Warfighter's Information Packager

Yigal Arens
Weixiong Zhang
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

{arens,zhang}@isi.edu

Yongwon Lee
Jon Dukes-Schlossberg
Lockheed Martin
Intelligent Systems Center
3251 Hanover St. H1-43
Palo Alto, CA 94304
{ylee,slosh}@ict.atc.lmco.com

Marc Zev
ISX Corporation
4353 Park Terrace Dr.
Westlake Village, CA 91361

mzev@isx.com

Abstract

The Warfighter's Information Packager (WIP) is a suite of distributed components that allows users to easily obtain information from diverse heterogeneous data sources and to display the results in a user-defined predictable manner.

WIP is based on research performed under DARPA's Intelligent Integration of Information program. WIP uses a combination of AI and non-AI technologies to take advantage of the information push technology being developed for DARPA's Battlefield Awareness and Data Dissemination (BADD) program, and being deployed during the Fall of 1998.

Together, the WIP components create a distributed system that serves as a valuable tool for information analysis by: 1) allowing the user to define high-level information products, *information packages*, which are parameterized by user interests and specific tasks and roles; 2) providing a web-based package viewer that dynamically constructs packages for the user on demand and performs value-added information *linking*; 3) allowing users to make high-value complex information requests that can span multiple data sources without a priori knowledge of the schema of the sources,; and 4) monitoring data sources and *anticipating* useful modifications to a user's information package.

1 Problem Description

The Warfighter's Information Packager (WIP) system addresses the problem of how to support the needs of users to view and manipulate required data in an information push environment. In such an environment, traditional query-based data retrieval is replaced by asynchronous information delivery based on information profiles registered with sources.

Information push is intended to address the following problem: When information is generated in many sources and on a continual basis, it is onerous and inefficient to require users to issue multiple repeated queries for their information. It would be far more efficient if responsibility for disseminating the information were transferred to the sources. The sources contain the actual information and can more readily detect changes. In such an architecture, the consumer registers an *information profile* with the source, or with a system overseeing the source, that describes the kind of information they need.

There are many ways in which such a scheme may be implemented. The BADD program was established to deal with the issue of information push in a battlefield environment. BADD has chosen an architecture wherein profiles are registered with a central Information Dissemination Manager (IDM), and the IDM ensures that data matching the profiles is transferred to the deployed sources. This still leaves open the questions of how to construct and use applications that require the pushed data, and how to determine which profiles must be registered in order to support these applications. The WIP system addresses these issues.

WIP provides end-users with a facility for producing an *information package*. WIP, using IDM, ensures that necessary data is pushed to deployed sources when it is available. An *Information Integrator* satisfies information package queries by retrieving data from the deployed sources. Throughout both the specification and utilization of the product, an *Anticipator* component observes user and world events and may trigger the modification of existing packages. This capability allows profiles to be updated dynamically, in response to changing circumstances.

The specific problems that WIP addresses are:

1. The warfighter has a task to perform and has specific information needs. Information needs in BADD are specified as profiles that do not necessarily match up conveniently with actual source queries. This is due to the profile registration facility of the IDM having a language far less powerful than familiar database query languages. Furthermore, the context in which the task ends up being performed may require variation of the profile—something that is hard for the end-user to anticipate.

2. The data is delivered to the end-user's computer environment in formats that may differ from one environment to the next due to variations in data management facilities. For example, one warfighter may have a database facility available for a certain type of data,

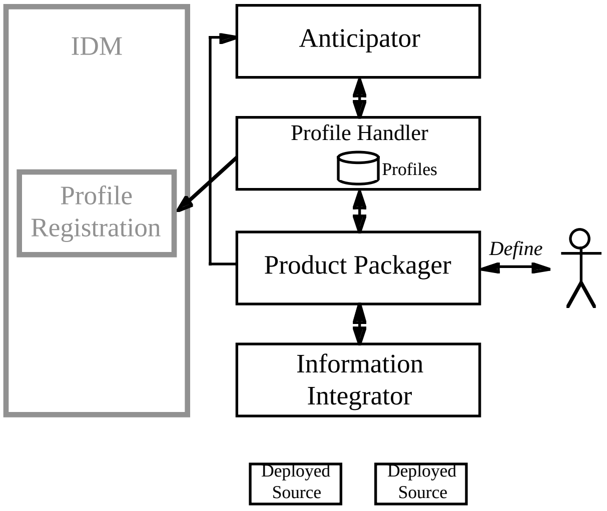

Figure 1: Information Package Specification

while another may not. In such a situation the IDM will deposit the information in a file. For similar reasons, data storage formats used in the deployed sources may differ from those used in original sources against which the profiles are registered. In addition, data needed for a single task may end up being distributed over different deployed sources. The end result is that any processing of deployed information will necessarily involve an information integration and/or translation task. Making WIP packages portable requires that this integration be done in a general manner and that queries to the information integrator be at a semantic level, independent of precise sources used and their organization.

