Future Internet **2010**, *2*, 156-189; doi:10.3390/fi2020156

OPEN ACCESS

future internet

ISSN 1999-5903

www.mdpi.com/journal/futureinternet

Article

A Distributed Infrastructure for Metadata about Metadata: The HDMM Architectural Style and PORTAL-DOORS System

Carl Taswell

Global TeleGenetics, Inc., 8 Gilly Flower Street, Ladera Ranch, CA 92694, USA;
E-Mail: ctaswell@computer.org; Tel: 1-949-481-3121

Received: 30 December 2009; in revised form: 12 April 2010 / Accepted: 20 May 2010 / Published: 1 June 2010

Abstract: Both the IRIS-DNS System and the PORTAL-DOORS System share a common architectural style for pervasive metadata networks that operate as distributed metadata management systems with hierarchical authorities for entity registering and attribute publishing. Hierarchical control of metadata redistribution throughout the registry-directory networks constitutes an essential characteristic of this architectural style called Hierarchically Distributed Mobile Metadata (HDMM) with its focus on moving the metadata for *who what where* as fast as possible from servers in response to requests from clients. The novel concept of multilevel *metadata about metadata* has also been defined for the PORTAL-DOORS System with the use of entity, record, infoset, representation and message metadata. Other new features implemented include the use of aliases, priorities and metaresources.

Keywords: architectural style; mobile metadata; hierarchical authority; distributed registry-directory system; IRIS-DNS System; PORTAL-DOORS System; PDS; HDMM

1. Introduction

A distributed registry-directory system as a cyberinfrastructure for *who-what-where* metadata management in support of navigation, search, and queries of the semantic web has been designed [1,2] using architectural principles inspired by the examples of the corresponding systems previously established for the original web. The Problem Oriented Registry of Tags And Labels (PORTAL) and the Domain Ontology Oriented Resource System (DOORS) for the semantic web have been devised to function as interacting systems of registries and directories for the semantic web in a manner analogous to those used for the original web, namely, the Internet Registry Information Service (IRIS) and Domain Name System (DNS). The PORTAL-DOORS System (PDS), collectively comprised of these interacting network systems of PORTAL registries and DOORS directories, has been architected as a distributed system for registering resource entities and publishing metadata about them. As a lower-level infrastructure system distinguished from higher-level tools and applications built upon the foundation of the infrastructure, PDS establishes an interoperable, platform-independent, application-independent, messaging interface standard for information exchange over the internet. Moreover, PDS has been purposefully designed to address a variety of major issues and problems including provenance, security, cybersilos, transition barriers, search engine consolidation and the spread of misinformation.

In a comprehensive literature review, Taswell [1] discussed the cybersilo problem and other barriers impeding the transition from original web to semantic web. The cybersilo problem has also been expressed as the data integration challenge of coping with an *information tsunami* in the presence of an *informatics Tower of Babel* [3] resulting from non-interoperable systems (*i.e.*, those that cannot effectively communicate with each other). Thus, traditional silos in scholarly scientific discourse have been perpetuated as cybersilos. Transition barriers refer to the complex array of issues that slow the adoption of the new semantic web while continuing use of the original non-semantic web. In a broad sense with reference to general concepts rather than specific technologies, the semantic web can be distinguished from the original web by the presence of sufficient metadata for hyperlinks to enable their navigation by an automated processing agent. This associated metadata may be either explicit markup text and/or implicit inferred context but the automated processing agent must be able to understand the meaning of the metadata (see also [4] for other perspectives on the semantic web, and Figure 2 in this report for the definitions of lexical versus semantic services as used in the context of PDS).

More recently, Mowshowitz and Kumar [5] published a commentary with growing concerns about increasing control of web search by a decreasing number of search engines essentially limited to an effective monopoly (currently Google has dominating market share) or oligopoly (currently Google, Yahoo, and Microsoft comprise the top three). To quote Mowshowitz and Kumar [5], this search engine consolidation "raises the specter of biased information and free speech abridgement" that implicitly results when information is only accessable and/or accessed through the filter of a search engine. In addition, Berti-Equille *et al.* [6] have discussed the problems of conflicting data from multiple data sources, the spread of false information, and the discovery of source dependency. Acemoglu *et al.* [7] provided a theoretical analysis of these concerns, especially, the spread of misinformation.

Guided by architectural principles intended to address all of these major problems, PDS has been designed to operate as a hybrid between the original web and the semantic web with mechanisms that serve both the original web and the semantic web simultaneously as well as either one independently of the other. PDS has also been conceived to operate as a bridge between the original web and the semantic web by enabling use of the system in diverse scenarios. These scenarios range from the simplest case with the minimal set of features required by the original web to the most sophisticated case with the maximal set of features permitted by the semantic web (see Section 6.3). Thus, PDS directly addresses the problem of transition barriers. Moreover, PDS has been devised to bootstrap itself in a manner in which both the infrastructure system and its content are distributed physically and virtually in terms of both the content and control of content. If fully implemented and operated as an internet-scale pervasively distributed infrastructure, this distributed design of the PORTAL-DOORS System itself would prevent the possibility of search engine consolidation in a manner entirely analogous to the success of the IRIS-DNS System in preventing the consolidation of internet domain name registries or directories.

However, a system's *descriptive architecture* as realized when implemented may deviate from the system's *prescriptive architecture* as intended when designed [8, page 59]. This discrepancy results from a process called *architectural degradation* which comprises the related phenomena of *architectural drift* and *architectural erosion* [8, page 61]. These phenomena are distinguished by the discrepancies that arise in the degradation process. Thus, in the process of erosion, the fundamental design decisions of the prescriptive architecture are violated in the descriptive architecture, whereas in the process of drift, these decisions are not violated [8, page 64]. For any system that envisions an internet-scale pervasively distributed infrastructure, the tasks of examining, documenting and reporting the descriptive architecture at various stages of implementation and deployment in comparison with the prescriptive architecture should remain a critical step in the overall process of iterative software development. These analyses enable asking important questions about the presence of any architectural degradation. More importantly, they enable a reasoned deliberation about whether the design decisions of the original prescriptive architecture should be maintained or whether any innovative deviations in the new descriptive architecture should be pursued.

Beyond the original published design [1] that serves as the abstract blueprint for PDS (*i.e.*, the prescriptive architecture), some concrete interface schemas together with basic ontologies have been drafted for prototype registries in fields relevant to biomedical computing and radiological informatics. These draft prototypes include a formal semantic definition of pharmacogenomic molecular imaging which provides a use case that demonstrates search across multiple specialty domains [9]. However, such XML-based models represent only a piece of the puzzle. A full reference implementation (*i.e.*, one that results in a more complete descriptive architecture) requires many other software components, especially back-end database servers and front-end clients for the PORTAL registries and DOORS directories.

In order to gain practical experience testing development of alternative implementations, it is necessary to begin working with real data stored in actual database servers. Therefore, the roadmap for PDS development shifted from the revision of XML schemas and OWL ontologies to the construction of a prototype relational database model. This database model has been purposefully chosen to be initially a traditional relational model rather than an RDF-based triple store. This decision was made not only because of the guiding principle that PDS must be capable of operating as a hybrid and a bridge but also because of the pervasive availability of relational databases in comparison with newer kinds of databases. Recent research [10] on the use of relational database models for semantic systems also suggests that relational databases may continue to play an important role rather than being completely displaced by RDF-based triple stores for semantic systems.

This report describes the current status of the descriptive architecture for PDS with respect to the relational database models, web service interfaces, and interoperable messaging schemas implemented for the latest enhancements of the prescriptive architecture for PDS. At present, the PORTAL-DOORS Project maintains an approach of prohibiting architectural erosion while permitting architectural drift that remains consistent with the philosophy, principles, and fundamental design decisions of the original prescriptive architecture [1]. PDS infrastructure schemas now permit an alternative bootstrapping design with integrated NEXUS servers [11] in addition to the original design with separate PORTAL and DOORS servers. The overall design has been further enhanced by the revisions introduced here for the new concept of *metadata about metadata* as defined in Section 5.2.

This new concept of metadata about metadata differs from the variety of diverse kinds of multilevel and/or hierarchical metadata that has been published in prior related work as reviewed in Section 2. Further distinctions between other metadata management systems and the new approach pursued by PDS is discussed in Section 3. Following this introduction and background, there are three more formal sections that present the architectural style called Hierarchically Distributed Mobile Metadata (HDMM) in Section 4, the architectural design of PDS in Section 5, and implementation and application of PDS in Section 6. After completing the presentation of the prescriptive and descriptive architectures for PDS in Sections 4–6, then Section 7 discusses important current questions and planned future work to answer those questions with experiments, while Section 8 concludes with a summary of the important themes.

2. Related Work

2.1. Metadata Management Systems

Prior work on metadata management systems includes diverse examples from the peer reviewed literature [12–29], patent literature [30–33], and published technical reports [34,35]. A doctoral dissertation written in 1998 by Dolin [36] remains especially noteworthy for its discussion of different kinds of hierarchical metadata for multilevel information systems. Dolin built a distributed system called Pharos [36] for locating heterogeneous information sources. Pharos employed concept-based classification trees with three different kinds of information hierarchies: topical, geographical and temporal. However, not all multilevel analyses are restricted to the task of locating resources [37]. Moreover, not all multilevel models are required to be hierarchical [38]. Thus, any review of metadata management systems must compare and contrast the different definitions and interpretations of the terms *hierarchical* and *multilevel* as used or implied by the corresponding features of the various metadata management systems.

These distinctions can best be revealed by posing questions. Does the multilevel or hierarchical aspect refer to the entity itself (*i.e.*, the data, information, source, or resource) or to the metadata about the entity? Regardless of whether the target object is the entity or the metadata, does the multilevel or hierarchical aspect refer to the identification, location, or both identification and location of the target object? Is the fundamental purpose of the metadata management system to classify objects, store and retrieve them, search and discover them, or some other purpose? How similar or dissimilar are the objects with respect to the criteria deemed most relevant and significant to the metadata management system?

As examples, metadata management systems for storage and retrieval of files in computer file systems [17,23,33] operate under a very different set of requirements and constraints when compared with metadata management systems for discovery of services or resources on the web [15,34,35]. Moreover, even the most successful and well-known metadata management system for the web employs at least two different kinds of multilevel or hierarchical features. In particular, the IRIS-DNS System maintains one hierarchical scheme for the identification of resources, and simultaneously, another independent hierarchical scheme for the distribution of the metadata records about the resources. For identification of resources, IRIS-DNS uses a hierarchical naming scheme with top-level domain names, domain names, sub-domain names, etc. For distribution of metadata records, IRIS-DNS uses a hierarchical request-forwarding and response-caching scheme with root servers, primary servers, secondary servers, etc.

As evident from this brief review and discussion, there are many kinds of multilevel hierarchical metadata management systems. However, in all cases found so far in the published literature, the definition and interpretation of multilevel or hierarchical can be considered to be a form of resource identification, resource location, resource content classification, or metadata record distribution. Nevertheless, in the course of work on the software development necessary to move from a prescriptive architecture to a descriptive architecture for PDS, a new and different form of multilevel metadata organization and management has been introduced which offers important practical advantages as explained in Section 5.2.

2.2. *Neuroinformatics Systems*

Recalling the biomedical informatics context which served as the original motivation for PDS [1] as well as the brain imaging informatics context which serves as the application context for the initial prototype registries in PDS [39–41], other related work in neuroinformatics will be reviewed briefly here. The book *Neuroinformatics* edited by Crasto [42] provides a compendium of informatics for brain science and medicine covering neuroscience knowledge management, computational neuronal modeling, brain imaging, and applications in neurogenetics for neurodegenerative disorders. Progress has been made by privately funded projects such as the www.brain-map.org portal for brain gene expression activity mapping of the Allen Institute for Brain Science as well as by publicly funded projects such as the www.neuinfo.org portal for the NIH Blueprint for Neuroscience Information Framework.

Despite continuing efforts (not only in neuroinformatics [43] but also in many other fields) to address the existing barriers to interoperability for current data stores and to begin the transition to a semantic web of meaningfully linked and integrated data [44], the challenging task of reaching the goal of truly interoperable data has only just begun and many hurdles remain in the way. In fact, current neuroinformatics portals still remain essentially isolated from other portals because there is no uniform standardized shared interface for all of the portals in neuroinformatics and other related biomedical sciences to communicate with each other. Thus, it remains necessary to interact individually and separately with each portal's custom interface either directly or else indirectly via customized mediators.

