

VFILM: A Value Function Driven Approach to Information Lifecycle Management

Jeffrey Cleveland^a, Joseph P. Loyall^a, Jonathan Webb^a, James Hanna^b, Shane Clark^c

^aBBN Technologies, 10 Moulton Street, Cambridge, MA 02138

^bUS Air Force Research Laboratory, 525 Brooks Rd, Rome, NY 13441

^cUniversity of Massachusetts, 140 Governors Drive, Amherst, MA 01003

ABSTRACT

Information Management (IM) services need lifecycle management, i.e., determining how long persistent information is retained locally and when it is moved to accommodate new information. This is important when bridging IM services from enterprise to tactical environments, which can have limited onboard storage and be in highly dynamic situations with varying information needs. In this paper, we describe an approach to Value Function based Information Lifecycle Management (VFILM) that balances the value of existing information to current and future missions with constraints on available storage. VFILM operates in parallel with IM services in dynamic situations where missions and their information needs, the types of information being managed, and the criticality of information to current missions and operations are changing. In contrast to current solutions that simply move the oldest or least frequently accessed information when space is needed, VFILM manages information lifecycle based on a combination of inputs including attributes of the information (its age, size, type, and other observable attributes), ongoing operations and missions, and the relationships between different pieces of information. VFILM has three primary innovative features: (1) a fuzzy logic function that calculates a ordering of information value based on multiple relative valued attributes; (2) mission/task awareness that considers current and upcoming missions in information valuation and storage requirements; and (3) information grouping that treats related information collectively. This paper describes the VFILM architecture, a VFILM prototype that works with Air Force Research Laboratory IM services, and the results of experiments showing VFILM's effectiveness and efficiency.

Keywords: Information Lifecycle Management, Hierarchical Storage Management

1. INTRODUCTION

Bridging Information Management (IM) services from enterprise to tactical environments requires lifecycle management to determine how long information is retained in limited onboard storage and when it is moved to make room for new information. If IM systems, such as the USAF Phoenix IM Services [8], do not provide support for cleaning up information repositories that are reaching their saturation point, information will be lost (in an unmanaged manner), archive operations will fail, software exceptions will be thrown, or in the worst case the IM services will fail.

Repositories will fill up and are likely to do so when they are needed the most, even with modern disks with the capacity of many Gigabytes or Terabytes. Consider that during Operation Anaconda in March 2002, U.S. air forces flew 65 combat sorties per day [15]. Thirty minutes of ISR video from a UAV in compressed MPEG-2 format requires 1.2 GB of disk space. A single high resolution RGB image in TIFF format (2248x2080 pixels) such as might be used for battle damage assessment or aimpoint generation requires over 13 MB of space.

An Information Lifecycle Management (ILM) service should manage how and what gets retained in each storage level, so that

- Critical information urgent to ongoing and upcoming missions is readily accessible.
- High speed, high cost storage is used for information that is most critical to ongoing and upcoming missions.
- Movement of information is based on a decrease in value to ongoing or upcoming missions and storage space being needed for higher value, more critical information.
- Support for information repositories and IM operations, e.g., query and archive, is maintained.

Report Documentation Page		Form Approved OMB No. 0704-0188
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.		
1. REPORT DATE APR 2011	2. REPORT TYPE	3. DATES COVERED 00-00-2011 to 00-00-2011
4. TITLE AND SUBTITLE VFILM: A Value Function Driven Approach to Information Lifecycle Management		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER	
	5e. TASK NUMBER	
	5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Raytheon BBN Technologies,Cambridge,MA,02138		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		
13. SUPPLEMENTARY NOTES In Proceedings of the SPIE Conference on Defense Transformation and Net-Centric Systems, Orlando, FL, April 25-29, 2011. U.S. Government or Federal Rights License		
14. ABSTRACT Information Management (IM) services need lifecycle management, i.e., determining how long persistent information is retained locally and when it is moved to accommodate new information. This is important when bridging IM services from enterprise to tactical environments, which can have limited onboard storage and be in highly dynamic situations with varying information needs. In this paper, we describe an approach to Value Function based Information Lifecycle Management (VFILM) that balances the value of existing information to current and future missions with constraints on available storage. VFILM operates in parallel with IM services in dynamic situations where missions and their information needs, the types of information being managed, and the criticality of information to current missions and operations are changing. In contrast to current solutions that simply move the oldest or least frequently accessed information when space is needed, VFILM manages information lifecycle based on a combination of inputs including attributes of the information (its age, size, type, and other observable attributes), ongoing operations and missions, and the relationships between different pieces of information. VFILM has three primary innovative features: (1) a fuzzy logic function that calculates a ordering of information value based on multiple relative valued attributes; (2) mission/task awareness that considers current and upcoming missions in information valuation and storage requirements; and (3) information group-ing that treats related information collectively. This paper describes the VFILM architecture, a VFILM prototype that works with Air Force Research Laboratory IM services, and the results of experiments showing VFILM's effectiveness and efficiency.		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 12	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

We have developed a Value Function based Information Lifecycle Management (VFILM) prototype with three novel advances over current ILM and HSM systems:

1. Mission awareness that considers current and upcoming missions in information valuation and storage requirements;
2. A fuzzy logic function that calculates a partial order of information value based on multiple relative valued attributes; and
3. Information grouping that treats related information collectively.

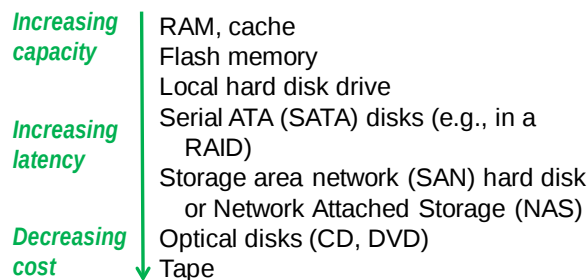


Figure 1. Example levels of hierarchical storage.

2. RELATED WORK

Information Lifecycle Management and Hierarchical Storage Management. Existing ILM solutions take the following forms [3][21]:

- Storage-centric offerings, i.e., multiple storage solutions with different capacity and price characteristics and software and consulting to use it (the point of view of storage vendors);
- Technologies for Hierarchical Storage Management (HSM), i.e., automatically moving information between storage levels (the point of view of some software vendors); or
- Business processes pertaining to the value, retention, and management of information (the point of view of some services companies).