We have chosen to use selected AI technologies, some developed as part of DARPA's Intelligent Integration of Information (I3) program, to address this problem.

2 Application Description

This section describes WIP's *product specification phase* and *product utilization phase*. Both phases complement each other, and the IDM, to provide an end-to-end information push system.

2.1 Information Package Specification

Figure 1 presents the architecture of the product specification phase. During this phase, the user interacts with the Product Packager to define the information package and the information profiles ultimately required for its successful utilization. It consults the Information Integrator in order to determine which sources are to be monitored. Since the local, deployed sources are mirrors of the original sources accessible through the IDM, (even if their organization and DBMS may differ) the Information Integrator can provide the Product Packager with necessary information to locate the appropriate sources. Defined profiles are handed to the Package Handler for local storage and for IDM registration. They are also handed to the Anticipator to infer user and world events that would necessitate modification of the information package.

The following sections describe the operations and capabilities of each of the modules in Figure 1 in more detail.

2.1.1 Product Packager

The Product Packager: 1) allows the user to define high-level information products, *information packages*, which are parameterized by user information needs and specific tasks and roles; 2) describes how the information should be formatted when presented to the user; 3) provides a web-based viewer that dynamically constructs packages for the user on demand and performs value-added information linking.

The Product Packager module, developed at ISX, is composed of two main components: the Package Editor and the Product Viewer. The Package Editor provides an environment for creating information package templates. These templates include parameterized queries and display formatting information. WIP's query description methodology allows users to easily describe their information needs based on a descriptive domain ontology. WIP's parameterization allows the dynamic modification of the precise semantics of queries without revising queries. This is accomplished by providing late-time binding of variables within the query expressions, enabling information requirements such as: "Get the target information for all targets on today's target list" to be expressed.

The Package Editor also performs the registration of necessary data profiles so that data can be deployed to the user's system. The Product Viewer, discussed more thoroughly in Section 2.2.1, collects the information results from the information integrator module, formats them based on the format defined in the package template, and displays the packaged product to the user.

2.1.2 Information Integrator

The Information Integrator module is an application of SIMS AI technology, developed at USC/ISI. SIMS serves as a single access point for information distributed over a collection of heterogeneous data sources. The underlying technology is described in Section 3.2. During the specification phase, the Information Integrator uses its knowledge of the distribution of data over sources accessible to the IDM (also mirrored, as noted earlier, by the distribution of potential data over local deployed sources) to provide the Product Packager with the identity of sources against which profiles must be registered. This enables the package being defined to have the information required for its operation at utilization time. Furthermore, SIMS' capability to determine which queries will have to

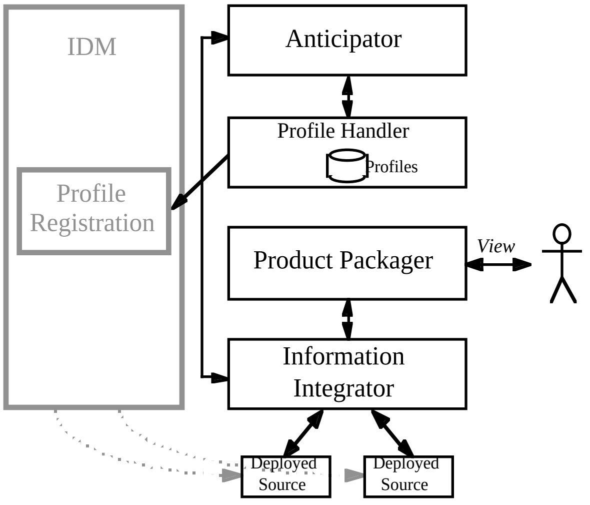

Figure 2: Information Package Utilization

be asked of each component source to satisfy the package's needs is used by the Product Packager. The package defines the profiles that will eventually be handed to the Package Handler for processing and registration.

