutions that need flexible geocoding capabilities.

Figure 4
Publishing Geocode Service to ArcGIS Server



At that point in the project, the design team had selected multiple reusable services from the enterprise service portfolio and developed a new GIS service. Figure 5 illustrates the business and technology services used in the Job Finder solution and how the services align to different business units.

Figure 5
Service Business Alignment



The distribution and alignment of services across business units and the technology architecture ensures that the line of business responsible for the information has the authority and responsibility to manage, monitor, and support its business data and systems. These business owners provide access to the information, data, and products by exposing their underlying systems through a standards-based interface for other systems to discover, consume, and use.

Also of note in this service alignment diagram is the fact that each business unit is using different underlying enterprise technologies to manage and use its data. Using SOA integration patterns and techniques, differences in underlying implementation technologies do not pose an obstacle because the use of SOA interface standards hides the underlying technical details about the service from the requesting service consumer. Abstracting the underlying technology makes it easier to invoke and use the service without knowing or worrying about integration details. The implementation-neutral characteristic is another key design objective of SOA, which contributes to creating an agile IT architecture. This approach also means that organizations do not have to adhere to a single computing environment across the organization to achieve information sharing and integration.

Designing and Developing Composite Services

As depicted in figure 5, the team identified or constructed all the business services needed to support the Job Finder solution. Each service is independent in function and design, and each is a fully functioning unit of logic without any dependencies on another service or services. According to SOA design principles, a service is a well-defined business function that is encapsulated as a reusable software asset.

Services in an SOA must also be "composable," where one service enlists one or more additional services to complete a workflow consisting of a series of business tasks. In this way, each designed service has two capabilities:

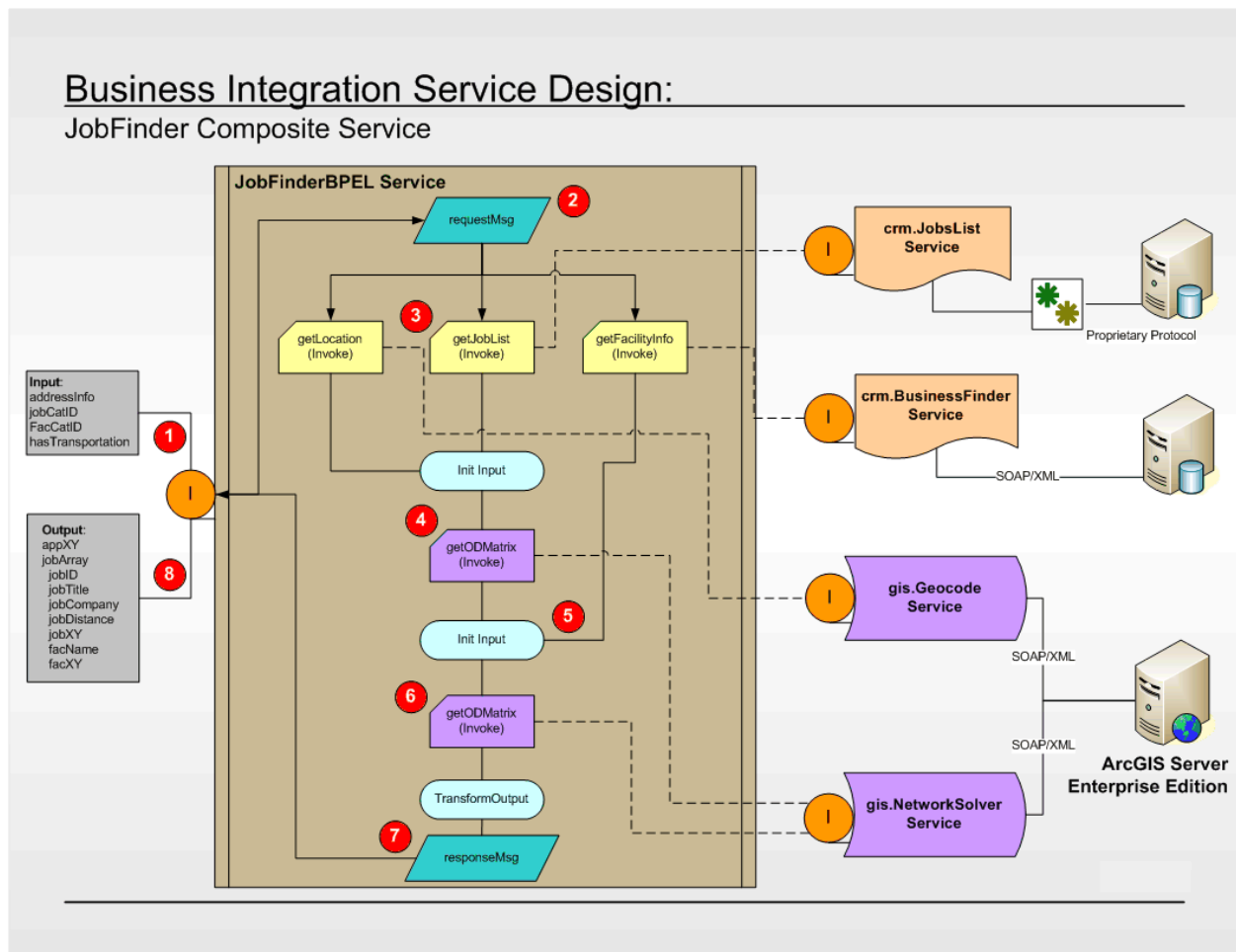
- A service must be designed in such a manner that it can enlist other services to accomplish a function without foreknowledge of what services it will incorporate.
- A service must be designed in such a manner that it can be pulled into a service composition by another independent service without foreknowledge of how it will be incorporated.