Moreover, general initiatives (*i.e.*, not specific to neuroinformatics) for cataloguing or connecting resources such as www.sitemaps.org, biositemaps.ncbcs.org and linkeddata.org represent short-term partial fixes that may help temporarily with some of the problems but will not suffice as a long-term solution for all of the problems. The main benefit of these approaches derives from their simplicity which enables their rapid deployment without any need for the iterative software development cycles of a more sophisticated infrastructure system such as PDS. However, the same simplicity that results in short-term advantages also results in long-term disadvantages for these approaches because they lack mechanisms to address all of the problems and satisfy all of the requirements for which PDS has been designed. For example, without a registry scheme of some kind, the Biositemaps initiative cannot assure globally unique identification of resources. And without a sufficient metadata framework of a more sophisticated kind, the Linked Data initiative cannot address versioning, provenance, distribution, security and other matters (see Section 3 for further discussion).

Finally, *none* of the current neuroinformatics portals or applications [43] have yet been designed to focus on the use of pharmacogenomic molecular imaging for clinical trials [9,40]. Enabling pursuit of these medical scientific investigations remains a primary driver for developing a knowledge engineering workbench as a software application [41] built upon the PDS infrastructure context involving several of the initial prototype registries including ManRay, BrainWatch and GeneScene.

3. A New Approach

The author has pursued a new approach distinguished by its goal of building a distributed shared infrastructure rather than a single centralized repository, independent site maps or simple peer-to-peer connections. The design for the infrastructure core, called the PORTAL-DOORS System for the semantic web [1], was modeled on the enormously successful design of the IRIS-DNS System for the original web. More specifically, the Internet Registry Information Service (IRIS) registers domain names while the Domain Name System (DNS) publishes domain addresses with mapping of names to addresses for the original web. Analogously, the Problem Oriented Registry of Tags And Labels (PORTAL) registers resource labels and tags while the Domain Ontology Oriented Resource System (DOORS) publishes resource locations and descriptions with mapping of labels to locations for the semantic web.

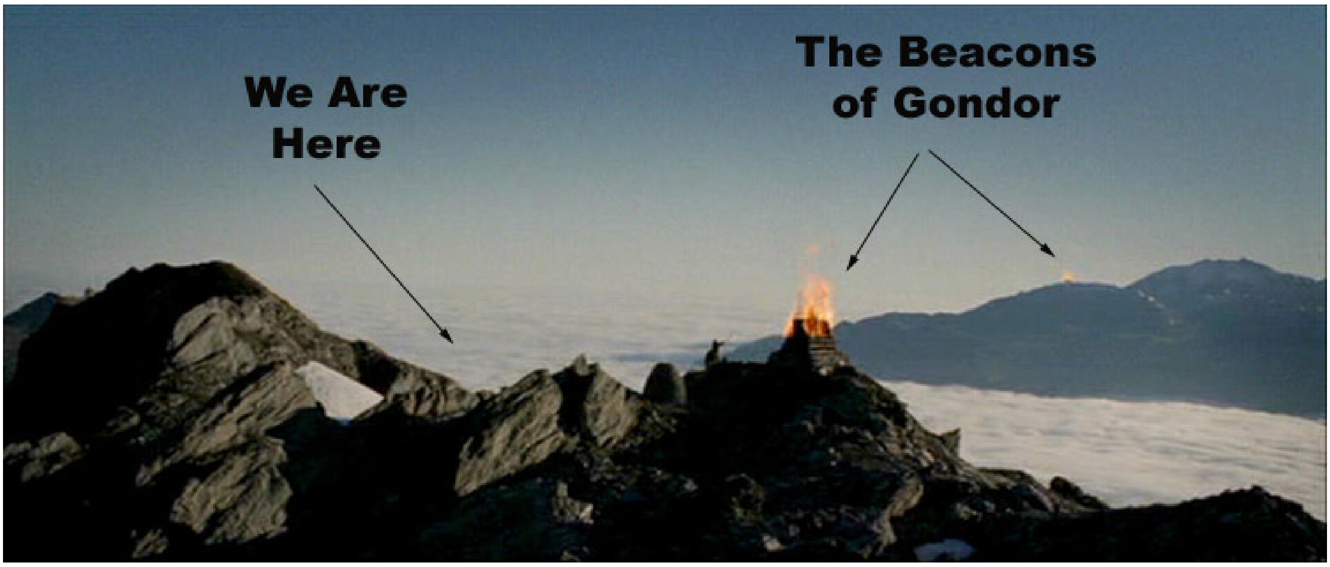
Both the IRIS-DNS System and the PORTAL-DOORS System share a common architectural style for pervasive metadata networks that operate as distributed metadata management systems with hierarchical authorities for entity registering and attribute publishing. Hierarchical control of metadata redistribution throughout the registry-directory networks constitutes an essential characteristic of this architectural style called Hierarchically Distributed Mobile Metadata (HDMM) with its focus on moving the metadata for *who what where* as fast as possible from servers in response to requests from clients [45].

PORTAL-DOORS and IRIS-DNS each operate as information-seeking support systems that function as hierarchical registry-directory networks for the distribution of mobile metadata. While the original motivation for the design of PORTAL-DOORS has been and remains that of serving the goals of neuroinformatics for the study of gene-brain-behavior relationships, PORTAL-DOORS was also designed [1,11] to solve several major problems of web engineering: cybersilos in scientific discourse, search engine consolidation, registry/repository centralization, the spread of misinformation, and barriers to progress in the transition from original web to semantic web (see Section 1 for further discussion).

For the original internet protocols and systems including IRIS-DNS, the architectures were configured purposefully to promote failsafe redundancy and high-speed efficiency for distributed communication networks. These systems were designed with full awareness that significant risks exist when consolidation or centralization results in monopolistic control of centralized hubs as single points of failure. Moreover, non-hierarchical peer-to-peer strategies may work fine for delivery of large amounts of data to a known destination, but they will not necessarily work well for search and discovery of a small datum at an unknown destination within a large universe of data.

Thus, simple flat non-hierarchical or peer-to-peer approaches that lack mobile metadata (such as the web site files of the Biositemaps initiative or the web page and data markup of the Linked Data initiative) will not scale to meet the demands presented by the ever accelerating growth in production of data, web pages, and web sites and services. Only an architecture with an HDMM style akin to the successful design of IRIS-DNS will scale sufficiently to meet the demands of the explosive growth in data required to solve the problems of brain diseases and other complex system challenges. This data will be available in varying kinds including public or private, raw or processed, analyzed as qualitative or quantitative results, interpreted as inferred conclusions, redacted for publication in literature, and thus, amenable to data mining, text mining, or both to varying degrees.

Figure 1. Beacons of Gondor dramatize a metaphor for the advantages of hierarchical communication networks that enable search and discovery of a small item in a very large world. If everybody remains trapped under the clouds in isolated valleys everywhere and unable to see elsewhere, then how will we (or software agents) communicate with each other fast enough to find and reach unknown destinations, persons (or agents), and small pieces of information in a large world that grows ever larger all the time?



To use a geographic metaphor, simple non-hierarchical approaches risk trapping the information seeker stuck and bogged down in the valleys of isolated lands around a world in which any information sought and found in an isolated valley is not shared with or redistributed to other isolated valleys. In contrast, properly designed and structured hierarchical approaches enable the information seeker to send message requests efficiently from any valley to the closest mountain peak and then from that mountain peak to other mountain peaks surveying all valleys in all lands (see Figure 1) in order to efficiently obtain the requested data which in turn then becomes automatically shared and redistributed in other valleys and lands as part of the response to the request. Such redistribution and sharing of information does not occur in the non-hierarchical approaches of initiatives like Biositemaps and Linked Data. Because there is no redistribution and sharing of mobile metadata in these initiatives, a crawler would traverse many more nodes of a linked network before possibly finding the search target node. As a result, searches would be less efficient and more likely to fail. Moreover, complicated queries dependent upon multiple searches would be more time consuming and less likely to return informative responses.

To use another metaphor, the simple non-hierarchical approaches lack the ability to scale and solve the worsening problems of finding needles in haystacks which can only be solved by the more sophisticated, versatile, and flexible hierarchical approaches of systems that implement the architectural style common to both IRIS-DNS and PORTAL-DOORS (see Section 7 for further discussion of these intuitive claims as the statement of a formal conjecture). Therefore, as the core infrastructure system for the biomedical informatics work pursued by the author, the PORTAL-DOORS System maintains the same principles of architectural design that have been so successfully tested and proven by IRIS-DNS for decades. In this regard, PORTAL-DOORS represents a dramatically different approach from all other current initiatives whether intended for the semantic web in general or for neuroinformatics portals in particular.

4. Hierarchically Distributed Mobile Metadata (HDMM) as an Architectural Style

According to Taylor *et al.* [8, page 73], "an architectural style is a named collection of architectural design decisions that (1) are applicable in a ... context, (2) constrain architectural design decisions [for] a system within that context, and (3) elicit beneficial qualities in each resulting system." Architectural styles are distinguished from architectural patterns in scope, abstraction and relationship. For example, with regard to scope, "an architectural style applies to a development *context* ... while an architectural pattern applies to a specific design *problem*" [8, page 73]. Moreover, architectural patterns are larger in scale than design patterns. Thus, in software engineering, there is a increasing progression of scale and scope from design pattern to architectural pattern to architectural style.

The REpresentational State Transfer (REST) architectural style [46] serves as an important example of an architectural style for network-based applications on the web. Other styles, such as peer-to-peer, have been named and described for distributed and networked architectures [8]. However, not all styles of distributed and networked architectures have been appropriately characterized and publicized with an identifying name and detailed description of the principles that constitute the essential distinguishing aspects of the style.

In particular, the architectural style that characterizes both IRIS-DNS and PORTAL-DOORS has just recently been elaborated with an explicit name and description [45] even though the PORTAL-DOORS System for the semantic web was purposefully designed by the author [1] by analyzing and emulating the architectural principles and paradigm of the IRIS-DNS System for the original web. Here, the current Section 4 presents the formal name and description for the architectural style shared by both IRIS-DNS and PORTAL-DOORS as pervasive registry-directory networks with Hierarchically Distributed Mobile Metadata (HDMM). Then below, the subsequent Sections 5 and 6 further discuss the architectural design and implementation of PORTAL-DOORS within the context of this HDMM architectural style.

IRIS registries [47] and DNS directories [48] provide the model for the architectural style that inspired the design of PORTAL registries and DOORS directories [1]. The most essential characteristics of this HDMM architectural style can be summarized by the following principles:

1. **Distributed infrastructure:** Pervasively distributed and shared infrastructure, content, and control of content including distributed and shared control over both the contribution and distribution of the content defined as the mobile metadata records.

2. **Hierarchical authorities:** A hierarchy of both authoritative and non-authoritative servers (root, primary, secondary, forwarding and caching) enabling global interoperable communication and exchange of the mobile metadata records while permitting independent administrative control of local policies governing the publication and distribution of the metadata records.

3. **Mobile metadata:** A focus on moving the mobile metadata for *who what where* as fast as possible with pervasive distribution and redistribution from servers in response to requests from clients that access non-authoritative local forwarding and caching servers updated regularly by the authoritative servers.

4. **Separated concerns:** A separation of concerns with registries for identifying resources and directories for locating resources that have been globally uniquely identified in the registries.

5. **Unrestricted identification:** A relative freedom of choice in the selection of identifiers with purposeful absence of any requirement to use the *same* root name or label for *all* identifiers, thus enabling essentially unrestricted choice of naming or labeling schemes for identification and thereby avoiding monopolistic control by any single organization.

Users of today's web browsers may not be familiar with the engineering of the hidden infrastructure system that enables them to navigate to any web site around the world. But it is the IRIS-DNS infrastructure system, which is responsible for registering domain names and mapping them to numerical IP addresses, that makes it possible for the user to browse the web in such an effortless manner almost always without ever typing, seeing, or even being aware of the existence of the numerical IP addresses.

Moreover, from the user's perspective, what is most important now is that the speed of this conversion from domain name to IP address occurs so rapidly that the user does not experience it as a hindrance or delay in browsing. Even if the particular web page itself downloads and displays slowly, usually at least the web site address is found quickly. And that happens because the small amount of metadata (domain name and IP address) moves so quickly across the internet even if the larger amount of data (web page text and media) does not. Because of this important point, the phrase *Hierarchically Distributed Mobile Metadata* and acronym *HDMM* was introduced (9 May 2009 at www.portaldoors.org) as a name for this architectural style that characterizes both IRIS-DNS and PORTAL-DOORS.

The term *mobile metadata* emphasizes the principle that the metadata moves throughout the distributed network of nodes which may include both stationary nodes such as wired rackmount servers and mobile nodes such as wireless handheld devices. When considering the latter case, the movement of the *mobile metadata* and the movement of the *mobile node* must be understood as different kinds of mobility. More generally, *mobile metadata* must be distinguished from *mobile software* and from *mobile systems*. Further, in the acronym HDMM, the MM serves as a mnemonic not only for *Mobile Metadata* but also for *Metadata about Metadata* (see Section 5.2), while the D recalls not only *Distributed* referring to location but also *Dynamic* referring to content. In other words, the metadata may both move to distributed and redistributed locations throughout the network, and also change frequently or intermittently with dynamically updated content.

