US 20090063522A1

(19) **United States**

(12) **Patent Application Publication**

Fay et al.

(10) **Pub. No.: US 2009/0063522 A1**

(43) **Pub. Date: Mar. 5, 2009**

(54) **SYSTEM AND METHOD FOR MANAGING ONTOLOGIES AS SERVICE METADATA ASSETS IN A METADATA REPOSITORY**

(75) Inventors: **Sharon Y. Fay**, Bay Village, OH (US); **Jay A. Holmstrom**, Cleveland, OH (US); **Bradley Carroll Wright**, St. Louis, MO (US)

Correspondence Address:
Fliesler Meyer LLP
650 California Street, 14th Floor
San Francisco, CA 94108 (US)

(73) Assignee: **ORACLE INTERNATIONAL CORPORATION**, Redwood Shores, CA (US)

(21) Appl. No.: **12/193,664**

(22) Filed: **Aug. 18, 2008**

Related U.S. Application Data

(60) Provisional application No. 60/956,614, filed on Aug. 17, 2007.

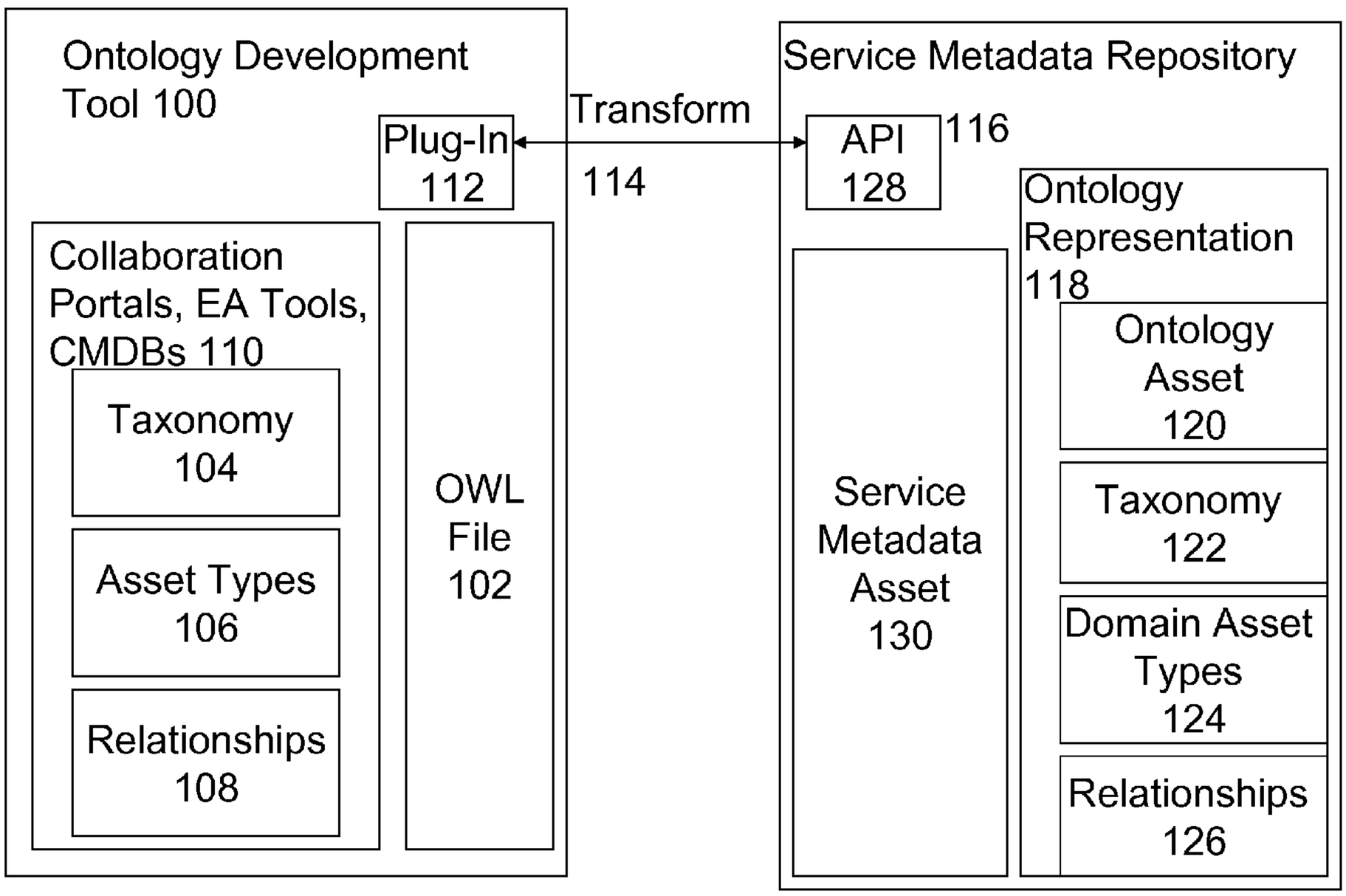
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/100**; 707/E17.044