Many current ILM and HSM technologies are variations on backup and retrieval software. They only work on files or documents; move data based on age or time/frequency of access; and are only triggered by storage full situations or by time.

Whereas much focus in ILM centers around the HSM part, most existing HSM offerings are mechanistic in nature, providing information movement and retrieval based on file systems and standardized control interfaces. They are invoked, manually or automatically, when storage space gets tight or based exclusively on time.

It is common in HSM to categorize levels of storage (level 0, level 1, level 2, etc.) based on their relative speed, capacity, and cost, as shown in Figure 1. In general, lower storage levels are considered higher speed, lower latency, more costly, and lower capacity. In operational terms, level 0 is the most accessible to ongoing missions (e.g., onboard storage on tactical platforms), where levels 1 and higher increase in capacity and latency to access information by edge platforms.

In reality, these levels and the media that occupy them are not a total order. For example, the capacity of modern disk drives exceed the capacity of a single optical disk, the cost of tape is not necessarily less than that of optical disks, and the capacity of multiple optical disks and multiple tapes is comparable (virtually unlimited). Furthermore, some of the storage media are not very relevant. Generally, it is not relevant to consider RAM and cache when discussing HSM technology, since they typically fall under the control of the operating system.

Information Management Services. IM services have emerged as necessary concepts for information exchange in net-centric operations [10][19][20], including Net-Centric Enterprise Systems (NCES) [5], which provide a set of services that enable access to and use of the Global Information Grid (GIG) [6]. The core concept of IM is an active information management model where clients are information publishers and consumers communicate anonymously with other clients via shared IM services, such as publication, discovery, brokering, archiving, and querying [4], [8]. Information is published in the form of typed information objects (IOs) consisting of payload and indexable metadata describing the object and its payload. Consumers make requests for future (subscription) or past (query) information using predicates, e.g., via XPath [24] or XQuery [25], over IO types and metadata values.

Common IM services include brokering (i.e., matching future published IOs to subscriptions), archiving of IOs, querying for archived objects, and dissemination of IOs to subscribing and querying clients, as shown in Figure 2. This approach is similar in concept to other publish-subscribe middleware, such as the Java Message Service (JMS) [10], Data Distribution Service (DDS) [19], CORBA Notification Service [20], or Web Services Notification (WS-N) [18].

In contrast to most of the solutions offered today, the military needs ILM solutions that are *mission-driven*, not simply triggered by a lack of available space or time, are *mission-aware*, not simply moving the oldest or least recently used data, and work with a variety of *structured information objects*, not just files or opaque documents.

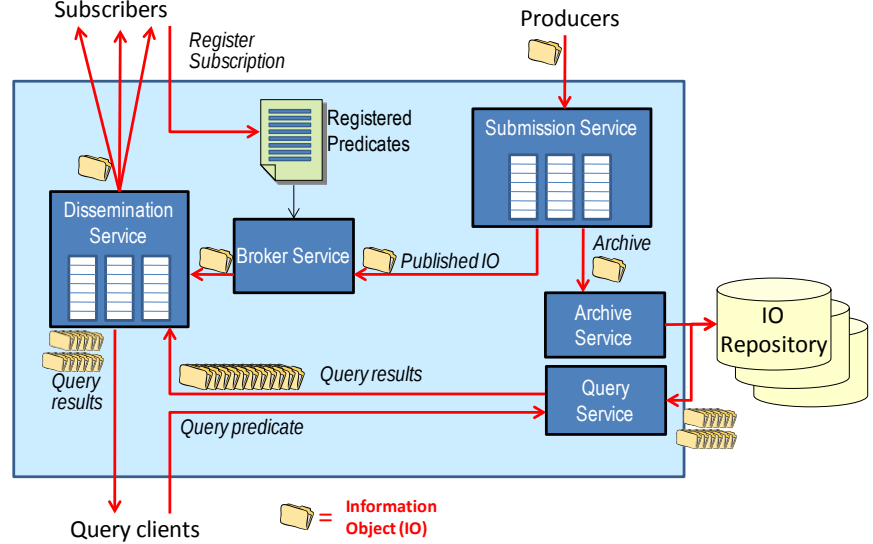


Figure 2. Core Information Management Services

3. A FUZZY LOGIC APPROACH TO INFORMATION VALUATION

One of the key challenges for ILM is determining when an IO's value is sufficiently depreciated to warrant moving it from level 0 store, in favor of another IO to occupy the same space (because its value to ongoing operations is greater) or in favor of maintaining the space free for occupation by future IOs (because the *potential* value of the future IOs to ongoing operations is greater).

Whether information should be moved out of level 0 store comes down to a difficult to quantify predictive measure, i.e., whether the information will be needed soon (or ever). Furthermore, it is a relative assessment, i.e., whether the space in level 0 store occupied by an IO X is best used for X or a different IO, or left available for future information. Furthermore, there can be multiple factors that go into deciding the relative worth of information objects, including the missions or operations that they are being used in (indicating how relevant they are to ongoing operations), the age of the IOs (indicating how fresh the information is), and the size of the IOs (indicating how much space they are using). Each of these factors has relative interpretation. That is, whether an item of information is relevant enough to keep, or large or old enough to move is relative to other items of information and to the anticipated use of the space if the information object is moved. Any discrete or static threshold for any of these factors will lead to inflexibility, i.e., it is likely to only be suitable in specific situations and not sufficient in others, and potential thrashing.

Therefore, we developed a *Value Depreciation Function (VDF)* that builds a *partial order* of information valuation so that at any time when information needs to be moved to make room in level 0, the information that is most *depreciated* in value relative to the others will be moved. We use a *fuzzy logic* rule based approach to produce the partial order from relative valued inputs because of the following:

- Whether an IO has sufficiently depreciated in value to move or not is not completely *true* nor completely *false*. Instead, it is more or less true or false depending on the other choices available, such as how bad the space is needed, what else there is to move, and what the information will be used for.
- The factors that go into determining an item of information's valuation lend themselves to relative interpretation. For example, whether an IO is old depends on the IO's age *relative* to that of other IOs. Whether moving an IO will free up much space depends on the IO's size *relative* to that of other IOs.