These HDMM principles do require hierarchical control and distribution of metadata records, but do not require hierarchical identification of resources. Whereas IRIS-DNS does employ a hierarchical identification scheme with top-level domain names, domain names, and sub-domain names, PORTAL-DOORS does not require any such hierarchical naming scheme. In fact, PORTAL-DOORS allows complete freedom with an identification scheme for which globally unique labels are simply required to be URIs. These URIs may or may not be hierarchical, and they may or may not be resolvable URLs, as long as they are URIs. However, both IRIS-DNS and PORTAL-DOORS systems do employ hierarchical control and distribution of metadata records. Thus, for the purposes of defining an architectural style applicable to both IRIS-DNS and PORTAL-DOORS, the interpretation and use of the term *hierarchical* pertains to the distribution of metadata records but not to the identification of resources (see Table 1).

Whereas IRIS-DNS implements the HDMM architectural style for the original web, PORTAL-DOORS extends and implements this style for the semantic web and grid. Table 1 summarizes some of the similarities and differences between PORTAL-DOORS and IRIS-DNS from the perspective of considering both as distributed online database systems with entity registering and attribute publishing implemented with the HDMM architectural style (see HDMM Principles 1–5).

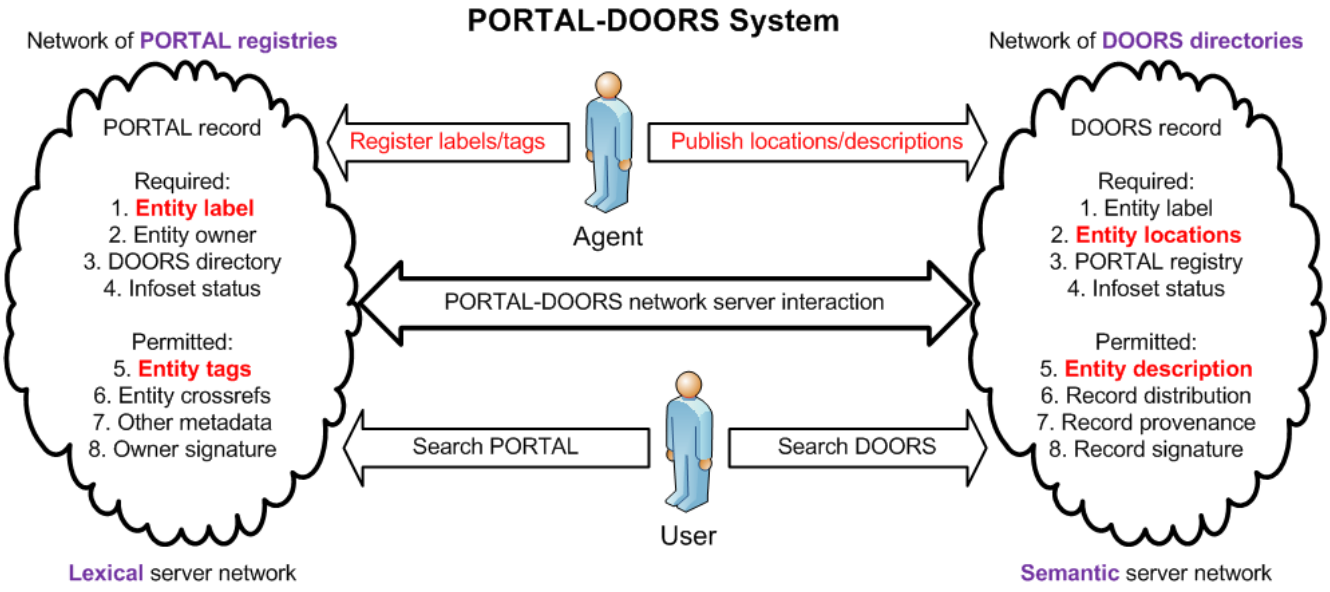
Table 1. Hierarchically Distributed Mobile Metadata (HDMM) Systems with Entity Registering and Attribute Publishing

	IRIS-DNS System	PORTAL-DOORS System
Dynamic metaphor	A distributed communications network brain of nodal neurons continuously updating, exchanging, and integrating messages about 'who what where'	
Static metaphor	A simple phonebook	A sophisticated library card catalogue
Registering system	IRIS registries	PORTAL registries
– Entity registered	domain	resource
– Identified by	unique name	unique label with optional tags
Publishing system	DNS directories	DOORS directories
– Attributes published	address and aliases	locations and descriptions
– Specified by	IP numbers	URIs, URLs, RDF triples referencing OWL ontologies
Entity identification	Hierarchical URL	Non-hierarchical URI
Record distribution	Hierarchical request forwarding and response caching	Hierarchical request forwarding and response caching
Serves original web	Yes, via mapping of character name to numeric address	Yes, via mapping of character label to URL for IRIS-DNS
Serves semantic web	No, because IRIS-DNS does not use RDF triples	Yes, via mapping of character label to semantic description
Crosslinks entities	No	Yes, via mappings within DOORS descriptions to other resources
Crosslinks systems	No	Yes, via mappings within PORTAL crossreferences to other systems

5. Architectural Design of the PORTAL-DOORS System

In accordance with the HDMM architectural style, PORTAL-DOORS has been designed to serve the semantic web and grid in a manner analogous to the way that IRIS-DNS has served the original web. The fundamental exposition of the architectural design for PORTAL-DOORS was published in the original blueprint paper [1] from which Table 1 and Figures 2 and 3 have been adapted. They have been updated with the revisions published in [11,39,45], and further enhanced by those introduced here for multilevel metadata and other new functionalities including aliases, priorities, and metaresources. Note also that the original *separate* design of PORTAL registries and DOORS directories [1] has been supplemented with an alternative bootstrapping *combined* design with integrated NEXUS registrars [11]. Both can coexist together. Further, PORTAL-DOORS extends the separation of concerns principle (see Item 4 above) to include the additional notion of separately optimising directories for semantic services (with use of logical reasoning, ontologies and the RDF/OWL/SPARQL stack of technologies) and the registries for lexical services (with use of character string processing, terminologies and only those XML technologies that do not require use of RDF triples). This separation of concerns enables the back-end use of traditional relational database stores for PORTAL registries and RDF-triple database stores for DOORS directories. Of course, XML stores and/or hybrid stores (such as OpenLink Virtuoso virtuoso.openlinksw.com which is an open source cross platform universal server) can also be used for both PORTAL and DOORS servers and services.

Figure 2. PORTAL-DOORS System Data Records: Resource metadata is registered and published by agents for search by users in the PORTAL-DOORS server networks. *Semantic* services here are defined as those using the RDF/OWL/SPARQL stack of technologies, whereas *lexical* services are defined as those using only character string processing, terminologies, or those XML technologies that do not require use of RDF triples. Fields within data records are considered *required* or *permitted* with respect to the schemas maintained by the root servers. The figure displays only the most important fields; for all fields, see the reference model implemented with XML Schemas.



5.1. Core Design

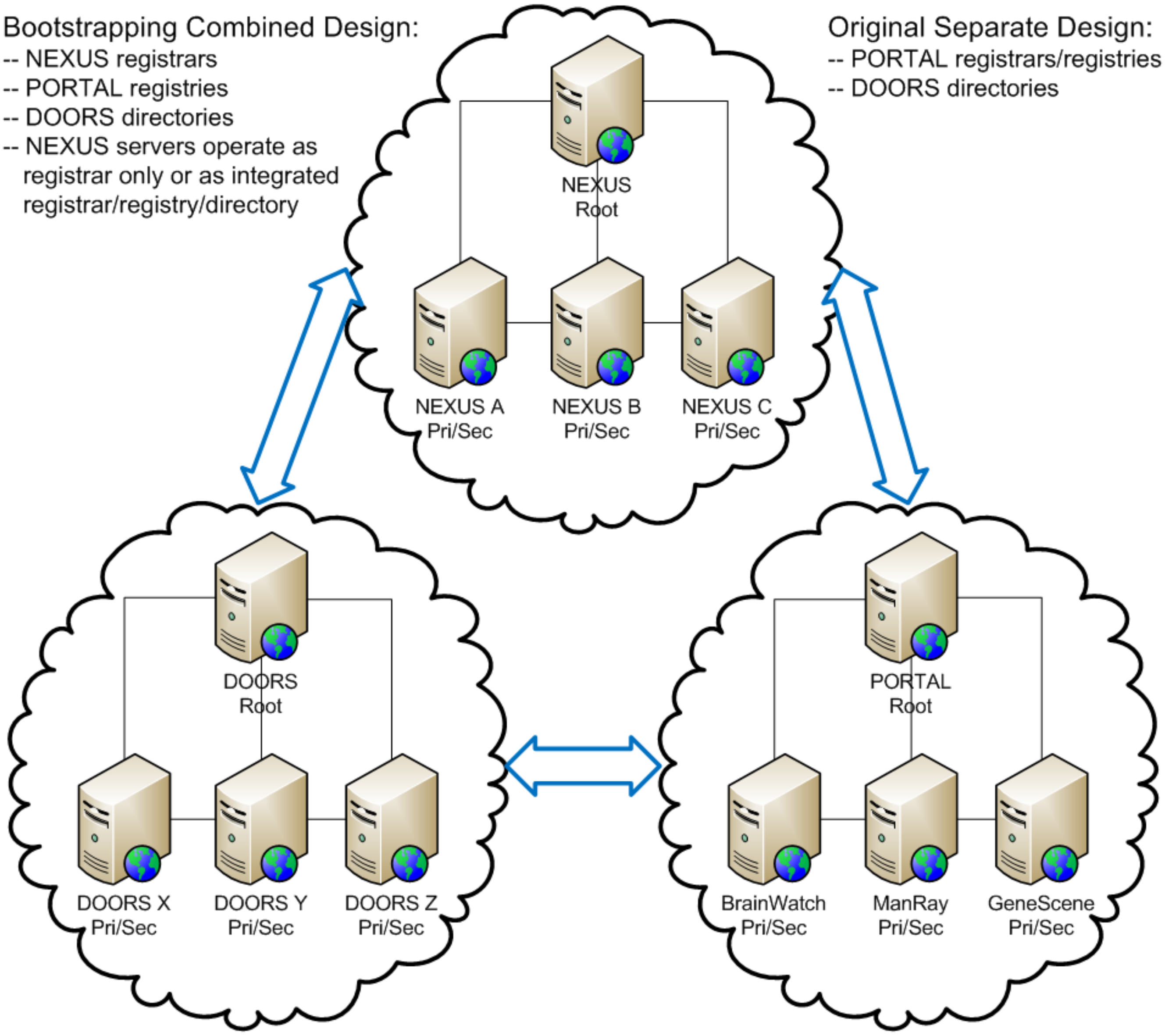
Figure 2 displays a diagram depicting the structure of data records at PORTAL registries and DOORS directories. Figure 3 displays a server network diagram for root, primary, and secondary DOORS directories interacting with root, primary, and secondary PORTAL registries with examples from problem-oriented specialty domains in biomedical informatics [39].

Details of the PORTAL-DOORS architecture have been previously elaborated in the publications [1,11,45] with formal definitions and specifications of the core model for database record fields and web server functions. Some important characteristics of the design include the following principles:

- A distributed network of registries and directories for resource metadata oriented by problem domain or specialist community rather than by technology format of the resource.
- A hierarchical system enabling local independence of communities while simultaneously maintaining global interoperability and compatibility for communication between and search amongst different specialty communities.
- A hybridized architecture with both XML Schemas and terminologies serving the original web and also RDF triples and OWL ontologies serving the semantic web to bridge and transition from the original web to the semantic web.
- Pervasively distributed and shared infrastructure permitting use of any micro-format, terminology or ontology to promote democratization and evolutionary adoption (*i.e.*, survival of the fittest, not necessarily the first).
- Hierarchical authorities (root, primary, secondary, forwarding, caching) and globally unique identifiers to prevent namespace conflicts when identifying resources while maintaining autonomy of local communities with control over local policies.
- Designed to accomodate any resource — whether abstract or concrete, offline or online, semantic or non-semantic — with either non-semantic descriptions using tags referencing terminologies or semantic descriptions using RDF triples referencing ontologies.
- Supported with cross-references to other systems whether legacy or contemporaneous.