Service composition enables the automation of business processes and incorporation of business rules in workflows. When services are composed into an ordered set of interconnected functionality, they create a composite application that solves a business need. A composite application is an application element built by combining multiple services within a service-oriented architecture. As services are the underlying building blocks of SOA solutions, service composition is the glue that connects services in an ordered manner to accomplish business automation. This prescribed interaction, or defined connection of independent services, is referred to as service orchestration. Through the use of service orchestrations, service-oriented solution environments become inherently extensible and adaptive. They are so integrative and key to SOA that orchestrations themselves are instantiated as XML-based services and abide by the same standards for interoperability as business services do.

The primary industry specification that standardizes orchestration services is Business Process Execution Language (BPEL). BPEL is the standards-based platform for orchestrating independent services for business collaboration and business automation. BPEL is an XML-based language for the formal specification of business processes and business interaction protocols. In service orientation, BPEL is used to order and choreograph the interaction of reusable services similar to the way a conductor carefully choreographs musicians in an orchestra to interact with one another using music. BPEL services are also referred to as business integration services.

The design of BPEL services is accomplished by enterprise SOA architects. Figure 6 depicts the design of the BPEL module, or business integration service, used in the Job Finder SOA solution. Stepping through the flow of the BPEL service will illustrate how this module integrates multiple services from disparate systems and technologies into an orchestrated and well-defined automated business workflow.

Figure 6
Job Finder BPEL Service



The following is a summary of this process flow:

1. The Job Finder Web application invokes the Job Finder Composite Application service by passing an XML message according to the operations described in the WSDL service interface. The WSDL service interface defines the methods, parameters, data types, and binding information for the BPEL service.
2. The BPEL service receives the client request message and initializes variables used in the choreographed workflow.
3. Three Web services are invoked in parallel to accomplish three distinct business services: get a list of day care facilities, get a geocoded value for the job seeker's address, and get a list of jobs currently open from the legacy jobs database.
4. The output from the gis.Geocode service and the crm.JobsList is sent to the gis.NetworkSolver service. This service acts on the location information contained in

both of the service results and calculates distance, travel time, and routes to potential job locations. The output of the first use of the `gis.NetworkSolver` service becomes one of the inputs to the next operation in the business process.