Fuzzy sets capture fuzzy, relative valued memberships better than traditional sets [26]. *Fuzzy logic* is a technique for making decisions based on combining the members of multiple fuzzy sets [9]. Traditional sets typically are described using a binary membership function, m , where a set $S = \{x \mid m(x) = 1\}$, i.e., $m(x) = 1$ means that x is a member of the set

and $m(x) = 0$ means that x is not a member of the set. Alternatively, a traditional set S is a pair, i.e., $S = (U, m)$, where U is the universe over which the set S can exist, and the function m determines the membership of any element, $s \in U$, in S . If $m(s) = 1$, then $s \in S$. If $m(s) = 0$, then $s \notin S$. As an example, consider the set of all IOs in a repository. The set BFT can be defined as the traditional set of all IOs that have the type, *BlueForceTrack*. That is, for a repository of IOs, R , $BFT = (R, f(i) = (\text{type}(i) == \text{BlueForceTrack}))$.

In contrast, consider defining all the sets of *large* IOs or *old* IOs. Although the *size* and *age* of each IO is quantitative, the judgment of whether something is *large* or *old* is a relative, fuzzy concept. These are not as well described by traditional sets, because of the traditional sets' binary notion. For example, assume that the *large* set is defined as a traditional set over the universe of all IOs, with a membership function $f(i) = (\text{size}(i) > 100)$. An IO of size 101 would be in the set *large*, as would an IO of size 1000, and an IO of size 1,000,000. All of these IOs would have the same membership in the set *large*, despite the orders of magnitude differences in their sizes. Conversely, an IO of size 99 would not be in the set *large*, despite being much closer to the IO of size 101 than the other elements in the *large* set.

A fuzzy set is defined as a pair $F = (U, m)$ like the traditional set, but the function m in a fuzzy set has a range in the interval $[0, 1]$, as shown in Figure 3. An element, $s \in U$, such that $m(s) = 0$ still means that s is not a member of the set, i.e., $s \notin S$. Any non-zero value for $m(s)$ indicates the *degree* to which s is a member of the set, with $m(s) = 1$ meaning that s is *fully* in the set.

In our size example above, the IOs of size 101, 1000, and 1,000,000 would each have a membership degree > 0 and ≤ 1 , and the membership degree of the IO of size 1,000,000 would be larger than that of the IO of size 1000, which in turn would be larger than that of the IO of size 101. In this way, a fuzzy set, $F = (U, m)$, provides a partial order over its members.

Fuzzy logic is a technique for making decisions based on combining the members of multiple fuzzy sets to produce a degree of membership in an output set, as shown in Figure 4. It consists of the following three steps [22]:

- Acquiring a number of input values.
- Processing the inputs according to a set of fuzzy logic rules.
- Averaging and weighting the outputs of all the individual rules into a single output decision.

Fuzzy logic has been used in various applications. The subway system in Sendai, Japan uses a fuzzy logic controller to control the subway train's acceleration, slowing, and braking to ensure a smoother ride than position based controllers [11]. Fuzzy logic has also been used in air conditioning and heating system controllers [1], rice cookers [23], industrial automation [7], 3D Animation software [16], and elevator controls [11], [17]. The International Electrotechnical Commission (IEC) standardized the *Fuzzy Control Language, FCL*, in IEC 61131-7 in 1997 [12].

We implemented the Value Depreciation Function using *jFuzzyLogic*, an open source Java implementation of Fuzzy Control Logic [13]. VFILM's VDF function uses fuzzy input sets including the relevance to current, future, or past mis-

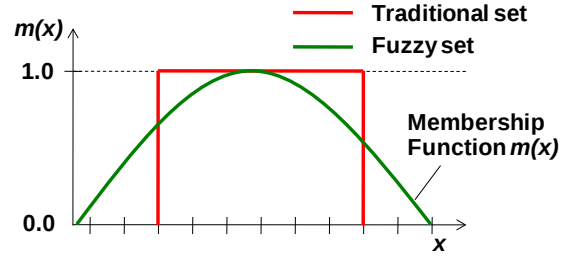


Figure 3. Whereas traditional sets have binary membership functions, fuzzy sets have degrees of membership.

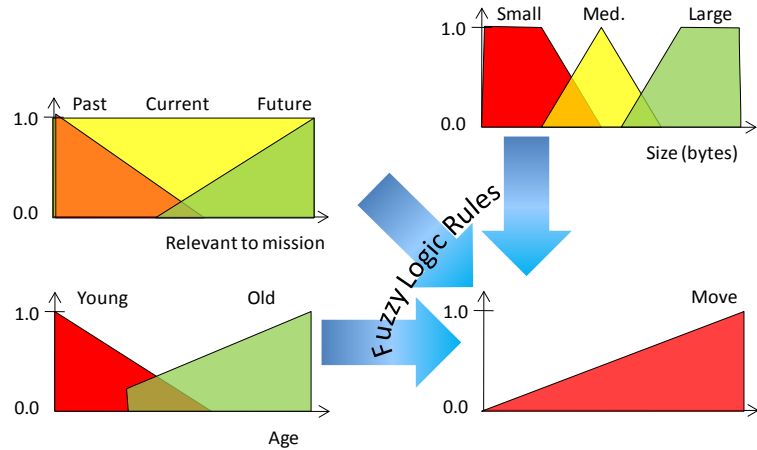


Figure 4. The combination of fuzzy input sets into relative membership in a *Move* set using fuzzy logic rules.

sions, the size of an IO, and the age of an IO, shown in Figure 5 with their jFuzzyLogic descriptions. The *x* axis represents the measured value of the input, e.g., the age of an IO calculated from the current time and a creation timestamp, and the *y* axis represents the degree of membership in a particular fuzzy set.

A set of fuzzy logic rules combines the input values for a particular IO into a degree of membership in a *Move* set. The higher the membership, the more devalued the worth of the information is, and the more likely it is to be moved. Figure 6 shows a set of rules we used in our VFILM prototype. These rules combine the *missionStatus* (i.e., relevance), *age*, and *ioSize* membership values into a membership degree in the *move* output set, with *missionStatus* weighted higher (weight of 1) than *age* (weight of 0.5) and *ioSize* (weight of 0.25).