2.1.3 Profile Anticipator

The Profile Anticipator dynamically updates user information needs in response to circumstances that have changed since the original information profile was defined. It handles conditional and unexpected information needs of users, once their basic information needs are registered. "Basic" information refers to required information to be delivered at all times, while "conditional" information refers to data to be delivered only when a certain condition is met (e.g., only when the mission is completed, send me the video image of the target). The Anticipator is an integral part of the WIP system not only to achieve dynamic intelligent information packaging but also to simplify the complexity of package entry.

Upon notification of an information profile registration, the Anticipator examines its contents and generates an anticipator profile for that user. An anticipator profile specifies the conditions under which the user's profile needs to be modified. For example, as a user moves from one location to another, the Anticipator monitors the user's current location to update information needs that may change. The anticipator profile is registered through the Product Packager just as the user's information profile is registered. The Anticipator then generates a package monitor that continuously and periodically requests the data for this package from the Product Packager to check the occurrence of "significant" and "interesting" events. When such events are detected, the monitor notifies the Anticipator that then carries out appropriate actions, i.e., the user's information profile is updated. When a user's information profile is deactivated, the associated anticipator profile and its package monitor are also deactivated.

The Anticipator maintains a rule base to provide domain-specific knowledge necessary for generating the anticipator profiles and to specify definitions of "interesting" events and associated appropriate actions. The Anticipator rules are represented in OEM (Object Exchange Model [Goldman et al. 96]), and implemented in Java.

2.1.4 Package Handler

The Package Handler maintains a local store of profiles defined specifically by WIP, handles communications, and registers and de-registers these profiles with the IDM.

2.2 Information Package Utilization

In the package utilization phase, packages are run and their information can be viewed and manipulated by the end-user. The package typically needs specific information—it requests data from the Information Integrator, which queries the deployed sources. The Profile Anticipator will monitor those same deployed sources via the Information Integrator in order to update users' information profiles, if and when necessary.

Figure 2 presents the architecture for the information package utilization phase. The following sections describe the operations and capabilities of each of the modules in more detail.

2.2.1 Product Packager

During the package specification phase, the Package Editor was responsible for user interactions. However, during the package utilization phase the Product Viewer is the focus of the system's operation. During the package specification phase the Product Packager enhances a user's ability to analyze information by allowing the creation of query expressions based on a high-level domain model, and by defining the formats of results both on a query by query basis and the package as a whole. In this phase, the Product Packager manages the returned information in order to provide the user with additional semantic insight to the data.

Using the Product Viewer, the user requests that a package be displayed. The Product Packager retrieves the requested package from the Package Handler and submits the package's queries to the Information Integrator. Upon receipt of the queried information, the Product Packager aids the user by identifying semantic links among the user's requested information and that requested by others, including information gathered on behalf of the Anticipator. The Product Packager then formats the information as specified in the package and renders the package, dynamically, based on the user's currently available bandwidth and display options, using the Product Viewer.

While formatting and rendering the package, the Product Packager adds hyperlinks to the value-added linked information.

2.2.2 Information Integrator

In this phase the Information Integrator performs its "standard" function of query answering. It receives queries from the Product Packager and the Anticipator for data in the local sources, and provides it. The queries are in a high-level language using terms from the domain model (see Section 3.2 for details), hiding from the other WIP modules the details of source distribution, organization and language. The same package will work in different environments as long as the Information Integrator's description of the available sources is revised to reflect the specifics of the local situation.

2.2.3 Profile Anticipator

Because the Anticipator can add and update a user's information needs, information packages can contain new information the user does not specify in his/her profile. The operation of the Profile Anticipator is not modified during the package utilization stage, and it performed its monitoring of data sources regardless of user interactions with the Product Viewer.

2.2.4 Package Handler

During the utilization phase, the Package Handler acts the same as during the specification phase.

3 Use of AI Technology

3.1 Product Packager and Profile Editor

3.1.1 Information Package Template Selection

In military situations, such as the one WIP is designed for, it is typical that many users have the same or a similar data capacity. For this reason and for the efficiency of use, having pre-staged information package templates (IPTs) would be advantageous. However, because it is impossible to predetermine the information needs of every user, a better way is to establish IPTs that reflect the information needs based on an encapsulation of a particular user role. For instance, it would be far easier to determine the general information needs of someone in the Bosnian theater and store that as a specific IPT, store the information needs of a tank commander in a separate IPT, etc. Then, when the user needs to register an information package he needs only describe himself by his roles (e.g. tank commander in the Bosnian theater) and a specialized IPT can be constructed based on the aggregation of smaller, specific information package templates.