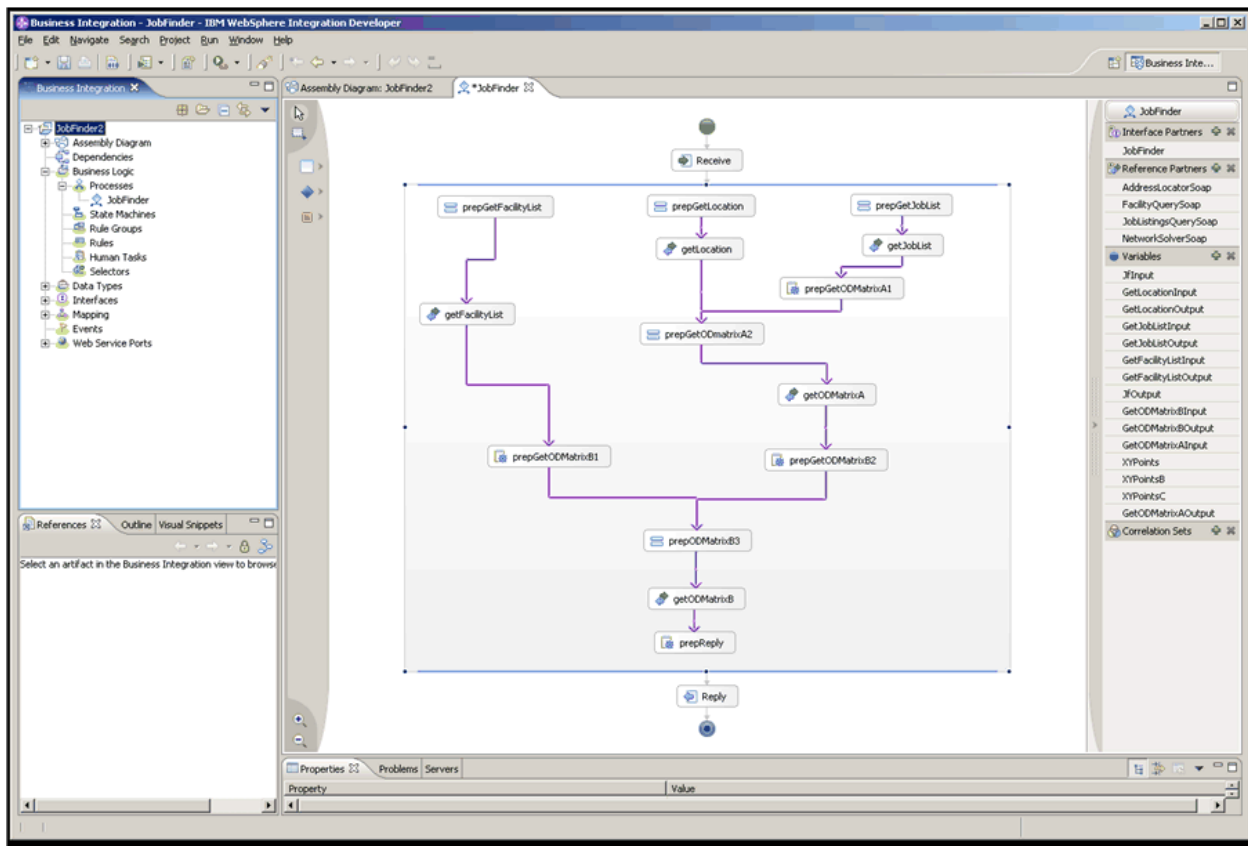
5. The list of day care facilities from the `crm.BusinessFinder` service is passed to the `gis.NetworkSolver` service. This time, the `gis.NetworkSolver` service determines which jobs have a day care facility located within two miles of the job site in accordance with the business rule. If the client request message indicates that day care is not needed (Boolean false), this second pass through the `gis.NetworkSolver` service is not executed.
6. The `gis.NetworkSolver` service identifies the jobs that satisfy the business rule of having a licensed day care facility located within two miles of the work site. The returned jobs are assembled and prepared to be included in the response message.
7. The response message is a well-formed XML message that is defined as the output in the BPEL WSDL service interface and is the output of the business process.
8. In the final step of the workflow, the requesting Web client receives the well-formed SOAP/XML response message to its service invocation sent in step 1 of the workflow. The Web client is responsible for receiving the response message, parsing it, and rendering the results in the presentation Web client application.