The *move* set represents the partial order of information (de)valuation, i.e., those IOs with the higher level of membership in *move* are those that should be moved first when space is needed, i.e., they are the least relevant to ongoing missions, the oldest, and will free up the most space (i.e., they are the largest).

This design offers a tremendous amount of flexibility and extensibility in the VDF. New factors can be added to the valuation by introducing a new fuzzy input set. The rules for combining the fuzzy inputs into the output measure can be extended. Finally the various input factors can be weighted so that some factors contribute more to the output set than others.

The VDF component of the ILM consists of the following pieces:

- Fuzzy sets representing the inputs and output of the VDF function.
- Fuzzy logic rules that combine the inputs into a

```

RULE 1 : IF missionStatus IS active
        THEN move IS unlikely

RULE 2 : IF missionStatus IS NOT active
        THEN move IS likely

RULE 3 : IF age IS young
        THEN move IS unlikely WITH 0.5;

RULE 4 : IF age IS NOT young
        THEN move IS likely WITH 0.5;

RULE 5 : IF ioSize IS small
        THEN move IS unlikely WITH 0.25;

RULE 6 : IF ioSize IS large
        THEN move IS likely WITH 0.25;

```

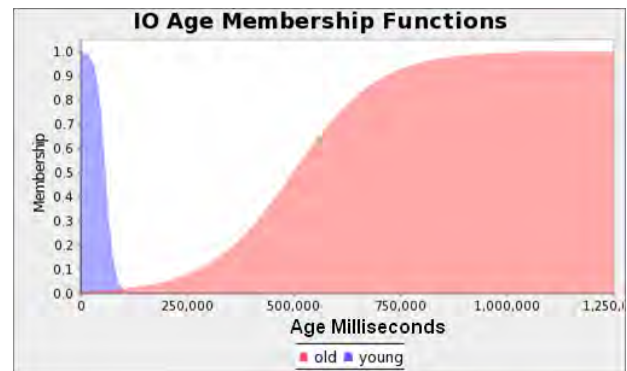
Figure 6. Rules for combining the fuzzy inputs into a *move* valuation.



```

FUZZIFY missionStatus
    TERM active := (0.0,0)(1.0,1);
END_FUZZIFY

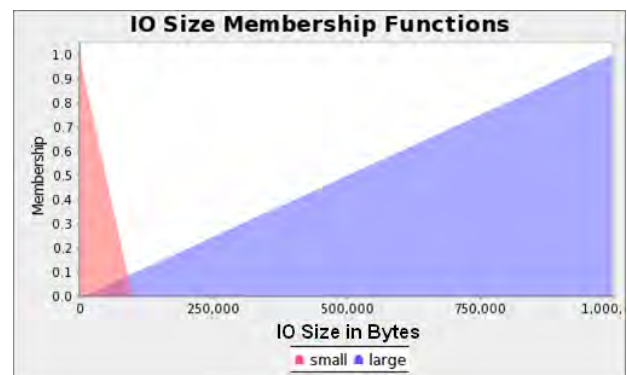
```



```

FUZZIFY age
    TERM young := sigm -0.0001 60000;
    TERM old  := sigm 0.00001 500000;
END_FUZZIFY

```



```

FUZZIFY ioSize
    TERM small := (0,1)(1000000,0);
    TERM large := (10000,0)(1000000,1);
END_FUZZIFY

```

Figure 5. Sample fuzzy input sets for the Value Depreciation Function.

degree of membership in the output set.

- Functions that access the values for the fuzzy inputs, which can be stored in information metadata, Phoenix Context objects, system condition monitors, operating system attributes, etc.

4. THE VFILM INFORMATION LIFECYCLE MANAGEMENT SERVICE

We incorporated the information valuation function, VDF, into an ILM service that provides the following advantages over existing ILM and HSM approaches:

- Multiple, non-traditional factors can be used in the VDF to value information, including but not limited to mission factors, relation to other information, and information characteristics, and the set of factors is extensible.
- The valuation and movement of information can be triggered by mission, system, policy, and other relevant events through an easily extendable event-handler implementation.
- The information valuation and movement functions are separated, so that either can be scheduled when needed or when resources are available, and can be executed as much as needed, e.g., to recover just enough space to continue.
- It avoids thrashing around fixed storage thresholds or drastic purges of information to free up storage. VFILM treats information valuation as a partial order of the “criticality” of information, so as much or little can be moved as needed.
- It can treat groups of information that are related collectively, so that they are valued and moved as a group, when appropriate.
- It provides a rich framework for specifying factors and policies for valuing and moving information. New FCL rules, fuzzy sets, policies, groups, and thresholds are readily added or changed.

The VFILM prototype ILM service is shown in Figure 7 and consists of the following components:

- The *ILM Event Manager* manages event handlers that handle events that can trigger information valuation and movement.
- The *ILM Controller* drives the behavior of the ILM in response to events.
- The *Value Depreciation Function* implements the valuation algorithm described in Section 3.
- The *Group Manager* maintains the definitions of groups of information.
- The *ILM-HSM Adapter* abstracts away the specifics of the HSM and Repositories being used. In the current prototype, the ILM-HSM Adapter implements representative HSM functionality.

4.1 The ILM Event Manager and Controller

As shown in Figure 8, the ILM Event Manager maintains a set of Event Handlers, each of which receives incoming higher level events and maps them to ILM events. The ILM Controller receives ILM Events from the Event Handlers and invokes valuation, movement, or update functions.

Generated events and discrete epochs, such as mission events and policy events, are delivered using an event channel. The consumer of the events is the Event Manager that selects the appropriate event handler to use for each event, based on the event type. Continuous conditions, such as the amount of free storage, can be monitored directly by event handlers.