The PORTAL-DOORS System is not another attempt once again to create a so-called ‘one stop shop’ that claims to be the ‘one and only’ destination for ‘all shopping needs’. In fact, the general philosophy of HDMM systems turns that notion upside down and argues that centralized single site ‘one stop shops’ also constitute single points of failure and/or single points of biased view, and thus cannot and will not solve all problems. Instead, there should be a multiplicity and diversity of registries and directories. Anybody who desires should be able to construct and maintain their own specialized registry or directory capable of exchanging the interoperable mobile metadata that becomes highly distributed, redistributed, and cached everywhere for speed and efficiency of search and location. Maintaining the interoperability of all registries and directories to communicate with each other transparently within the same PORTAL-DOORS infrastructure system facilitates achieving the goal of efficient search.

Figure 3. PORTAL-DOORS System Server Network: PDS server networks with interacting clouds of NEXUS registrars, PORTAL registries, and DOORS directories. NEXUS servers may expose either the NEXUS registrar service for the separate design or the integrated set of NEXUS registrar, PORTAL registry, and DOORS directory services for the combined design. These resource metadata server networks for PORTAL registering of labels and tags and DOORS publishing of locations and descriptions are analogous to domain metadata server networks for IRIS registering of names and DNS publishing of addresses. Primary PORTAL registries may be established by an organization or person who maintains any local policies governing registration of resources at that particular primary PORTAL registry. Examples shown here (GeneScene, BrainWatch, ManRay) implement policies with a problem-oriented focus on their respective specialty domains. Specific criteria for registration are determined by the local schema of the PORTAL primary which must nevertheless comply with the global requirements of the PORTAL root in order to assure interoperability between different PORTAL primaries.



5.2. New Multilevel Metadata Design

Managing the mobile metadata, both conceptually and technically, with regard to a hierarchy of metalevels also serves this goal. Therefore, as a new revision and enhancement for the PORTAL-DOORS System, a multilevel metadata analysis is introduced here in Section 5.2 while the corresponding multilevel metadata structures are further detailed in Section 6.1. The analysis begins with consideration of the collection of objects relevant to the resource in varying contexts.

1. **Resource entity:** The object of interest considered by the registrant to be the resource whether concrete or abstract, online or offline, semantic or lexical, real or virtual. This *resource entity* may be registered at a particular PORTAL registry only if it satisfies the registration requirements of that PORTAL registry. Depending upon the problem-oriented specialty domain of the PORTAL registry and its registration policies, examples may include persons, patients, investigators, authors, or organizations; online virtual entities or offline physical entities; data services, data storage tools, and data records (independent of and unrelated to any PORTAL-DOORS metadata record); analysis services and data processing tools; authored information, books, journals, papers, web sites, and web pages; and many other examples and categories within any field of interest defined by the administrators of the particular PORTAL registry.

2. **Resource record:** The database object containing information about the resource entity for the purpose of persistent storage. This *resource record* is stored in a database at a PDS server (a PORTAL, DOORS, or NEXUS server). Note that for the same resource entity, the information stored in a resource record at a PORTAL, DOORS, or NEXUS server will be different, and may also be different within each of the networks of PORTAL, DOORS, and NEXUS servers depending on their operation as authoritative primary or non-authoritative secondary and caching servers.

3. **Resource infoset:** The memory object containing information about the resource entity for the purpose of managing, displaying, and analyzing the information about the resource entity of interest. This *resource infoset* is assembled by the responding PDS server that gathers all of the relevant information from possibly multiple distributed records located at various different PORTAL, DOORS, and NEXUS servers.

4. **Resource representation:** The serialized object, obtained from the memory object, representing all of the information collected and assembled about the resource entity for the purpose of interoperable information exchange compliant with the PDS interface. One or more of these *resource representations* are sent by the PDS server in response to requests from clients if the server is configured to return a response without a message envelope.

5. **Resource message:** The message object containing one or more serialized resource representations within an envelope for the purpose of interoperable information exchange compliant with the PDS interface. This *resource message* is exchanged between different PDS servers and/or is sent by the targeted PDS server in response to requests from clients.

The term *information* is used in the list above in a general sense referring to content without implying any special connotations about a hierarchy of metadata levels. This term *information* will continue to be used to refer to any part of the content collectively contained in all of the metadata levels independent of any discussion of data versus metadata versus meta-metadata, or of multilevel metadata.

Metadata can be associated with each of the five objects listed above. The following list summarizes the metadata for each of the five objects together with the design principles that govern software implementation for the database, web service, and interoperable messaging interface schemas for the PORTAL-DOORS System.

1. **Entity metadata:** All metadata pertaining to the entity itself including tags, labels, locations and description of the entity as well as references to the owner and contact for the entity; corresponds to PDS schema element **EntityMetadata** and considered primary or *Level 1* metadata about the entity itself.

2. **Record metadata:** All metadata pertaining to the stored records about the entity and the process of registering and managing the records including timestamps for creating and updating the records, references to the governing registries and directories, as well as references to the registrant and agents for the records; note that the registrant and agent for the records may be different from the owner and contact for the entity; corresponds to PDS schema element **RecordMetadata** and considered secondary or *Level 2* metadata about the *Level 1* metadata.

3. **Infoset metadata:** All metadata pertaining to the dynamic infoset about the entity assembled from the distributed stored records including status, validation timestamps if validated, and any entailments if inferred by a reasoning engine; corresponds to PDS schema element **InfosetMetadata** and considered tertiary or *Level 3* metadata about the *Level 1* and *Level 2* metadata.

4. **Representation metadata:** Current design limited to use with only an identifier as an attribute on a wrapper element collating the three elements **EntityMetadata**, **RecordMetadata**, and **InfosetMetadata** respectively for the primary, secondary, and tertiary metadata; corresponds to PDS schema type **ResourceRepresentation** with element instances **PORTAL**, **DOORS**, and **NEXUS**.

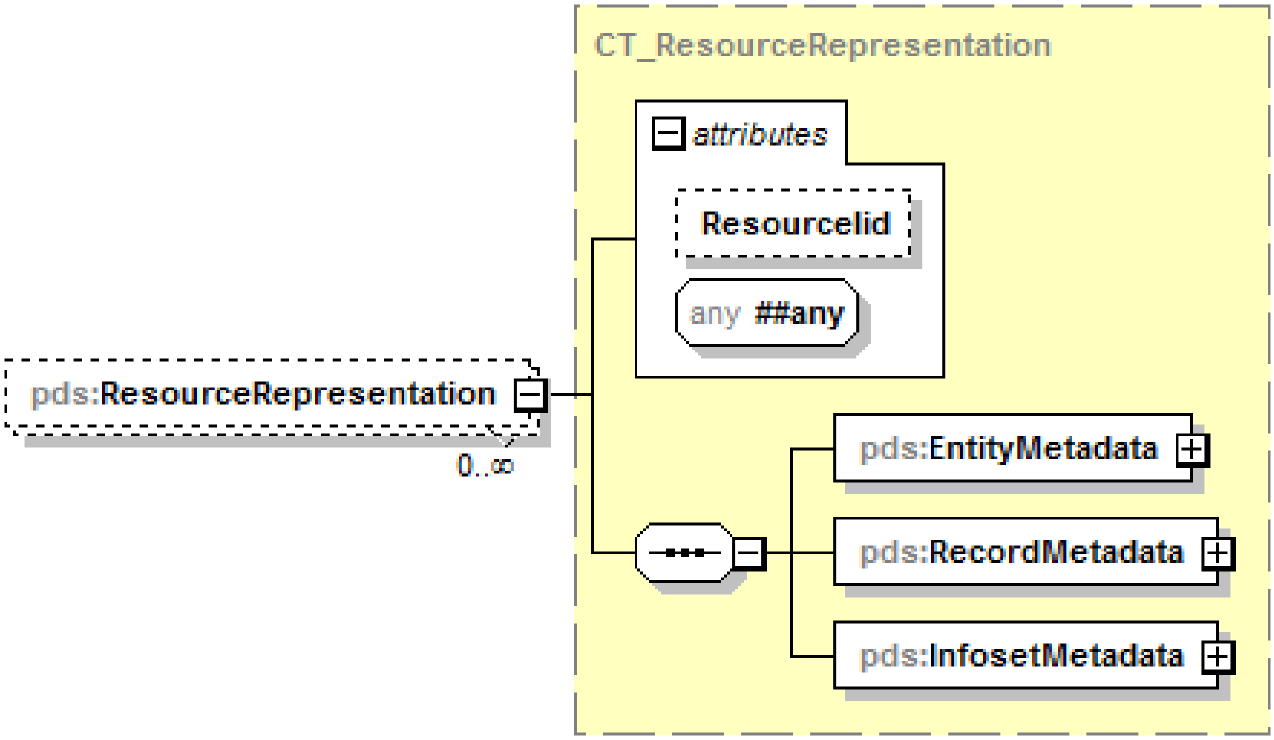
5. **Message metadata:** All metadata pertaining to the messaging envelope and the process of exchanging messages throughout the PORTAL-DOORS System; design based on using an analogy with the IRIS-DNS System; corresponds to PDS schema element **PDS** as the root element for all PDS messages.

As noted in the list above, implementation of the current design concepts for metadata in the PORTAL-DOORS System (see also Section 6) explicitly uses the term *metadata* in the names for the corresponding PDS schema elements for the entity, record, and infoset objects (but not for the representation and message objects) to emphasize the distinctions made for the first three hierarchical levels of metadata for any given resource (see Figure 4).

5.3. Other New Design Features

As part of the design philosophy, the PORTAL-DOORS Project maintains the following general goals: 1) practical utility, 2) flexible adaptability, 3) universal applicability with an approach of inclusivity rather than exclusivity, and 4) forward revisions by extension rather than restriction in order to maintain

Figure 4. Resource representation: entity metadata is primary or *Level 1* metadata about the entity itself, record metadata is secondary or *Level 2* metadata about the *Level 1* metadata, and infoset metadata is tertiary or *Level 3* metadata about the *Level 1* and *Level 2* metadata; see also Section 5.2.



backward compatibility and adherence to the original design goals and principles. These general design principles stand in contrast with the more specific design principles that characterize the HDMM architectural style for both the IRIS-DNS System and the PORTAL-DOORS System (see Section 4). Thus, in support of these general goals, the current iteration of re-design for the PORTAL-DOORS System also introduces several other important new functional characteristics and features:

- **Aliases:** The original design of PDS as published in [1] specified use of a URI or IRI as the globally unique identifier, called a *label*, for any resource registered in the system. The current revision reported here now allows multiple labels for any given resource and distinguishes between a single required *canonical label* and multiple permitted *alias labels* for the resource. All labels, whether canonical or alias, for all resources must always be globally unique throughout the system. Thus the original design requirement for uniqueness of labels has not been violated by this revision.

- **Priorities:** A number of the metadata fields, such as alias labels, supporting tags, supporting labels, locations, crossreferences, secondary registries and secondary directories permit multiple instances of the field for the same given resource. The current revision of PDS now allows for priorities to be assigned to these instances so that they can be ranked in order. A *priority* is defined to be a single-byte integer rank in the range from 0 to 255 with precedence order in natural counting order, *i.e.*, first 0 and last 255. In the case of multiple instances of labels for a resource, the canonical label is always identified by the assigned priority 0 with all other alias labels assigned any priority in the range from 1 to 255.

- **Metaresources:** The original blueprint design [1] also specified that resources can only be registered and managed by owners of the resources. This design principle yields a system that does not allow anonymous public editing of resources which is contrary to the policies adopted by many wiki systems. However, it is possible to design an extension of the initial PORTAL-DOORS System that maintains the original principle while also enabling secondary resources to be registered and managed by individuals who are not the owners of the primary resource. These secondary resources about primary resources are called *metaresources*. The secondary metaresources are declared by specifying their entity type as a special type called *meta-entity*. Secondary metaresources are required to maintain a reference to their targeted primary resources. This approach assures that all metaresources about the same targeted resource can refer consistently to that resource yet be managed independently of it as the primary resource and of each other as the other secondary resources. A scientific journal article as primary resource with multiple reviews as secondary metaresources constitute a simple example. All of the referees who write the secondary reviews and the authors who write the article should have control over their own resources without interference by others.

- **Agents:** The original description of PDS [1] distinguished between the roles of users and owners of resources and resource metadata. This terminology must be refined when considering the design of an implementation for a web site application or service. Therefore, the term *resource owner* now refers to the collection of person(s) and/or organization(s) presumed to own the resource while the term *resource agent* refers to the person who registers, manages and edits the information about the resource and who is presumed to be acting on behalf of the resource owner. As before, the term *resource user* refers to the person who anonymously consumes information from PDS. Thus, *users* have read privileges throughout PDS, whereas *agents* have read/write privileges at those registries and directories where they have been explicitly granted write privileges for creating and editing records. For the reference implementation described in Section 6, agents may edit information in *author*, *editor*, or *administrator* modes if granted access to these successively higher privileges. The reference implementation adopts the following conventions: In author mode, the agent may edit only records initially entered by the agent. In editor mode, the agent may edit any records in the same registry. In administrator mode, the agent may edit any registry or directory records accessible via the same registrar.