Once the BPEL service design was complete and validated against the desired workflow and information requirements, an integration developer was tasked to create the BPEL service in the integration environment. In this case study, WebSphere Process Server is the enterprise software used as the integration platform. The BPEL service and other services in the Job Finder solution were enabled and exposed to the enterprise via the integration platform. The next section discusses a high-level view of the development process for the Job Finder BPEL service.

The integration development team accomplishes the development activities to create the Job Finder BPEL service using the WebSphere Integration Development (WID) environment, a plug-in development module to the Eclipse open source integrated development environment (IDE). WID is used to code, test, and validate the functionality of the BPEL service. As depicted in figure 7, WID is a visual and intuitive development environment for constructing the business integration choreography service.

Within this BPEL module, multiple disparate systems and data sources share and exchange information through standard messages. These standards-based messages are defined and governed by SOA interoperability standards. These architectural messaging standards are the key advantage of an SOA approach: an integrated technology architecture that is focused on meeting current business needs yet remains flexible enough to change as business needs evolve and new business opportunities arise. The visual programming interface provided in WID assists the integration developer in quickly creating the business integration service. This intuitive development environment is also beneficial when the business process needs to be modified based on evolving business needs.

Figure 7
WebSphere BPEL Development Environment



Designing and Developing Job Finder Web Client

The last component for design and development was the Web client interface that county workers use to interact with the Job Finder solution. The ArcGIS Web Application Development Framework (ADF™) .NET for the Visual Studio® 2005 development environment was used to create the Web application.

A view of the Web client application main page is provided as figure 8. The county worker populates the contents of this form by calling the `crm.CitizenLocator` service to retrieve address information for the resident. The worker completes the form by selecting the job category of interest and indicating whether the job seeker is using public transportation or requires day care. Each of these selections is completed using check boxes or a dynamically populated combination drop-down list. The county worker submits the completed form to the network as a SOAP/XML message. The input message invokes the Job Finder business process by calling the BPEL service on the enterprise (WebSphere) integration platform.

Figure 8
Job Finder Web Client Search Form

The screenshot shows a web browser window titled "Hillsborough County Florida County Employee Site - Windows Internet Explorer". The address bar shows the URL "http://wddb10.esri.com/employmentassistancev2/". The page header is "Hillsborough County Employment Department". On the left, there is a sidebar with "Employment Resources" and "County Resources". The main content area is titled "For a returning client, enter ID below and submit to retrieve information". It contains a "Client ID:" label, a text input field with "PATT00011", and a "Get Info" button. To the right, there is a section titled "Current Information On record" with fields for Name, Street, City, State, Zip Code, and Phone. Below these fields are checkboxes for "Has Vehicle" (checked) and "Needs Daycare" (unchecked). At the bottom, there is a "Main Job Interest Category" dropdown menu with "Food and Lodging" selected, and a "Find Jobs" button.

Hillsborough County Employment Department	
Employment Resources <ul style="list-style-type: none">County Job InformationFlorida State Job OpportunitiesUS Job BankFederal Job Information County Resources <ul style="list-style-type: none">Departments and AgenciesA-Z IndexOther Government Websites	<p>For a returning client, enter ID below and submit to retrieve information</p> <p>Client ID: <input type="text" value="PATT00011"/> <input type="button" value="Get Info"/></p> <p>Current Information On record</p> <p>Name: <input type="text" value="Alice Patterson"/></p> <p>Street: <input type="text" value="706 W PARK AV"/></p> <p>City: <input type="text" value="Tampa"/></p> <p>State: <input type="text" value="FL"/></p> <p>Zip Code: <input type="text" value="33602"/></p> <p>Phone: <input type="text" value="813-272-8169"/></p> <p><input checked="" type="checkbox"/> Has Vehicle <input type="checkbox"/> Needs Daycare</p> <p>Main Job Interest Category <input type="text" value="Food and Lodging"/></p> <p><input type="button" value="Find Jobs"/></p>