Each event handler maps the incoming or monitored higher-level events to a set of ILM events, and the set of ILM events is passed to the ILM controller for execution. The ILM events serve as the “language” of the ILM and *trigger* the ILM to conduct information valuation, information movement,

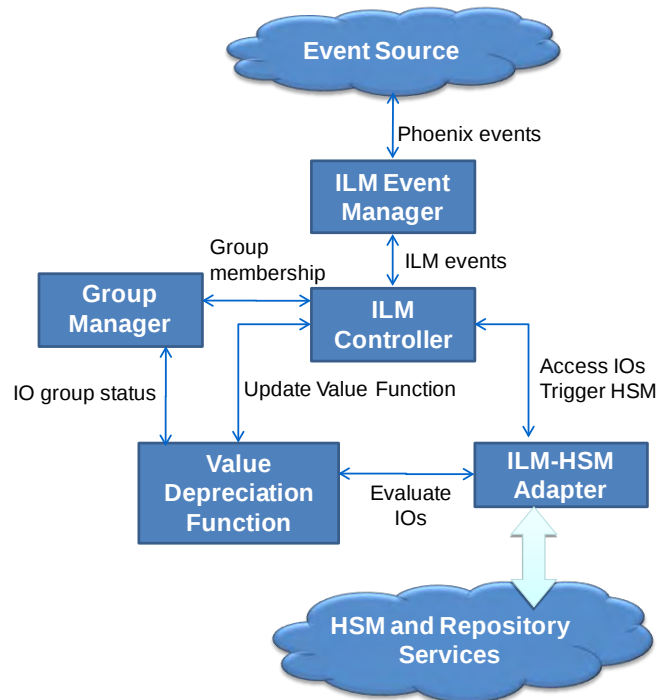


Figure 7. Design of the ILM Service.

group updates, and/or policy modification. We prototyped the following set of ILM events:

- *NeedSpace* – Indicates that a particular amount of space should be made available by moving information.
- *Cleanup* – Check the relative valuation of information across the hierarchical levels and *re-balance* the location of information, so that the most critical information (lowest depreciation valuation) is in level 0 store and less critical information (highest depreciation valuation) is in higher storage levels.
- *UpdateThreshold* – Change the threshold of available space that the ILM should maintain in level 0 storage.
- *Valuation* – Execute the VDF valuation function on a set of IOs (provided as a parameter).
- *GroupUpdate* – Create a new group or change the attributes of a group of IOs.
- *RuleChange* – Add or change a fuzzy logic rule determining the valuation of IOs.
- *MoveIOs* – Move a set of IOs from one repository to another (usually in different storage levels).

Event Handlers are pluggable and the VFILM prototype provides the following set:

- *Mission Event Handler* – Reacts to incoming Mission Events.
- *File System Monitor* – Monitors the level of free space and triggers a *Need Space* event when the available space drops below a specified threshold.
- *Group Policy Handler* – Listens for events associated with policy governing groups of information.
- *Admin Policy Handler* – Provides ILM administration, such as setting the free space threshold or inserting new Event Handlers.

We defined the following Mission Event Types for the VFILM prototype:

- *MissionPrep* – Indicating that a planned mission will start sometime in the future and the ILM should prepare for it.
- *MissionBegin* – Indicating the start of a mission.
- *MissionEnd* – Indicating the end of a mission.

The Mission Event Handler maps these three Mission Event Types to the ILM Events and operations indicated in Table 1.

Table 1. Mission Events and mapping to ILM Events representing the prototype VFILM Mission Domain Model.

Mission Event Type	ILM Events	Resulting ILM operations invoked by the ILM Controller
MissionPrep	Cleanup	Runs the valuation function on multiple storage levels and sorts the results so that the IOs are balanced across the storage levels according to their valuation and the storage thresholds.
MissionBegin	GroupUpdate	Creates a group representing the mission.
	Valuation	Triggers valuation of all IOs matching the mission predicate.
	NeedSpace	Moves IOs to free up enough available space for the mission.
MissionEnd	GroupUpdate	Removes the group associated with the mission.
	Valuation	Triggers valuation of all IOs associated with the mission, i.e., matching the mission predicate.

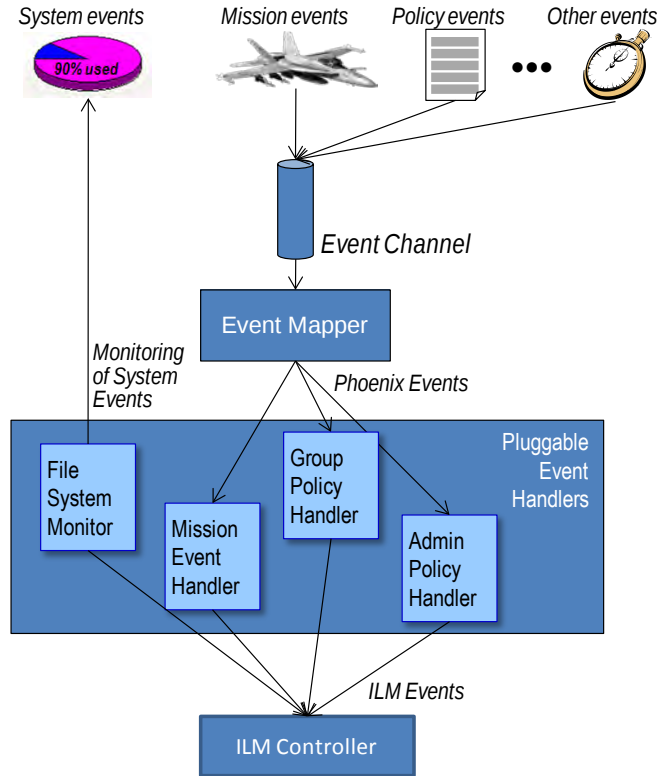


Figure 8. Design of the ILM Event Manager.

4.2 The Group Manager

In many cases, IOs are not independent entities and there are significant advantages to having the ILM exploit the interdependencies. One realization of information interdependencies is association with a common group. As shown in Figure 9, information in a system can be associated with many overlapping and co-existing groups, based on shared types (e.g., *blue force tracks*, *BFTs*), source (e.g., a specific platform), role (e.g., intelligence, surveillance, and reconnaissance, *ISR*, for mission A), epoch (e.g., a sortie), location within a particular region, and so forth. VFILM supports the association of IOs that are related and that should be treated collectively into groups. Events can affect a group of IOs and IOs can be collectively valued and moved.

Groups are defined using predicates over observable attributes of IOs. The *Group Manager* maintains a collection of *Group Contexts*, which hold the following information about a group, as shown in Figure 10:

- *Identifier* – A name to identify the group.
- *Predicate* – A predicate defining the IOs in the group. The predicate is defined over observable attributes of an IO.
- *Valuation rules* – The VDF rules used for IO valuation in the group.
- *Precedence* – Used to determine which group definition is used during IO valuation when an IO is part of multiple groups.
- *Stored values* – Input values associated with a group and used during IO valuation.