These new design principles are elaborated further in the context of the implementation with examples discussed in Section 6. Note that each of these revisions constitute extensions rather than restrictions of the original design. Thus, none of the original design principles have been violated.

6. Implementation and Application of the PORTAL-DOORS System

Software version 0.5 was developed [11] for the PORTAL-DOORS System that eliminates the redundancies, clarifies the terminology, and resolves the circular reference problem of the original blueprint

[1]. This alternative scheme called the *combined design* can coexist together with the original scheme called the *separate design*. The current version 0.6 reported here implements the new design features described above in Sections 5.2 and 5.3. All software has been implemented on the Microsoft Windows Server 2008 platform with SQL Server 2008 and Internet Information Server 7.0 using Microsoft Visual Studio 2008 ASP.NET 3.5sp1 and Altova XMLSpy 2010 as integrated development environments. Code, samples, and documentation for the data structures and interoperable messaging schemas reported in this article have been packaged in the accompanying zip file provided as Supplementary Material with this article. Interested readers may unzip the package and consult the README.txt file for further details on its contents.

Figure 2 displays a diagram summarizing the basic structure implemented for data records with both required and permitted fields at PORTAL registries and DOORS directories for both separate and combined schemes. Not all fields are shown in that summary figure. However, Figures 5 and 6 together display all fields for a resource record managed at a NEXUS server including the administrative support structures necessary for managing resource agents and their access rights. Any node in the PDS network can now be built as a separate PORTAL node, separate DOORS node, or a combined PORTAL-DOORS node also called a NEXUS node. Figure 3 displays network clouds of these interacting PORTAL registries, DOORS directories, and NEXUS registrars. The alternative combined design offers significant advantages in enabling an efficient self-referencing, self-describing, and bootstrapping process amongst the core system constituents (agents, registrants) and components (registrars, registries, and directories). So far, only single-site functionality has been implemented; see the development roadmap in Section 6.5 for planned multi-site functionality.

6.1. Implementation of Current Version 0.6

When providing registrar services for separate PORTAL and DOORS nodes, NEXUS registrars operate in a manner consistent with the original separate design. However, when providing registrar services for a combined PORTAL-DOORS node, NEXUS registrars can also operate in a manner that enables integrated storage of both PORTAL and DOORS record data on the same server as currently implemented in version 0.6 and reported here. Figure 5

displays a diagram depicting the relational database model for the current 0.6 draft version of the PDS schemas available at www.portaldoors.org. This data structure model shows the primary and foreign keys that provide referential integrity constraints for the relational database tables of a NEXUS server node in the network system. Figure 6 displays the main table in relation to the auxiliary and administrative support tables for managing agent access to the system.

All PDS tables in the database are named with the prefix **pds_** to distinguish them from the tables of other administrative providers such as Microsoft's ASP.net authentication and authorization services and their database tables named with the prefix **aspnet_**. Further, in order to simulate management of PORTAL, DOORS, and NEXUS network nodes at the same site in the same database, the tables for each of these servers are named respectively with the prefixes **pds_P**, **pds_D**, and **pds_N** while tables common to all three server types are named with the prefix **pds_A**. In the following discussion, the prefix **pds_** appears in the figures but not in the text where it should be assumed.

Figure 5. Relational database model for NEXUS combined design server with integrated storage of both PORTAL and DOORS data record fields as a NEXUS data record. See Figure 6 for the administrative content of a NEXUS record.

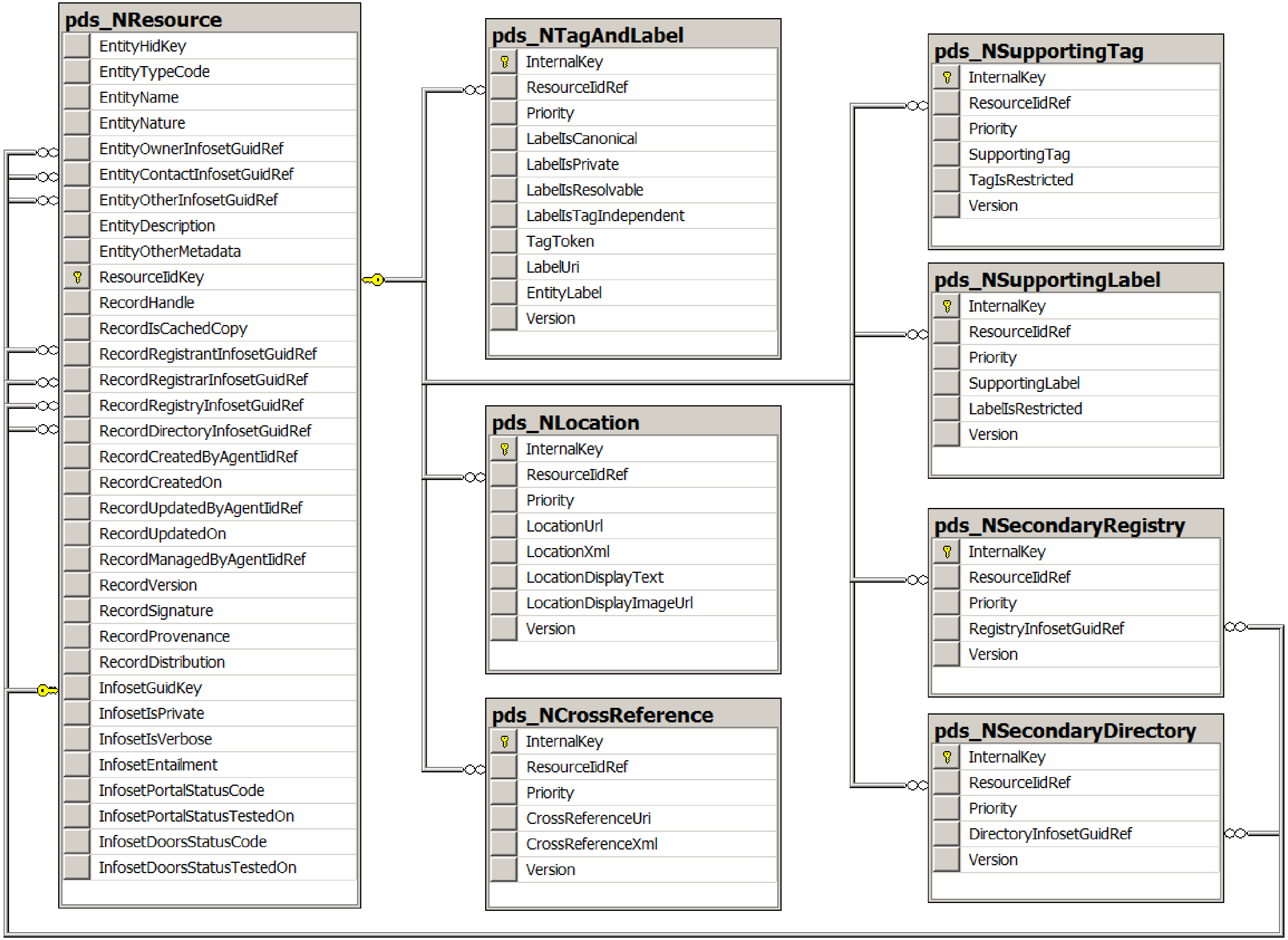
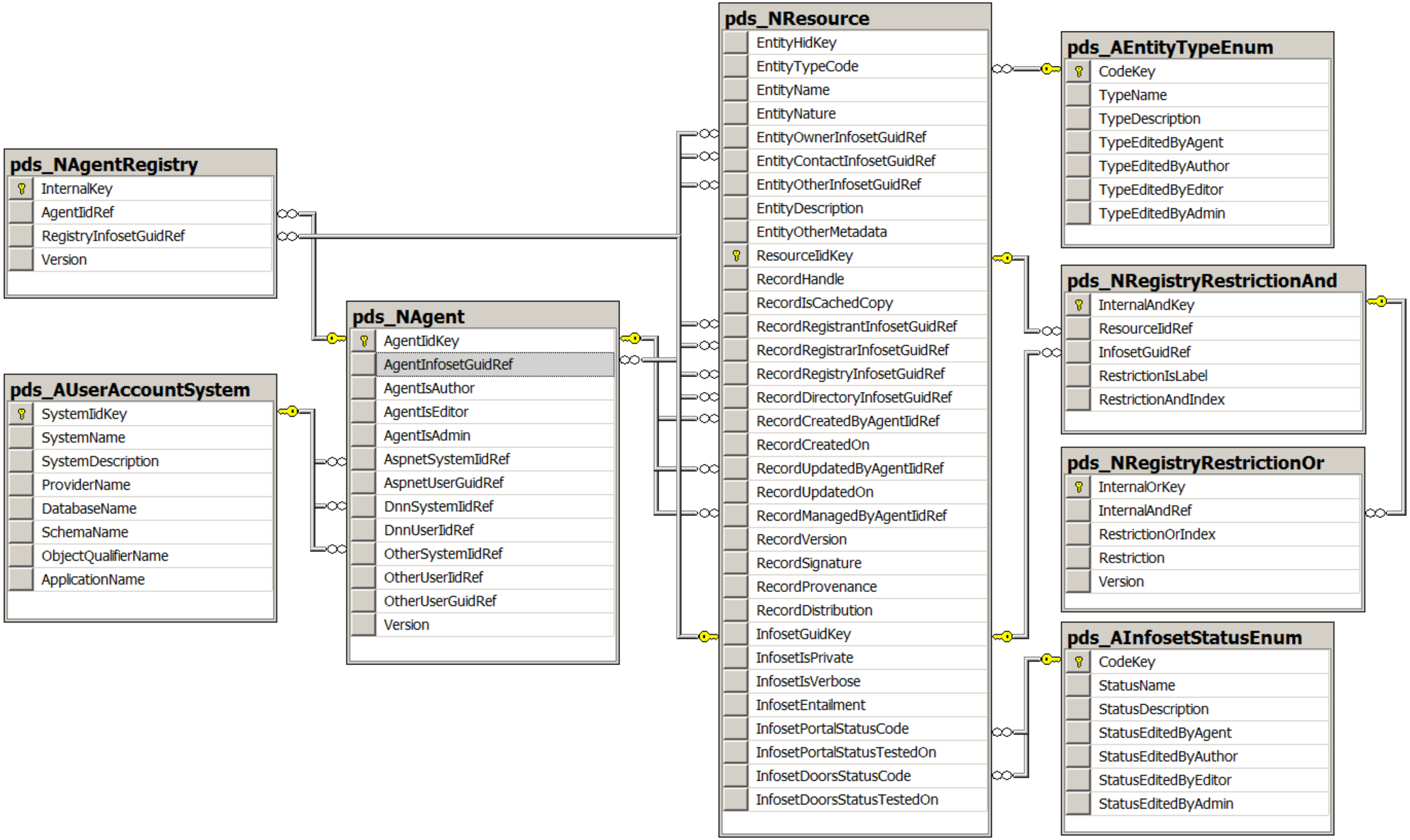


Figure 6. Relational database model for the auxiliary and administrative support tables for system and agent management in relation to the main table for the NEXUS combined design server. See Figure 5 for the non-administrative content of a NEXUS record.



With a conventional master-detail relationship, the table **NResource** serves as the main table for NEXUS resource records with primary key **ResourceIidKey** (an integer identifier) for the related records connected in a one-to-many relationship via foreign keys **ResourceIidRef** in each of the dependent tables **NTagAndLabel**, **NLocation**, **NCrossReference**, **NSupportingTag**, **NSupportingLabel**, **NSecondaryRegistry**, and **NSecondaryDirectory**. With the column ordering for the main table **NResource** as displayed in Figure 5, note that the fields displayed above the primary key **ResourceIidKey** are entity metadata fields whereas those displayed below the primary key are first the record metadata fields and then the infoset metadata fields.

Because of the conceptual distinctions between the different kinds of metadata and the different ways that the metadata can be used, providing distinct keys for the different subsets of metadata offers greater convenience for various usage interface and programming contexts. The primary key **ResourceIidKey** is intended mostly for internal use with the foreign keys **ResourceIidRef** by the database to maintain the master-detail relationships between the main and dependent tables for the virtual record created for each resource. All other keys visible as explicit fields in the main table **NResource** of Figure 5 are considered optional: **EntityHidKey** is a T-SQL hierarchical identifier for the entity, **RecordHandle** is a character string identifier for the record, and **InfosetGuidKey** is a T-SQL globally unique identifier for the infoset.