After the main page form is submitted, the Job Finder BPEL process runs through the orchestrated business process to create a list of potential jobs for the requesting candidate. Once the filtered list of jobs is created and assembled via the integration middleware, WebSphere passes the output of the BPEL business process back to the Web client as a SOAP/XML response message. Since this is a short-duration request that runs very quickly, the client is configured to make a synchronous call to the BPEL business service. When the business service completes execution, its results are immediately returned to the requesting client for rendering.

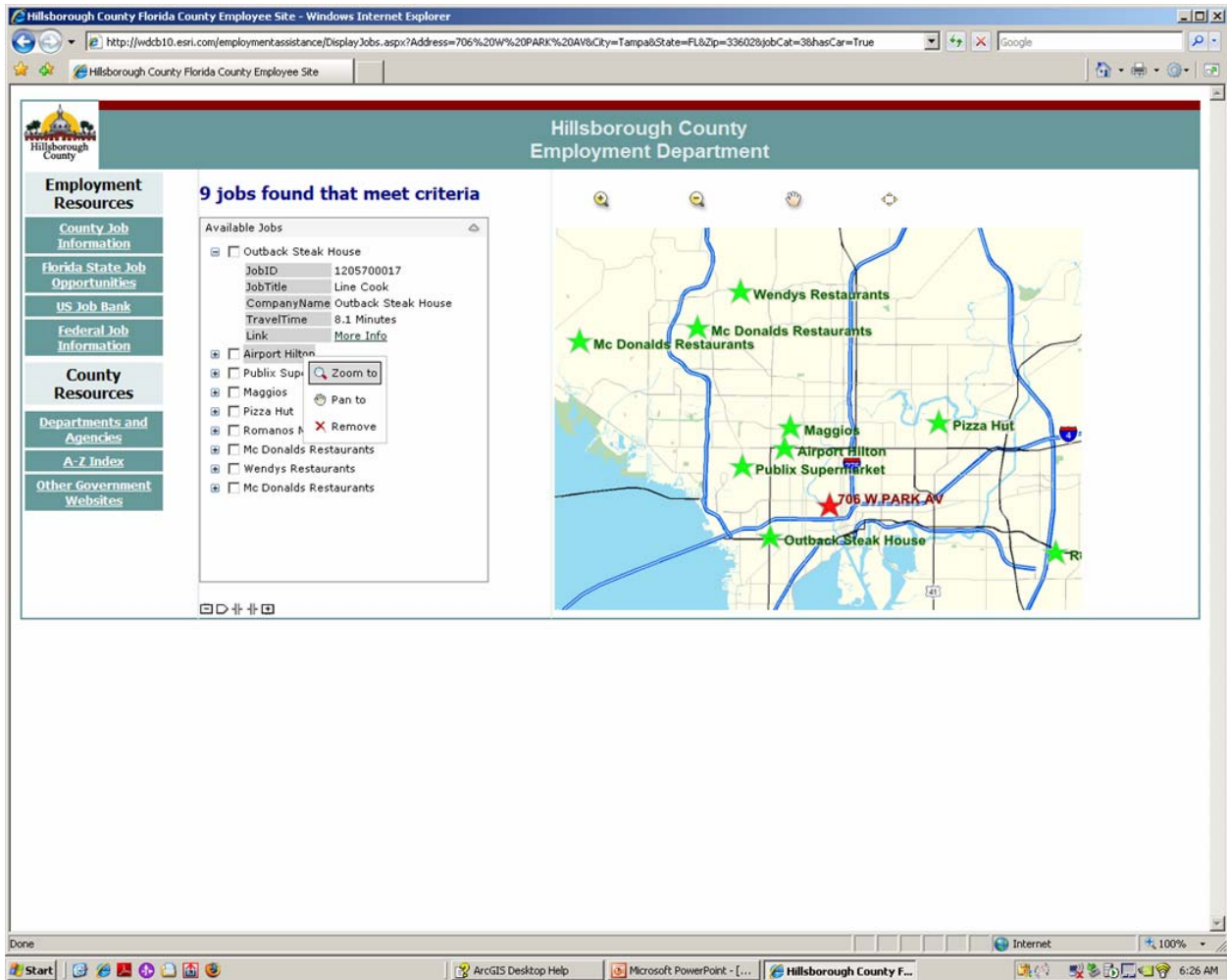
Map and geodata services from ArcGIS Server are responding to the GIS service requests contained in the BPEL business service. The customer relationship management (CRM) systems are responding to the requests as well. Upon receipt of the response message from the integrated business process and from the map and geodata services, the Web ADF client parses the SOAP/XML messages and dynamically populates the data and map controls to create an easy-to-use and intuitive Web page display of the text and map-based information that is now represented as an integrated solution.