Missions are represented as just another type of group. When a Mission Start event occurs, a Group Context is created for IOs associated with the mission.

4.3 The ILM-HSM Adapter

The ILM-HSM Adapter is a control interface from the ILM to HSM functionality that is intended to support a variety of HSM implementation options. As such, it provides a consistent interface for the ILM to specify IOs and files that should be moved independently of the specific HSM or repository that is used. In a situation where a full HSM solution is not appropriate, the ILM-HSM Adapter can be responsible for the movement of information. If the repository stores IOs as files on disk this could involve moving the file and updating the repository's reference or leaving a symbolic link to the file's new location. With other repository implementations where IOs are stored in relational databases, this could involve removing the IO from one table and inserting it into another.

In situations where an HSM is used, the ILM-HSM adapter serves as the interface to the HSM. This depends on the specifics of the HSM utilized but could involve assigning priority values to managed files based upon information value or modifying a management policy.

In the VFILM prototype, we simulated HSM functionality within the ILM-HSM Adapter. We manage a repository where metadata is stored in a Berkeley XML DB [2] and full IOs are stored as individual files on disk, an example layout is shown in Figure 11. The ILM-HSM can handle multiple repositories, can move just IOs retaining metadata in the level 0 store, or move metadata and IOs to level 1 store.

The ILM-HSM Adapter maintains the following two extra databases on each level of storage to facilitate information movement:

- A *Value Store* – Contains the IO context ID, IO value (i.e., the result of the most recent valuation execution), and the storage level of the IO.
- An *ILM Index* – Maintains an index of the IOs in the level of storage indexed by the fields used to define group membership allowing rapid lookup of IOs associated with a group.

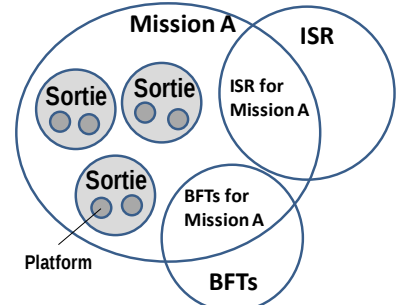


Figure 9. Information can be organized into many groupings, some of which have associated lifecycles.

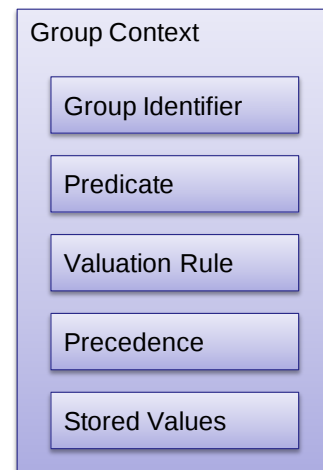


Figure 10. The Group Context contains the information needed to represent a group of IOs.

The ILM-HSM Adapter provides multiple options for moving IOs. The first option moves the payload of IOs only, retaining indexable metadata in level 0 store. This enables all queries to be performed on the repositories in level 0, although retrieval of results might need to reach into higher levels. Because retaining all of the metadata in level 0 store can still lead to the level 0 store filling up, the second option is to move metadata and IOs together. When the metadata is moved out of level 0, the query service must be aware of the hierarchical storage to conduct queries on the repositories in other levels. The movement of IOs only and the movement of metadata and IOs together can coexist.

5. THE ILM PROTOTYPE PERFORMANCE

Figure 12 shows a screenshot of the VFILM prototype in action. It shows the IOs in the level 0 and level 1 storage, with the x axis showing the IO valuation result (higher values, i.e., IOs farther to the right, represent information that is more depreciated in value). The y axis indicates the number of IOs at each valuation level. The red IOs are in a repository on level 0 and the blue IOs are in a repository on level 1. In this screenshot, the ILM has perfectly balanced the IOs, with all the IOs in level 1 storage having a higher depreciation value than those in level 0 storage.

Figure 13 shows the VFILM prototype's ability to maintain a threshold of available space in level 0. The x axis indicates the passage of time during a run of the prototype. The red line indicates the amount of available space (y axis) at any point in time. When the available space reaches the lower threshold, it triggers a move event until the available space reaches the *Stop Move Threshold*. The vertical lines indicate events, such as mission events, that affect information valuation or movement.

The VDF scales linearly with the number of IOs being processed. Figure 14 shows the result of an experiment in which we published a pre-defined number of IOs and then triggered evaluation over the entire set. We ran the experiment with two different IO sizes (100 KB and 1 MB) in five different repository configurations, containing 1MB, 25MB, 50MB, 75MB, and 100MB total. The results are shown in Figure 14 with a linear best fit line for each and the fit line's coefficient of determination (R^2).¹

For the smaller IO size (100 KB), there is a close linear fit and the variance is low for all set sizes. For the larger IO size (1 MB), the scaling still appears to be linear, but the variance is much higher. Figure 14 also shows that the VDF does not scale with the total size (in MB) of the evaluation set. With the two series plotted together, it is apparent that scaling is dependent upon the number of IOs

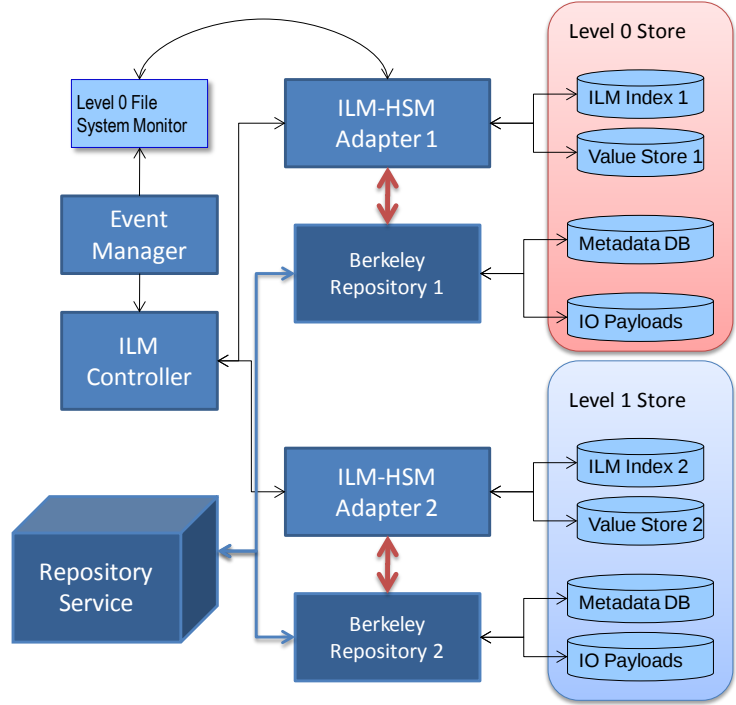


Figure 11. Design of the ILM-HSM Adapter.