Technically, the PDS design specification requires only the resource label as the globally unique identifier for the resource metadata record. Although not visible as an explicit field, it is available as the **EntityCanonicalLabel** from a T-SQL view on the related tables **NResource** and **NTagAndLabel**. Data types for the optional keys have been chosen to facilitate conventions as well as meaningful intended uses. For example, **ResourceIidKey** as an integer is used to maintain all master-detail table relations for a single resource (see in Figure 5 all foreign keys linking into the right side of the field **ResourceIidKey** in the table **NResources**), while **InfosetGuidKey** as a guid (the T-SQL uniqueidentifier datatype) is used to maintain all references from one resource to another distinct resource within the self-referencing self-describing scheme of the relational data model (see in Figure 5 all foreign keys linking into the left side of the field **InfosetGuidKey** in the table **NResource**).

As another example, short-length character string handles for a record are more appropriate for agents (if persons, not webbots) editing the record at a single site, whereas medium-length guids for an infoset are more appropriate for servers communicating and exchanging records between multiple PDS sites.For internal PDS processing (interpreted as either within a single PDS server or within the PDS network between PDS servers), medium-length guids can also be more convenient than potentially very long-length labels assuming that the guids and labels are maintained in a strict one-to-one mapping correspondence for the same resource.

Using more than one identifier (*i.e.*, in addition to the required resource label), such as the example pair of both a record handle and a resource label, also enables the agent to maintain the information for the resource entity — even changing the label — without being required to delete the record and create a new record. The new facility that enables the use of alias labels together with the canonical label for a resource entity provides another mechanism to achieve a similar task while also enabling use of multiple different identifiers appropriate in different contexts or at different times. In this case, both a newer alias and an older alias can be maintained in addition to the canonical label if desired. Alternatively, an alias label can be re-declared to be the current canonical label.

For the resource entity metadata within the main table **NResource**, there are three directly self-referencing relations from fields with the suffix **_GuidRef** to three other resources for the **EntityOwner**, **EntityContact**, and **EntityOther**. Any resource may be registered with references **EntityOwner** and **EntityContact** to other resources for the entity owner and contact, but only metaresources of the special type meta-entity may be registered with a reference **EntityOther** to the targeted primary resource (see Section 5.3). In fact, the metaresource cannot be validated without this reference **EntityOther**. For the resource record metadata within the main table **NResource**, there are four directly self-referencing relations from fields with the suffix **_GuidRef** to four other resources for the **EntityRegistrant**, **EntityRegistrar**, **EntityRegistry**, and **EntityDirectory**.

There is no requirement that the necessary information for all of these seven other possible resources be stored at the same NEXUS server node. However, if so, then each can be referenced via the **_GuidRef**, and if not, then it can be referenced via the analogous **_Label** fields (not shown in Figure 5). For example, the resource for the **EntityContact** can be referenced internally via the **EntityContactGuidRef** or externally via **EntityContactLabel**. Check constraints can be used to prevent both the **_GuidRef** and the **_Label** for the **EntityContact** from being simultaneously non-null. Alternatively, appropriate programming logic can be used to maintain precedence of the internal reference via the **_GuidRef** over the external reference via the **_Label**, or vice versa, depending on the non-null values of these fields in the context of the status of the boolean field **RecordIsCachedCopy**.

For the resource record metadata within the main table **NResource**, there are also three *indirectly* self-referencing relations from fields with the suffix **_ByAgentIidRef** to three other potential resources for the **RecordCreatedBy**, **RecordUpdatedBy**, and **RecordManagedBy** agents. The indirect self-referencing via the auxiliary linking table **NAgent** (see Figure 6) provides a simple permission management system implemented with the feature of sufficient flexibility to interface with various user account provider systems, and simultaneously, to render optional the publication of any information pertaining to agents as resources distinct from owners, contacts and registrants.

Thus, the linking table **NAgent** mediates between the set of tables for PDS and another set of tables for the authentication and authorization system for managing agent access to inserting, updating, and deleting records in the NEXUS tables. The linking table has a primary key **AgentIidKey** and various alternative optional fields available for linking to user membership providers such as the field **AspnetUserGuidRef** for linking to Microsoft's ASP.net membership provider, **OtherUserGuidRef** for linking to an alternate generic user membership provider, etc. In addition, the table **NAgent** provides the foreign key **AgentInfosetGuidRef** for linking back to a resource in the main table **NResource** for use in a scenario where the agents as persons with responsibility for managing resources in the database are themselves identified and described as resources in the main table.

Regardless of whether an agent is published as a resource, or vice versa, whether a resource is an agent, registrant or contact of type person or of any other type, all resources may be flagged as non-publishable by the boolean field **InfosetIsPrivate** in the table **NResource**. Also, regardless of code implementation with persistence of the value stored in the field **EntityLabel** or otherwise computed dynamically by concatenation of the **EntityPrincipalTag** with the label of the entity's registry, it should be emphasized that any PDS implementation must maintain the important requirement of uniquely identifying resources by the resource entity label which must be an IRI or URI. For PDS draft version

0.6, both SQL code for the relational database model and XML Schemas for the interoperable messaging interface are available for download from http://www.portaldoors.org with an operational NEXUS server implemented at https://www.telegenetics.net now available for agent access with registration of resources relevant to the biomedical problem-oriented specialty domains of the GeneScene, ManRay, BioPORT and BrainWatch registries [39].

Note, however, that the BioPORT registry has recently been renamed the Beacon registry, and that several new registries including the CTGaming [49], Gaia, Eywa, and HELPME registries have been added to the set of prototype registries for continuing development of PDS. In addition, operational RESTful web services for user or machine public access via the http protocol are available at NEXUS servers as demonstrated by the following examples:

- http://pds.biomedicalcomputing.net/[nexusServiceEndpoint]
- http://pds.brainwatch.net/[nexusServiceEndpoint]
- http://pds.genescene.net/[nexusServiceEndpoint]
- http://pds.nucmedlib.net/[nexusServiceEndpoint]

Each of these services makes available the following endpoints with templates:

- /nexus/resrep/find?{parameter=value}
- /nexus/resrep/search?{parameter=value}
- /nexus/resrep/{entityTypeName}/{entityPrincipalTag}
- /{registryPrincipalTag}/{entityPrincipalTag}

where the last template corresponds to the entity canonical label for the resource as demonstrated by the following examples:

- http://pds.biomedicalcomputing.net/beacon/ellitron
- http://pds.brainwatch.net/brainwatch/bha
- http://pds.genescene.net/genescene/omim
- http://pds.nucmedlib.net/manray/snm

The endpoints /nexus/resrep/find? and /nexus/resrep/search? are distinguished in that the former performs an exact case-sensitive lookup whereas the latter performs a case-insensitive partial match. Currently, both endpoints respond to querystring parameters **ptag**, **stag**, and **label** for lookups or matches on principal tags, supporting tags, and labels, respectively. The latter finds or searches both canonical labels and alias labels. Parameters can be combined, and additional parameters will be implemented in the next software version.

The design with endpoint templates beginning with /nexus/ for a NEXUS server enables distinction with /portal/ for a PORTAL server and /doors/ for a DOORS server that might be at the same host.

The design with endpoint template beginning with /nexus/resrep/ for access to a resource representation enables distinction with /nexus/agent/ for access to an agent, noting that an agent must first create an account on the system prior to creating any records for resource representations.

6.2. Infrastructure System versus Tools and Applications versus Content

PORTAL-DOORS as a lower-level infrastructure system must be distinguished from higher-level tools and applications built on the foundation of the infrastructure. PORTAL-DOORS as a mobile metadata management, communication, and distribution system must also be distinguished from the actual metadata that the infrastructure is designed to send, receive, and exchange throughout the system. Fundamentally, the PORTAL-DOORS System establishes an interoperable, platform-independent, application-independent, messaging interface standard for information exchange over the internet. The design of this infrastructure system is guided by the HDMM architectural style and mandated to fulfill additional requirements in order to serve both the original web and semantic web as specified in the blueprint paper [1], here in this report, and partially implemented in the current draft version 0.6 of a reference implementation written in XML Schema *.xsd files.

Work to complete this reference implementation must clarify and stabilize not only the structural data model for the resource records and messages, but also the functional behavioral model for the PORTAL and DOORS services in response to requests from clients. Servers and clients must also communicate over transport protocols. The PORTAL-DOORS Project maintains a vision of serving more than one transport protocol as discussed in Section VII.E. of [1]. Initial drafts of the PDS schema files (prior to version 0.5) assumed use of the IRIS core protocol. The previous draft (version 0.5) addressed only the structural data model. The current draft (version 0.6) has re-introduced use of a specific transport protocol but replaced the IRIS core protocol with an http protocol using RESTful web services. At present, in a bootstrapping stage of development for the PORTAL-DOORS System, RESTful web services provide a more favorable environment for promoting adoption of the system. However, a fully dedicated and optimized protocol specifically for PORTAL-DOORS may ultimately prove necessary to achieve the speed and efficiency comparable to that which exists now for IRIS-DNS.

As PORTAL-DOORS continues to be developed and implemented, any client tool, application, or web site that accesses PORTAL-DOORS must be distinguished from the system itself. The PORTAL-DOORS System should not be considered either a single site or repository any more than the IRIS-DNS System of domain name registries and directories could be construed to be a single site or repository. For both IRIS-DNS and PORTAL-DOORS infrastructure systems, server-side data stores and services and client-side tools and applications can be written in any language on any platform. Client tools are necessary for agents to edit the information maintained at an individual server data store. Client tools are also necessary for agents and users to navigate, search and query the information stored not only at a particular server but also throughout the entire network of servers. These tools include faceted browsers, keyword search utilities, and SPARQL query interfaces.

Even more complex applications can be built in which the navigation, search, and query tools may be embedded within more sophisticated applications that hide these tools from the user interface. An important example is an application component that would provide natural language answers to natural language questions in the context of the overall function of the software application. In this example, the component converts the user's natural language question to a SPARQL query submitted to PORTAL-DOORS, and then converts the query response from PORTAL-DOORS back to a natural language answer for presentation to the user.

The usefulness of any technology system designed to manage content, regardless of how it is constructed from interface standards, server networks, client tools, and applications, is only as good as the content that it manages and exposes to producers and consumers of the content. Without this content exposed by the system, the system itself remains of limited practical utility. Thus, generation of content remains an important aspect of the development of any content management system. At present with a web browser interface, entry of metadata records into PORTAL-DOORS is performed by human agents much akin to the manner of entry for metadata records into IRIS-DNS.

However, software agents such as webbots and converters could be developed which would be able to generate metadata records for resources automatically. Presumably, there would be a trade-off in the quality of content produced versus the rate of content production when comparing records created automatically by software agents with records curated by human agents. This trade-off would not be applicable to those situations where an existing structured database only needs an appropriate interface for inbound queries and wrappers for outbound responses in order to expose metadata records for resources contained within the database. It would also not be applicable for existing structured databases such as those discussed in Section 7 that are easily convertible to PDS format with automated utilities.

6.3. General Usage Scenarios for the PORTAL-DOORS System

PORTAL-DOORS has been designed to be as flexible as possible with both backward and forward compatibility from Web 1.0 to Web 3.0. Given the partition with lexical non-semantic services on the PORTAL side and semantic services (with use of the RDF/OWL/SPARQL stack) on the DOORS side, and also the partition with both *required* and *permitted* elements for each of PORTAL and DOORS, there are many possible scenarios for usage of the entire PORTAL-DOORS System. Some examples include:

- Minimal use of required elements for both PORTAL registries and DOORS directories: This scenario essentially reduces use of the system to an alternative equivalent to the use of PURLs [50] (and other similar services). However, it does so *without* requiring use of a pre-determined URL identifier root like purl.oclc.org and instead allowing use of any identification scheme as long as it is a URI or IRI.

- Maximal use of permitted elements for PORTAL registries but minimal use of required elements for DOORS directories: This scenario enables exploiting the full metadata management facilities of the PORTAL non-semantic services (which include provisions for tags, micro-formats, cross-references, etc) without any obligation to use the DOORS semantic services (that necessitate use of the RDF/OWL/SPARQL stack of technologies and tools). This scenario enables resource agents to publish metadata now in non-semantic formats and defer until later any possible transition to semantic formats which would then be facilitated by the prior staging in the non-semantic formats.

- Minimal use of required elements for PORTAL registries but maximal use of permitted elements for DOORS directories: This scenario serves those situations where there is no barrier to transition the metadata from original web formats to semantic web formats, and the resource owner and agent do not wish to maintain the metadata in both semantic and non-semantic formats. This scenario requires that the resource agent registering and publishing the metadata already has access to established ontologies that can be referenced by semantic tools for describing the resource.