An example of the Web client results page is shown in figure 9. The client contains a map control consisting of the job seeker's residence, locations of the jobs, and the detailed travel routes along public or private transportation networks. The client also contains a

J-9666

data grid that makes the textual information about the job, its location, travel time, and business name readily available to the job seeker. Job results are sorted based on travel time distances calculated by the GIS server.

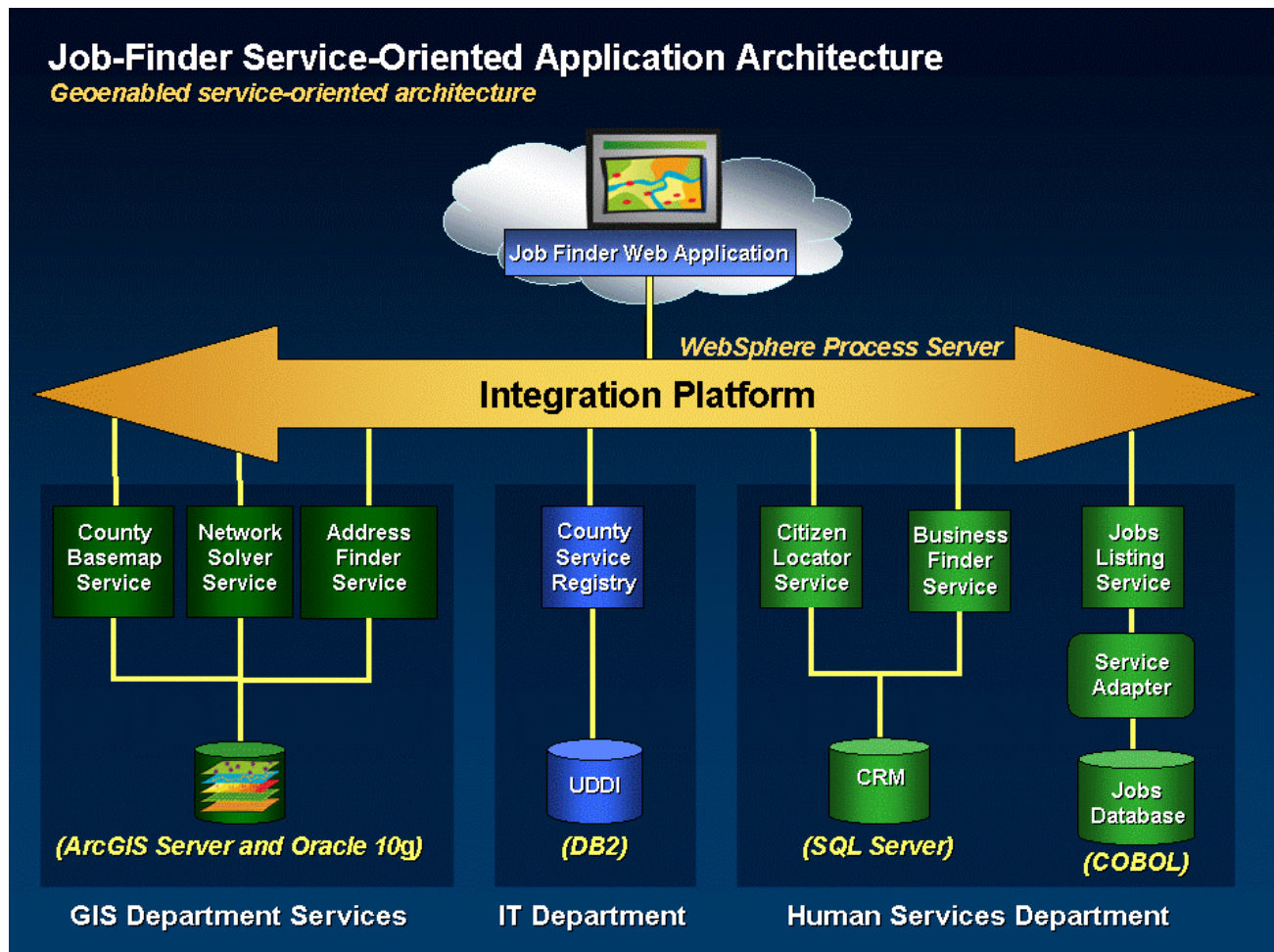
Figure 9
Geoenabled Job Finder Search Results