The ability to perform the aggregation of IPTs relies on the definition of roles for both the users and the IPTs. The purpose of a role is to provide a meaningful way of recognizing an IPT by the type of information requests contained within it. An IPT's role does not necessarily identify it uniquely, but it serves as a way to measure the commonality between that IPT and others. The idea is to define a role as some set of attributes common to all IPTs. Once a role is established for an IPT it is used as the index to find it again.

Collecting the appropriate IPTs for a user is not as simple as performing a *find* in a relational database. The determination of which IPTs are suitable can be based on a data-driven, rule-based methodology. Within military operations there are very specific rules about how information can flow, typically this means up through the chain of command. An example how this might work is as follows: a tank company commander in the Bosnian theater is interested in creating an appropriate IPT. An appropriate set of IPT might be: one for tank commanders, one for the Bosnian theater, one for the army. However, an IPT for a battalion would not be appropriate. The battalion commander in the same theater might get the tank commander's IPT plus IPTs for artillery companies. But if no IPT exists for the battalion level for Bosnia, a battalion level IPT for Europe might be appropriate. The IPT selection process is defined within an expert system, using a rule-based domain ontology.

3.1.2 Semantic Linking of Information

The Product Packager manages the information that is returned from the information integrator for all users of the WIP system at a given deployed site. As the Package Products are requested and viewed via the Product Viewer, a request is made to the Information Integrator to gather the data. Using a rule-based domain ontology as a guide, the Product Packager reviews all the data gathered from all users and makes a determination whether the data is relevant to the information requested in the original package. For instance, if the package has been registered to a tank unit and weather data has been collected and it is raining, the Product Packager would determine that road conditions would be a useful piece of information to include. If road conditions for the appropriate geographical area are available, then the Product Packager will add a hyper-inked connection to the original package pointing to the value added data.

3.2 Information Integrator

The Information Integrator module is an application of SIMS technology in the battlefield data dissemination domain. The overall goal of the SIMS project at USC/ISI is to provide integrated access to information distributed over multiple, heterogeneous sources: databases, knowledge bases, flat files, Web pages, programs, etc. In providing such access, SIMS insulates human users and application programs from the need to be aware of the location of sources and distribution of queried data over them, individual source query languages, their organization, data model, size, and so forth. The processing of user requests should be robust, capable of recovery from execution-time failures and able to handle and/or report inconsistency and incompleteness of data sources. At the same time SIMS has the goal of making the process of incorporating new sources as simple and automated as possible.

The SIMS approach to this integration problem has been based largely on research in Artificial Intelligence, primarily in the areas of knowledge representation, planning, and machine learning. A model of the application domain is created, using a knowledge representation system to establish a fixed vocabulary for describing objects in the domain, their attributes and relationships among them. Using this vocabulary, a description is created for each information source. Each description indicates the data model used by the source, the query language, network location, size estimates, etc., and describes the contents of its fields in relation to the domain model. SIMS' descriptions of different information sources are independent of each other, greatly easing the process of extending the system. Some of the modeling is aided by source analysis software developed as part of the SIMS effort.

Queries to SIMS are written in a high-level language (a subset of SQL or Loom) using the terminology of the domain model – independent of the specifics of the information sources. Queries need not contain information indicating which sources are relevant to their execution or where they are located. Queries do not need to state how information present in different sources should be joined or otherwise combined or manipulated.

SIMS uses a planner to determine how to identify and combine the data necessary to process a query. In a pre-processing stage, all data sources possibly relevant to the query are identified. The planner then selects a set of sources that contain the queried information and generates an initial plan for the query. This plan is repeatedly refined and optimized until it meets given performance criteria. The plan itself includes, naturally, sub-queries to appropriate information sources, specification of locations for processing intermediate data, and parallel branches when appropriate. The SIMS system then executes the plan. The plan's execution is monitored and replanning is initiated if its performance meets with difficulties such as unexpectedly unavailable sources. It is also possible for the plan to include explicit replanning steps, after reaching a state where more is known about the circumstances of plan execution.

Changes to information sources are handled by changing source descriptions only. The changes will automatically be considered by the SIMS planner in producing future plans that utilize information from the modified sources. This greatly facilitates extensibility.

A variety of detailed publications describing SIMS is available (e.g., [Arens et al 96], [Arens et al 93], [Knoblock 95], [Knoblock 94]).

3.3 *Profile Anticipator*

The current prototype Anticipator module, developed at LM/ISC, implements data-driven, rule-based anticipation of information needs: data sources are continuously monitored for occurrences of interesting events. When such events occur, one or more anticipation rules are fired to update user's information needs. Researchers in both database (source subscription) and AI (knowledge engineering and representation) have enabled the current approach.