- Maximal use of permitted elements for both PORTAL registries and DOORS directories: This usage scenario provides the significant benefit of exposing as much metadata as possible to as many clients as possible including both older non-semantic as well as newer semantic tools and applications.

Enabling these usage scenarios constitutes an important goal for the PORTAL-DOORS Project which also includes the following tasks:

- Complete development of a specification model for the PORTAL-DOORS System as the interoperable informatics infrastructure using the Hierarchically Distributed Mobile Metadata (HDMM) architectural style for a distributed network of registries and directories.
- Complete implementation of a reference model with XML Schemas for the interoperable communication interface standards and with RESTful web services for the transport protocol.
- Build open source software clients and servers for multiple platforms, operating systems and programming languages according to the detailed roadmap (see Sec. 6.5) for continuing development of the previously published designs and prototypes.

The PORTAL-DOORS Project for development of the PORTAL-DOORS System thus serves to build the necessary foundation and core infrastructure for an information-seeking support system [51] upon which higher-level applications can be constructed.

6.4. *Specific Use Cases for the PORTAL-DOORS System*

The original PORTAL-DOORS blueprint paper [1] discussed the following use cases:

- Assisting with organization of the 'bioinformatics resourceome' and the description, discovery and use of resources for e-science and e-medicine in health care and life sciences (see [1] Sec. III).
- Cataloguing resources for biomedical computing (see [1] Sec. IV and VIII).
- Cataloguing patents and trademarks and relating them to products and services for e-business (see [1] Sec. IX).
- Assisting with semantic search, decision support and knowledge management applications in translational research and drug discovery for personalized medicine (see [1] Sec. XI).

More detailed descriptions of examples in the context of biomedical translational research include the following use cases of PORTAL-DOORS as an information-seeking support system for:

- Pharmacogenomic molecular imaging [9].

- PET and SPECT brain imaging [40].

Although originally conceived and described in the context of health care and life sciences, the diversity of possible use cases for PORTAL-DOORS remains as universal as the diversity of possible use cases for IRIS-DNS.

6.5. Development Roadmap for the PORTAL-DOORS System

Current plans envision following a PORTAL-DOORS Project roadmap with iterative software development for the PORTAL-DOORS System with these milestones:

- Version 0.5: Implementation as an AJAXified web application with back-end database and front-end web browser client for partial PORTAL server functionality and partial DOORS server functionality. Version 0.5.4 was the last 0.5.* version published on 3/29/2009.

- Version 0.6: Implementation as RESTful web services with both ASP.net based clients enhanced with user-friendly graphical user interfaces and editors for managing (entering and updating) data records at PORTAL-DOORS servers on Microsoft Windows platforms. The current version 0.6.4 is operational as a RESTful web service for user access and an AJAXified web application for agent access.

- Version 0.7: Completion and revision of lexical PORTAL functionality including interoperability with terminology tools.

- Version 0.8: Completion and revision of semantic DOORS functionality including interoperability with ontology tools.

- Version 0.9: Implementation as RESTful web services with JAVA based servers and clients for Linux and Mac OS X platforms.

- Version 1.0: Official release of PORTAL-DOORS System models and schemas for an authoritative server at a single site for all platforms.

- Version 2.0: Multi-site functionality (including security) for distributed interacting authoritative servers.

- Version 3.0: Multi-site functionality (including provenance) for distributed interacting non-authoritative servers operating with request forwarding and response caching amongst the distributed servers.

7. Discussion

As a cyberinfrastructure, PDS can be considered an information-seeking support system [51,52]. With an appropriately enhanced user interface, PDS can be considered a facetted search tool [52,53]. Regardless of use as infrastructure system or application tool, PDS interlinks registries, directories, databases, and knowledgebases across domain-specific fields, disciplines, and specialties. It assures globally unique identification of resources while promoting interoperability and enabling cross registry and cross directory searches between different problem-oriented, not technology-restricted, domains because of the fundamental definition of a resource as any entity, abstract or concrete, real or virtual, online or offline.

PDS has been designed as a hybrid bootstrap and bridge to transition from the old lexical web to the new semantic web, and allows for all constructs from free tagging and folksonomies to microformats, terminologies and ontologies. It supports mass participation and collaboration via its hierarchical and pervasively distributed but localizable infrastructure, and as a consequence, provides a democratized solution to the problem of search engine consolidation. Mowshowitz and Kumar [5] discuss both the realities and the risks of search engines that effectively restrict access to information, and argue that this problem represents a serious concern.

With its infrastructure designed in a distributed manner that permits localized control of policies and content and thereby prevents the possibility of search engine consolidation, the PORTAL-DOORS model is most similar in conceptual paradigm to the IRIS-DNS model that inspired it. In contrast, it can be compared to other familiar models for information management systems exemplified by the Google search engine (www.google.com) or the Wikipedia encyclopedia (www.wikipedia.org). In the case of Google, the server infrastructure is distributed (to some degree) but not the control of content (unless paid placement is considered). In the case of Wikipedia, the servers and content are centralized but the control of content is shared by all contributing anonymous authors and editors. However, in the case of PORTAL-DOORS, the server infrastructure, the content control, and the content itself are all shared and distributed (with the exception of any content declared to be private by its owners). Moreover, the design of the PORTAL- DOORS framework remains analogous to that of the IRIS-DNS framework with mechanisms that enable metadata about resources to be hierarchically distributed and redistributed as mobile content with request forwarding, response caching, and dynamic updating.

Continuing progress on the development of PDS with its NEXUS registrars, PORTAL registries, and DOORS directories will focus on implementing all features of the design including both data structures and operational methods for both independent and interacting servers. Content for PORTAL-DOORS will be contributed manually by human agents as has been done for IRIS-DNS. Later, when additional automated or semi-automated software agents, webbots, and/or converters become available, more content will be generated or converted as has already been done in the case of the 25,588 records of the MeSH2010 PORTAL Thesaurus [54,55]. For manually contributed content compared with automatically generated content, there may be a trade-off in the quality of content produced versus the rate of content production. This trade-off would not be applicable to those situations where existing databases only need an appropriate interface for inbound queries and wrappers for outbound responses.

As the internet, web, and deep web continue to grow in size, the study of algorithmic search in network graphs will become more important. Distributed networks can be studied as random graphs and other dynamic models of network growth for analysis of node degree distributions, clustering, and preferential attachment [56]. It is in this context that it will be necessary to ask questions such as what is the best search path for which situation and setting.

- Should search be pursued via hierarchical, peer-to-peer, or alternative network paths?
- Which search path is best when attempting to locate a resource known to exist somewhere?

- Which search path is best when attempting to establish the existence of a resource not known to exist *a priori*?
- Which search path is best for a commoditized resource for which any instance will satisfice?
- Which search path is best for a uniquely individual resource for which only the unique instance will suffice?

These questions relate to investigating and elucidating the search problem of identifying and locating an unknown network node in contrast to finding the optimal network path between a known source node and a known target node. In other words, establishing a communications channel as a network path for the movement of data between a known origin and known destination remains fundamentally a different problem than the search and discovery problem for an unknown, possibly nonexistent, node in a network that spans the globe.

In response to these questions for future study, consider the Hierarchically Distributed Mobile Metadata (HDMM) architectural style (see Section 4 and the "Beacons of Gondor" metaphor in Figure 1) that characterizes both the IRIS-DNS System and PORTAL-DOORS System as distributed registry-directory who-what-where metadata management systems. Consider also the success of the IRIS-DNS System for the original web. Assume that comparable success could be attained by the PORTAL-DOORS System for the semantic web if it is successfully developed and deployed into a distributed network cyberinfrastructure as extensively as the IRIS-DNS System. If so, then the following prediction, called the HDMM Conjecture, can be stated as a formal hypothesis to be tested.

> **HDMM Conjecture:** *Semantic HDMM networks should scale more efficiently than semantic peer-to-peer networks and thus should be more useful for searching by various query criteria for an unknown resource entity at an unknown resource location,* i.e.*, when existence of the resource is not known a priori.*

Section 3 provides additional discussion of the intuitive analysis, assumptions and claims that led to this conjecture. Future work with rigorous experiments and simulations comparable to those for large-scale networks [57–61], peer-to-peer networks [62–64] and super-peer networks [24,26] will either prove or disprove this conjecture. In the conduct of these experiments and simulations, careful attention must be paid not only to the definitions for each model (hierarchical versus super-peer versus decentralized peer-to-peer), but also to the different applications and corresponding sets of parameters, requirements, and constraints that pertain to each application. For example, Mastroianni [26] provides evidence in support of the super-peer model when applied to orchestrating the commodity of computing services in a multi-organizational grid.

However, without reference to careful *ad hoc* usage and definition of terms and models in the context of the application and any experiments comparing different models for that application, generic usage of the term *super-peer* may blur the distinction between *hierarchical* and *peer-to-peer* especially when a term such as *hierarchical* can itself be used in so many different ways as discussed in Sections 2.1 and 4. Does this model question concern the process of registration, identification, publication, location or description? In the context of the HDMM architectural style for the IRIS-DNS and PORTAL-DOORS

Systems which both permit mirrors of the roots and pervasive redistributed caching of all mobile metadata records, should the HDMM systems be construed as hierarchical, super-peer or peer-to-peer?

Whereas some aspects of the HDMM systems do operate in a peer-to-peer sense and in a super-peer sense, the most essential aspect remains the presence of a single root authority (which may be mirrored) for establishing the hierarchy of authorities governing the registration of resource entities and publication of resource metadata records, thus giving rise to the architectural style's name Hierarchically Distributed Mobile Metadata (see Section 4). Therefore, as stated above in the HDMM Conjecture, this report maintains the simpler dichotomy of hierarchical versus peer-to-peer without considering super-peer. This simplification enables comparison of PORTAL-DOORS as an example of a hierarchical system in contrast with Biositemaps and Linked Data as examples of peer-to-peer approaches.

8. Conclusion

As part of an ongoing iterative reassessment and revision of the architectural design for the PORTAL-DOORS System, the architectural style common to both PORTAL-DOORS and IRIS-DNS as pervasive registry-directory networks for who-what-where metadata management, respectively for the semantic web and original web, has been named the Hierarchically Distributed Mobile Metadata (HDMM) style. This HDMM architectural style has been characterized with a description of the design principles and constraints that define it.

For the HDMM architectural style, the notion of hierarchy refers primarily to the manner in which the mobile metadata is distributed and redistributed under the control of hierarchical authorities including authoritative primary and non-authoritative secondary and caching servers. It does not refer to the hierarchical naming scheme used for identification of resources in IRIS-DNS because there is no requirement for any such hierarchical naming scheme in PORTAL-DOORS.

With respect to the novel introduction here of multilevel *metadata about metadata* and implementation of the use of entity metadata, record metadata, infoset metadata, representation metadata and message metadata, the notion of hierarchy refers to the structure of the multiple levels of metadata. The *metaness* of a level of metadata (*i.e.*, its *metalevel*) can be most easily appreciated as a count of the hierarchical levels of indirection (*i.e.*, a count of the number of times metadata describes underlying metadata in successive layers of metadata about metadata). Thus, this usage of the notion of hierarchy refers to the manner in which the metadata can be organized conceptually in a recursion of layers and thereby structured and processed more effectively. This novel conceptual organization differs from all previous multilevel and/or hierarchical organizations of metadata which until now have been organized as hierarchies of a faceted, topical, categorical, spatial or temporal nature.

Additional new design features for PORTAL-DOORS including the use of aliases, priorities and metaresources have also been introduced, defined, and implemented. Finally, the current status of the PORTAL-DOORS System and future plans including the development roadmap for the PORTAL-DOORS Project have also been detailed in this report.

Acknowledgements

The author acknowledges support for this open source software project (www.portaldoors.org) from Telerik Corporation (www.telerik.com) which provided a subscription to their developer productivity tools and from Microsoft Corporation (msdn.microsoft.com) which provided a subscription to their developer technologies, integrated development environment, and servers platform. In addition, the author thanks G. Lazarin, D. Lockery, K. Marcynuk, J. Neidich, S. Sinesky and S. Taswell for contributing resource metadata records to the PDS servers. Finally, the author thanks J. Peters and the anonymous reviewers for their helpful comments and assistance with the preparation of this manuscript.