(57) **ABSTRACT**

Embodiments of the invention are generally related to Ontologies and service metadata repositories, and particularly related to systems and methods for transforming Ontologies into service metadata assets in a service metadata repository. One embodiment includes a plug-in from an ontology development tool communicating to an application programming interface for a service metadata repository. One embodiment includes transforming an ontology from an ontology web language file in an ontology development tool into a service metadata asset in the service metadata repository.



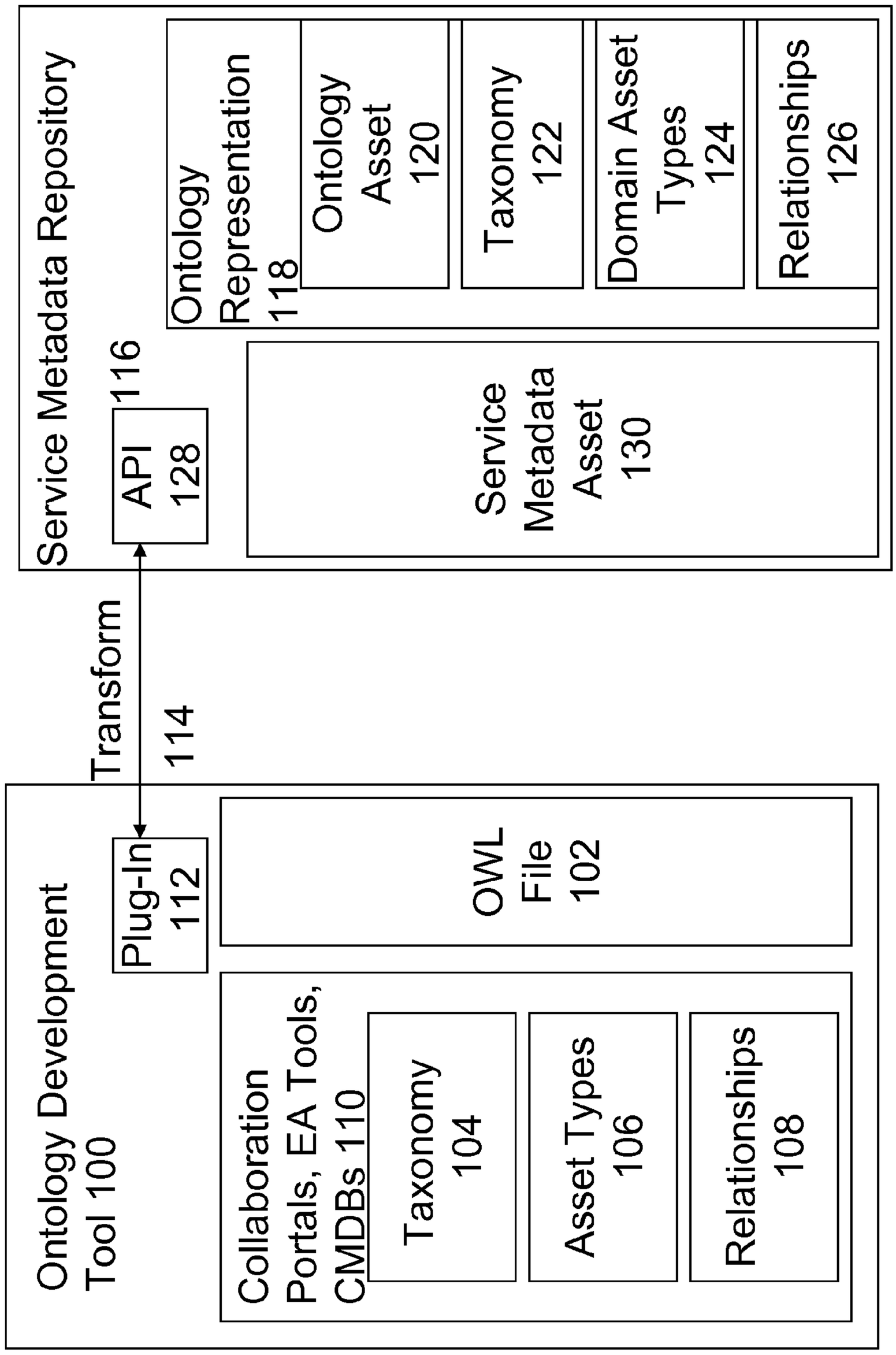

Ontology Development Tool 100
Collaboration Portals, EA Tools, CMDBs 110
Taxonomy 104
Asset Types 106
Relationships 108
Plug-In 112
OWL File 102
Transform 114
Service Metadata Repository 116
API 128
Service Metadata Asset 130
Ontology Representation 118
Ontology Asset 120
Taxonomy 122
Domain Asset Types 124
Relationships 126