Summary

This case study has described a business modernization project that allowed an outdated business process to be significantly improved through the use of Web services, geospatial information systems and data, integration middleware, and an SOA design approach to enterprise architecture. Two business units with disparate underlying technologies and different workflows were integrated through a service-oriented design approach to synchronize systems, data, processes, and people. Figure 10 is a view of the final geoenabled service-oriented application architecture of the Job Finder business solution.

Figure 10
Job Finder Application Architecture



Benefits Realized through Service Orientation

The strategic IT direction and governance policies provided by the county CIO explicitly directed all participating departments to make new investments in IT programs and plans aligned with the county's enterprise-wide strategy for business solutions based on the tenets and principles of SOA. Using the IT policy and governance plan, the project team delivered a solution built on an agile underlying technical infrastructure that will allow flexibility to meet business changes and opportunities. The use of standard Web services, BPEL business automation language, and SOA-enabled technologies, such as ArcGIS and WebSphere Process Server, is allowing the county to understand key business processes, create the disciplined services that deliver them, and enforce the proper use of the underlying architecture. Specifically, the following benefits of service orientation are realized through the Job Finder application:

- **Improved application integration:** Like in most organizations, application integration in the county is at the core of many IT challenges. As demonstrated in this project, service-oriented design and Web service technologies make it possible

to connect many different computing platforms, software environments, and business units.

- **Information and process sharing:** The standards of service orientation allow the county to effectively share business data and processes between different business units and with the public and other organizations as business needs require.
- **Reuse and flexibility:** As demonstrated, existing components are easily integrated regardless of the underlying technology and implementation details. These advantages help realize cost and risk reductions as well as improve agility in anticipation of future changes.
- **Business process standardization:** In the course of the project, it was important for the business owners and managers to define existing and desired business needs. Through this process, the workflows, information sources, and use cases are better understood and documented.
- **Standardization and openness:** The wide-scale adoption of the standards that underlie the specifications of service orientation makes this a solid choice for continued success in implementing durable solutions with reduced risk of proprietary traps. Leveraging commercial technologies, like ArcGIS, that embrace the standards of service orientation ensures interoperability and integration benefits.

The success of the Job Finder service-oriented solution is creating new levels of cooperation and interest across the organization. Business leaders realize there are many more application scenarios that can benefit from service orientation and are providing creative project ideas to the county CIO for consideration.