References

1. Taswell, C. DOORS to the Semantic Web and Grid with a PORTAL for Biomedical Computing. *IEEE Trans. Inform. Technol. Biomed.* **2008**, *12*, 191–204. In the Special Section on Bio-Grid.
2. Taswell, C. Corrections to "DOORS to the Semantic Web and Grid With a PORTAL for Biomedical Computing". *IEEE Trans. Inform. Technol. Biomed.* **2008**, *12*, 411.
3. Beck, J.R.; Adams, R.M. The cancer Biomedical Informatics Grid (caBIG): Enabling the patient-centric molecular medicine revolution. Technical report, National Cancer Institute, Bethesda, MD, USA, 2005. Slide presentation dated 10/15/2005.
4. Marshall, C.C.; Shipman, F.M. Which semantic web? HYPERTEXT '03: Proceedings of the 14th ACM Conference on Hypertext and Hypermedia; ACM: New York, NY, USA, 2003; pp. 57–66.
5. Mowshowitz, A.; Kumar, N. And Then There Were Three. *IEEE Computer* **2009**, *42*, 108–107.
6. Berti-Equille, L.; Sarma, A.D.; Xin.; Dong.; Marian, A.; Srivastava, D. Sailing the Information Ocean with Awareness of Currents: Discovery and Application of Source Dependence. Proceedings of CIDR 2009, 2009.
7. Acemoglu, D.; Ozdaglar, A.E.; ParandehGheibi, A. Spread of (Mis)Information in Social Networks. Working paper series, Social Science Research Network eLibrary, 2009.
8. Taylor, R.N.; Medvidovic, N.; Dashofy, E.M. *Software Architecture: Foundations, Theory, and Practice*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.
9. Taswell, C. PORTAL-DOORS Infrastructure System for Translational Biomedical Informatics on the Semantic Web and Grid. Proceedings of the American Medical Informatics Association Summit on Translational Bioinformatics, San Francisco, CA, 2008, p. 43.
10. Zhuge, H.; Xing, Y.; Shi, P. Resource space model, OWL and database: Mapping and integration. *ACM Trans. Internet Technol.* **2008**, *8*, 1–31.
11. Taswell, C. Alternative Bootstrapping Design for the PORTAL-DOORS Cyberinfrastructure with Self-Referencing and Self-Describing Features. In *Semantic Web*; Wu, G., Ed.; In-Teh: Vukovar, Croatia, 2009; chapter 2, pp. 29–37.
12. Eccles, J.R.; Saldanha, J.W. Metadata-based generation and management of knowledgebases from molecular biologal databases. *Comp. Meth. Prog. Bio.* **1990**, *32*, 115–123.
13. Sienknecht, T.F.; Friedrich, R.J.; Martinka, J.J.; Friedenbach, P.M. The implications of distributed data in a commercial environment on the design of hierarchical storage management. *Performance Evaluation* **1994**, *20*, 3–25.

14. Nadkarni, P.M. QAV: Querying entity-attribute-value metadata in a biomedical database. *Comp. Meth. Prog. Bio.* **1997**, *53*, 93–103.
15. de Carvalho Moura, A.M.; da Costa Pereira, G.; Campos, M.L.M. A metadata approach to manage and organize electronic documents and collections on the web. *J. Brazil. Comp. Soc.* **2002**, *8*, 16–31.
16. Shaw, N.G.; Mian, A.; Yadav, S.B. A comprehensive agent-based architecture for intelligent information retrieval in a distributed heterogeneous environment. *Decis. Sup. Sys.* **2002**, *32*, 401–415.
17. Brandt, S.A.; Miller, E.L.; Long, D.D.E.; Xue, L. Efficient Metadata Management in Large Distributed Storage Systems. Proceedings of the 20 th IEEE/11 th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS03). IEEE Computer Society, 2003.
18. Caldas, C.H.; Soibelman, L. Automating hierarchical document classification for construction management information systems. *Automat. Constr.* **2003**, *12*, 395–406.
19. Stevens, R.D.; Robinson, A.J.; Goble, C.A. myGrid: personalised bioinformatics on the information grid. *Bioinformatics* **2003**, *19*, i302–i304.
20. Arabshian, K.; Schulzrinne, H. GloServ: Global Service Discovery Architecture. MobiQuitous 2004 First Annual International Conference on Mobile and Ubiquitous Systems; IEEE Computer Society: Los Alamitos, CA, USA, 2004; pp. 319–325.
21. Cai, M.; Frank, M.; Yan, B.; MacGregor, R. A subscribable peer-to-peer RDF repository for distributed metadata management. *J. Web Semant.* **2004**, *2*, 109–130.
22. Jeong, D.; Baik, D.K. Incremental data integration based on hierarchical metadata registry with data visibility. *Information Sciences* **2004**, *162*, 147–181.
23. Zhu, Y.; Jiang, H.; Wang, J. Hierarchical Bloom filter arrays (HBA): a novel, scalable metadata management system for large cluster-based storage. CLUSTER '04: Proceedings of the 2004 IEEE International Conference on Cluster Computing; IEEE Computer Society: Washington, DC, USA, 2004; pp. 165–174.
24. Mastroianni, C.; Talia, D.; Verta, O. A super-peer model for resource discovery services in large-scale Grids. *Future Gener. Comput. Syst.* **2005**, *21*, 1235–1248.
25. Nottelmann, H.; Fischer, G. Search and browse services for heterogeneous collections with the peer-to-peer network Pepper. *Inform. Process. Manag.* **2007**, *43*, 624–642.
26. Mastroianni, C.; Talia, D.; Verta, O. Designing an information system for Grids: Comparing hierarchical, decentralized P2P and super-peer models. *Parallel Comput.* **2008**, *34*, 593–611.
27. Nguyen, D.Q.; Sonnet, H. Multilevel metadata exploration using Meta-Previewer and Cascading-View widgets. RIVF 2008 IEEE International Conference on Research, Innovation and Vision for the Future, 2008, pp. 298–305.
28. Wang, J.; Feng, D.; Wang, F.; Lu, C. MHS: A distributed metadata management strategy. *J. Syst. Software* **2009**, *82*, 2004–2011.
29. Hartung, M.; Loebe, F.; Herre, H.; Rahm, E. Management of evolving semantic grid metadata within a collaborative platform. *Information Sciences 180 (2010)* **2010**, *180*, 1837–1849.
30. Teare, K.; Popp, N.; Ong, B. Navigating network resources based on metadata. US Patent 6151624, 2000.
31. Call, C.G. Methods and apparatus for disseminating product information via the internet using universal product codes. US Patent 6154738, 2000.

32. Dixon, C.J.; Pinckney, T. Website reputation product architecture. US Patent Application 11/342319, 2006.
33. Akelbein, J.P.; Haustein, N. System and Method for Managing Data Using A Hierarchical Metadata Management System. US Patent Application 12/164197, 2009.
34. Arabshian, K.; Schulzrinne, H. A Hybrid Hierarchical and Peer-to-Peer Ontology-based Global Service Discovery System. Technical Report CUCS-016-05, Columbia Univ, New York, NY, 2005.
35. Milne, K.; Wing, P. A Brief Introduction to Enterprise Metadata Management for Microsoft SharePoint Server 2010 Developers. http://msdn.microsoft.com/en-us/library/ee832800(office.14).aspx, accessed on 28 March 2010.
36. Dolin, R.A. Pharos: A Scalable Distributed Architecture for Locating Heterogeneous Information Sources. PhD thesis, University of California Santa Barbara, 1998.
37. de Leeuw, J.; Meijer, E., Eds. *Handbook of Multilevel Analysis*; Springer Science+Business Media, LLC: New York, NY, USA, 2008.
38. Rasbash, J.; Browne, W.J. Non-Hierarchical Multilevel Models. In *Handbook of Multilevel Analysis*; Springer Science+Business Media, LLC: New York, NY, USA, 2008; chapter 8, pp. 301–334.
39. Taswell, C. Implementation of Prototype Biomedical Registries for PORTAL-DOORS. Proceedings of the American Medical Informatics Association Summit on Translational Bioinformatics, San Francisco, CA, 2009. AMIA-0036-T2009.
40. Taswell, C. Biomedical Informatics for Brain Imaging and Gene-Brain-Behavior Relationships. W3C Semantic Web HCLSIG F2F Meeting 4/30-5/1, 2009.
41. Taswell, C. Knowledge Engineering for PharmacoGenomic Molecular Imaging of the Brain. Proceedings of SKG 2009 The 5th International Conference on Semantics, Knowledge, and Grid. IEEE Computer Society, 2009, pp. 26–33.
42. Crasto, C.J., Ed. *Neuroinformatics*; Vol. 401, *Methods in Molecular Biology*, Humana Press Inc.: Totowa, NJ, USA, 2007.
43. Cheung, K.H.; Lim, E.; Samwald, M.; Chen, H.; Marenco, L.; Holford, M.E.; Morse, T.M.; Mutalik, P.; Shepherd, G.M.; Miller, P.L. Approaches to neuroscience data integration. *Brief. Bioinform.* **2009**, *10*, 345–353.
44. Berners-Lee, T.; Hall, W.; Hendler, J.; Shadbolt, N.; Weitzner, D.J. Creating a science of the Web. *Science* **2006**, *313*, 769–771.
45. Taswell, C. The Hierarchically Distributed Mobile Metadata (HDMM) Style of Architecture for Pervasive Metadata Networks. Proceedings of I-SPAN 2009 The 10th International Symposium on Pervasive Systems, Algorithms and Networks. IEEE Computer Society, 2009, pp. 315–320.
46. Fielding, R.T.; Taylor, R.N. Principled design of the modern Web architecture. *ACM Trans. Internet Technol.* **2002**, *2*, 115–150.
47. Newton, A.; Sanz, M. RFC3981: IRIS: The Internet Registry Information Service (IRIS) Core Protocol. http://www.ietf.org/rfc/rfc3981.txt, accessed on 19 October 2006.
48. Mockapetris, P.V. STD13 RFC1035: Domain Names - Implementation and Specification. http://www.ietf.org/rfc/rfc1035.txt, accessed on 19 October 2006.
49. Lockery, D.; Peters, J.F.; Taswell, C. Clinical Telegaming PORTAL for Telerehabilitation Systems. Technical report, University of Manitoba, Canada, 2010.

50. Shafer, K.; Weibel, S.; Jul, E.; Fausey, J. Introduction to Persistent Uniform Resource Locators. http://purl.oclc.org/docs/inet96.html, accessed on 4 March 2007.
51. Marchionini, G.; White, R.W. Information-Seeking Support Systems [Guest Editors' Introduction]. *IEEE Computer* **2009**, *42*, 30–32.
52. Morville, P.; Rosenfeld, L. *Information Architecture for the World Wide Web*, 3rd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2007.
53. Schraefel, M.C. Building Knowledge: What's beyond Keyword Search? *IEEE Computer* **2009**, *42*, 52–59.
54. Taswell, C. Use of the MeSH Thesaurus in the PORTAL-DOORS System. Proceedings AMIA 2010 Symposium on Clinical Research Informatics, San Francisco CA, 2010, pp. AMIA–033–C2010.
55. Taswell, C. Use of NLM Medical Subject Headings with the MeSH2010 Thesaurus in the PORTAL-DOORS System. Proceedings 8th HealthGrid 2010 Paris, 2010.
56. Newman, M.E.J. The Structure and Function of Complex Networks. *SIAM Rev.* **2003**, *45*, 167–256.
57. Fujimoto, R.M.; Perumalla, K.; Park, A.; Wu, H.; Ammar, M.H.; Riley, G.F. Large-Scale Network Simulation: How Big? How Fast? MASCOTS 2003 Proceedings of the 11th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer Systems; IEEE Computer Society: Los Alamitos, CA, USA, 2003; pp. 116–123.
58. Nicol, D.M.; Liljenstam, M.; Liu, J. Advanced concepts in large-scale network simulation. WSC '05: Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference, 2005, pp. 153–166.
59. Wei, S.; Mirkovic, J. A realistic simulation of internet-scale events. ValueTools '06: Proceedings of the 1st International Conference on Performance Evaluation Methodolgies and Tools; ACM: New York, NY, USA, 2006; p. 28.
60. Fujimoto, R.M.; Perumalla, K.S.; Riley, G.F. *Network Simulation*; Synthesis Lectures on Communication Networks, Morgan & Claypool Publishers: San Rafael, CA, USA, 2007.
61. Carl, G.; Kesidis, G. Large-scale testing of the Internet's Border Gateway Protocol (BGP) via topological scale-down. *ACM Trans. Model. Comput. Simul.* **2008**, *18*, 1–30.
62. Qiu, D.; Srikant, R. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications; ACM: New York, NY, USA, 2004; pp. 367–378.
63. Bharambe, A.R.; Herley, C.; Padmanabhan, V.N. Analyzing and Improving a BitTorrent Networks Performance Mechanisms. INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006, pp. 1 –12.
64. Deaconescu, R.; Rughinis, R.; Tapus, N. A BitTorrent Performance Evaluation Framework. ICNS '09: Proceedings of the 2009 Fifth International Conference on Networking and Services; IEEE Computer Society: Washington, DC, USA, 2009; pp. 354–358.