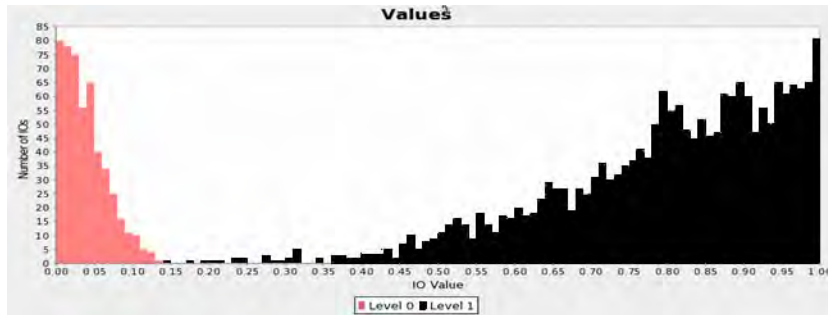


Figure 12. IO valuation histogram

¹ The coefficient of determination is a measure of the proportion of variability in a data set that is accounted for by the fit line, with 1 being the best possible.

rather than the data size of the evaluation set. With equivalent data sizes, large IOs are evaluated much more quickly because the VDF's evaluation function is called fewer times.

The time that the HSM takes to move IOs depends on both the number and the size of IOs being moved. We tested this with two different IO sizes (100 KB and 1 MB) and five different repository sizes (containing 10, 250, 500, 750, and 1000 IOs). The results for both IO sizes appear in Figure 15. The coefficient of determination indicates a strong linear correlation in both cases. Figure 15 also clearly shows that the two IO sizes do not scale according to the same linear function. It takes longer to move an equivalent number of large IOs because the HSM executes the same number of move operations, but each operation requires more data to be moved on the file system. Given the same amount of data to move, large IOs are moved more quickly because the HSM needs to execute fewer move operations in order to free the same amount of space. For a given number of IOs, it takes longer to move larger IOs due to reading and writing the bytes. For m IOs each of size s , the time to execute the HSM is $O(m*s)$

6. CONCLUDING REMARKS

In this paper, we have presented the design and prototype that provides a foundation for an ILM capability in enterprise and tactical environments. The VFILM prototype includes triggering of information lifecycle management based on mission events and mission-based policy, valuation of information using fuzzy logic algorithms based on the information's urgency to ongoing mission operations, grouping of information based on common attributes and dependencies, and migration and retrieval of information objects and groups.

The major contributions of this paper include:

- A prototype ILM service that provides mission-aware information valuation, control of HSM movement of information between levels of storage, and support for IM services and repositories.
- An ILM-HSM interface that abstracts the details of specific HSMs, file systems, and repositories.
- A novel approach to information valuation, supporting an extensible multi-factor assessment of the relative values of information using fuzzy logic. The approach produces a partial order of information depreciation, handles dynamic conditions that can change the worth of information, and avoids the thrashing that is possible with fixed or static valuation thresholds.
- A set of experimental results showing the performance of the ILM service.

Further research building upon this foundation can explore additional richness in the VFILM prototype, e.g., to expand its mission models and the factors utilized in valuation; expanding the query service to be more aware of the hierarchical storage levels and to exploit this knowledge to order query responses; and to explore distributing and coordinating ILMs for improved storage and access to critical information.

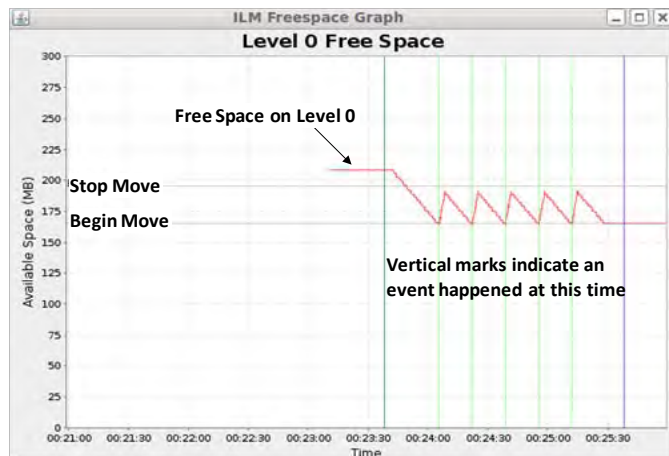


Figure 13. VFILM free space graph

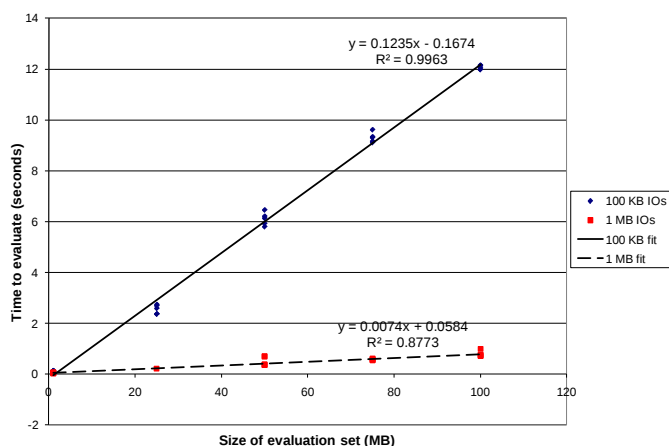


Figure 14. The VDF scales linearly with the number of IOs, but not with the size of the evaluation set.