FIG. 1

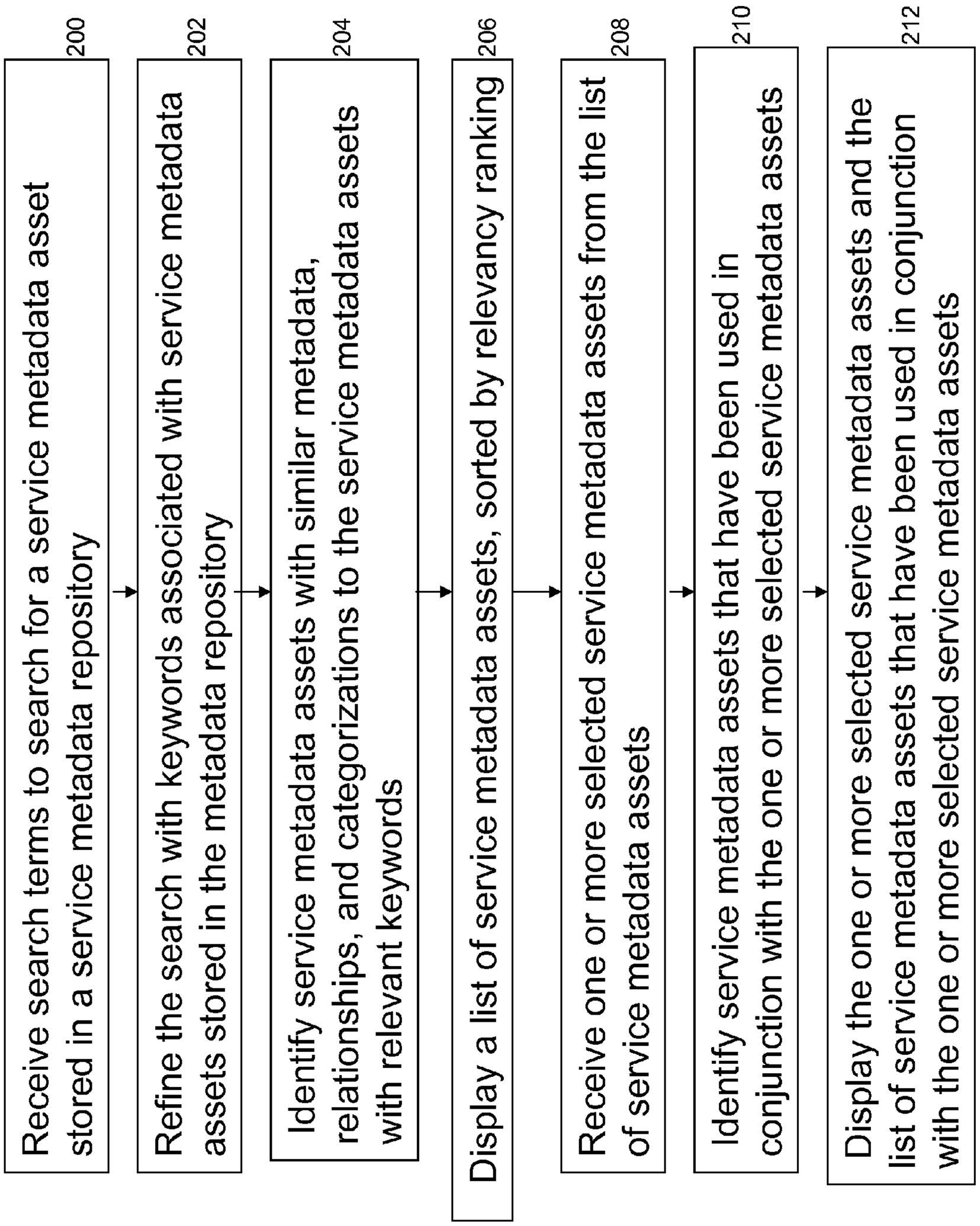

Receive search terms to search for a service metadata asset stored in a service metadata repository
200
Refine the search with keywords associated with service metadata assets stored in the metadata repository
202
Identify service metadata assets with similar metadata, relationships, and categorizations to the service metadata assets with relevant keywords
204
Display a list of service metadata assets, sorted by relevancy ranking
206
Receive one or more selected service metadata assets from the list of service metadata assets
208
Identify service metadata assets that have been used in conjunction with the one or more selected service metadata assets
210
Display the one or more selected service metadata assets and the list of service metadata assets that have been used in conjunction with the one or more selected service metadata assets
212