The anticipator uses domain-specific rules to determine data sources to be monitored, conditions (interesting events) to be checked, and actions to be executed when conditions are met. The anticipator rules are obtained through standard knowledge engineering techniques with military experts. All rules are deterministic, i.e., there is no probabilistic inference. The rules are described using high-level domain terms and the translation of these terms to source specific terms is performed by the Information Integrator module. A key feature of anticipator rules is their expressive event descriptions. Currently, a limited set of event descriptions is possible including comparisons of values of multiple attributes in two information packages.

The rules are represented in OEM, taking advantage of its flexibility (less structural constraints) and self-declarative and self-describing features. It is worth noting that anticipator rules contain neither data source nor implementation specific information. For example, to generate source monitors, rules specify sources and possibly attribute names, but not database queries in some query language (e.g., SQL).

The WIP anticipator module along with the Package Handler is an extension of Profile and Event Manager components of Intelligent Information Dissemination Server (IIDS) [Dukes-Schlossberg et al., 97], and is an implementation of an ongoing effort toward a fully automated information profiling system. We are currently designing an advanced information profiling system, which uses both Bayesian networks and anticipation rules to predict user's information needs from their identity information such as user type, location, status, and mission. That is, rather than explicitly asking what information is needed, users' information needs are predicted and their information profiles automatically constructed from what and where they are, and what mission they are engaged in.

4 Application Innovation

The design of the BADD system, as envisioned by DARPA and executed under its guidance, does not involve any technology that could be classified as part of Artificial Intelligence. The participants in the effort described in this paper found that the incorporation of additional technology based on AI research and techniques could relatively easily (e.g., using an order of magnitude smaller funding) provide a very substantial added value. The information integration functionality provided by the Information Integrator, which uses AI planning technology, enables WIP to be operational on different information sources in multiple existences, including databases, legacy information systems and the World Wide Web. The semantic information linking function of Product Packager and anticipating function provided by the Anticipator, both of which use domain ontology and rule-based AI technologies, significantly extend WIP's usability to the end users. Combined, these two functions not only provide information that directly meets users' needs, but also supply and present context and other related information that makes the required information meaningful.

We would also like to point out that although WIP is developed for the BADD system in a military domain, the concept and WIP system itself can be easily applied to other applications. Specifically, WIP can be adapted to a personal digital assistant in an information rich environment to help a user collect information based on particular information needs. The information can then be presented in a format compliant with the user's preferences.

5 Evaluation

The WIP system is the integration of three distinct technologies: information integration, product packaging and information anticipation. The development of each of these technologies has been pursued independently, and the integration of components has taken place in the past between some of the contributors. Prototype systems with limited functionality exist for the components of the Product Packager. The Information Integrator (SIMS), responsible for satisfying information requests, is currently supporting several projects in extended prototype form. The Anticipator, which is responsible for anticipating users' information needs, is currently in development. The development schedule currently places the integration of WIP with the rest of the BADD system during the Summer of 1998. BADD will then be deployed as a working system in the Fall of 1998.

References

[Arens et al. 96] Yigal Arens, Craig A. Knoblock, and Wei-Min Shen. *Query Reformulation for Dynamic Information Integration.* Journal of Intelligent Information Systems, Vol. 6, No. 2/3, 1996.

[Arens et al. 93] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu, and Craig A. Knoblock. *Retrieving and Integrating Data from Multiple Information Sources.* International Journal of Intelligent and Cooperative Information Systems. Vol. 2, No. 2. Pp. 127-158, 1993.

[Dukes-Schlossberg et al. 97] Jon Dukes-Schlossberg, Yongwon Lee, and Nancy Lehrer. *IIDS: Intelligent Information Dissemination Server.* Proceedings of MILCOM 1997.

[Goldman et al 96] Roy Goldman, Sudarshan Chawathe, Auturo Crespo, Jason McHugh. *A Standard Textual Interchange Format for the Object Exchange Model (OEM)*. Technical Report, Stanford University, 1996.

[Knoblock 95] Craig A. Knoblock. *Planning, Executing, Sensing, and Replanning for Information Gathering.* Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995.

[Knoblock 94] Craig A. Knoblock. *Applying a General-Purpose Planner to the Problem of Query Access Planning.* Proceedings of the AAAI Fall Symposium on Planning and Learning: On to Real Applications, 1994.