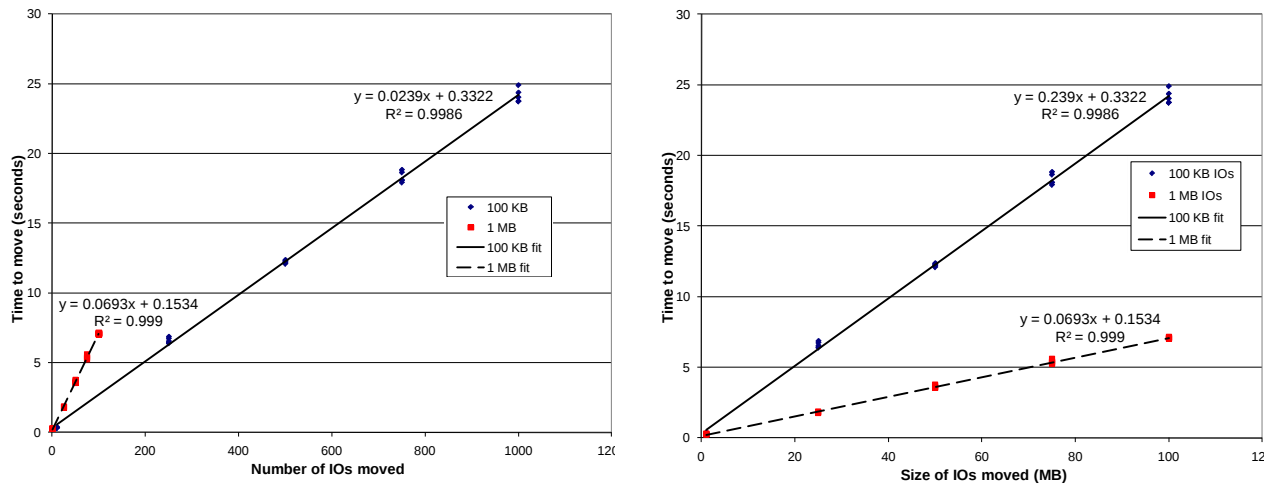


Figure 15. The time that the HSM takes to move IOs depends on both the number and size of IOs.

REFERENCES

- [1] Von Altrock, C., "Fuzzy logic and neurofuzzy technologies in appliances," Proc. Embedded Systems Conference, (1996).
- [2] <http://www.oracle.com/technetwork/database/berkeleydb/overview/index.html>.
- [3] Chudnow, C. T., "Information lifecycle management and the government," *Computer Technology Review*, (2004).
- [4] Combs, V., Hillman, R., Muccio, M., and McKeel, R., "Joint battlespace infosphere: information management within a C2 enterprise," Proc. Tenth Int. Command and Control Technology Symposium, (2005).
- [5] Defense Information Systems Agency, Net-centric enterprise services, <http://www.disa.mil/nces/>.
- [6] DoD CIO, "Department of Defense Global Information Grid architectural vision, vision for a net-centric, service-oriented DoD enterprise, version 1.0," (2007). <http://www.defenselink.mil/cio-nii/docs/GIGArchVision.pdf>.
- [7] Gebhardt, J. and von Altrock, C., "Recent successful fuzzy logic applications in industrial automation," Proc. Fifth IEEE International Conference on Fuzzy Systems, (1996).
- [8] Grant, R., Combs, V., Hanna, J., Lipa, B., and Reilly, J., "Phoenix: SOA based information management services," Proc. SPIE Defense Transformation and Net-Centric Systems Conference, (2009).
- [9] Hajek, P., "Fuzzy logic", [The Stanford Encyclopedia of Philosophy (Fall 2010 Edition)], Edward N. Zalta (ed.), <http://plato.stanford.edu/archives/fall2010/entries/logic-fuzzy/>, (2010).
- [10] Hapner, M., Burridge, R., Sharma, R., Fialli, J., and Stout, K., "Java Message Service Version 1.1," Sun Microsystems, <http://java.sun.com/products/jms/>, (2002).
- [11] Ho, M., and Robertson, B., "Elevator group supervisory control using fuzzy logic," Canadian Conference on Electrical and Computer Engineering, Halifax, NS, Canada, (1994).
- [12] International Electrotechnical Commission (IEC), "IEC 1131 – programmable controllers, part 7 – fuzzy control programming," (1997).
- [13] jFuzzyLogic, <http://jfuzzylogic.sourceforge.net>.
- [14] Kosko, B., and Isaka, S., "Fuzzy logic," *Scientific American*, 76-81 (1993).
- [15] Kugler, R., Baranick, M., and Binnendijk, H., "Operation Anaconda lessons for joint operations," Center for Technology and National Security Policy, National Defense University (2009).
- [16] MASSIVE 4.0, Supporting V-Ray Rendering, is Now Available. *Business Wire*, February 18 (2010), <http://www.thefreelibrary.com/MASSIVE+4.0,+Supporting+V-Ray+Rendering,+is+Now+Available-a0219126342>.
- [17] Mitsubishi Electric. AI Supervisory Control System. http://www.mitsubishi-elevator.com/innovations/control_system.html.
- [18] OASIS. Web Services-Notification. www.oasis-open.org/committees/wsn.
- [19] Object Management Group. Data Distribution Service for Real-time Systems, Version 1.2. OMG Specification, formal/07-01-01, <http://www.omg.org/cgi-bin/doc?formal/07-01-01> (2007).

- [20] Object Management Group. Notification Service, Version 1.1. OMG Specification, formal/2004-10-11, http://www.omg.org/technology/documents/formal/notification_service.htm (2004).
- [21] Pariseau, B., "Does EMC's ILM strategy really solve complexity?" Storage Technology News, (2005).
- [22] Sowell, T., "Fuzzy logic for 'just plain folks'," <http://www.fuzzy-logic.com/>.
- [23] Toothman, J., "How rice cookers work," <http://home.howstuffworks.com/rice-cooker2.htm>.
- [24] World Wide Web Consortium, "XML path language (XPath) version 1.0," W3C Recommendation, <http://www.w3.org/TR/xpath> (1999).
- [25] World Wide Web Consortium, "XQuery 1.0: an XML query language," W3C Recommendation, <http://www.w3.org/TR/xquery> (2007).
- [26] Zadeh, L.A., "Fuzzy sets," *Information and Control*, 8, 338-353 (1965).