FIG. 2

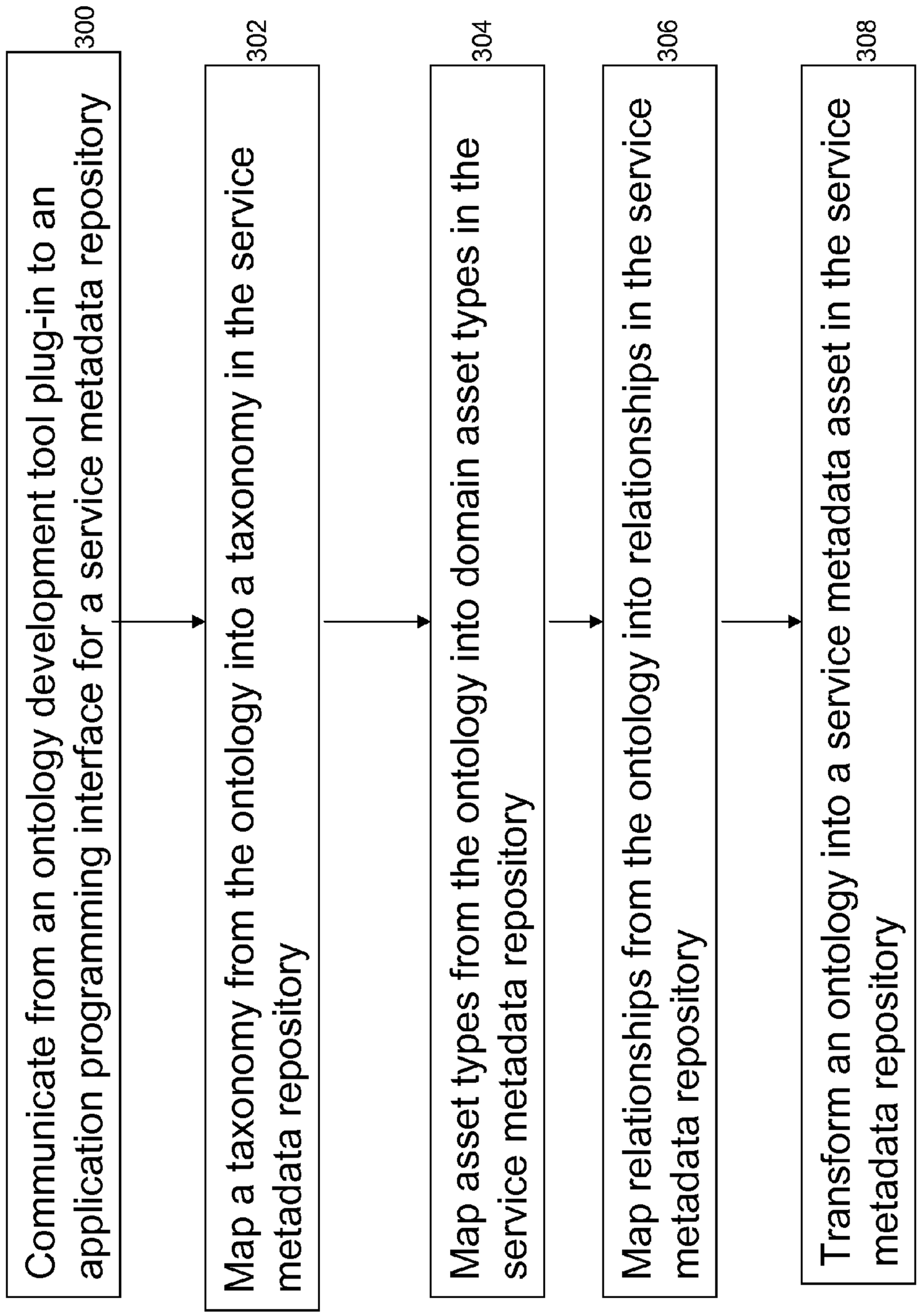

Communicate from an ontology development tool plug-in to an application programming interface for a service metadata repository
300
Map a taxonomy from the ontology into a taxonomy in the service metadata repository
302
Map asset types from the ontology into domain asset types in the service metadata repository
304
Map relationships from the ontology into relationships in the service metadata repository
306
Transform an ontology into a service metadata asset in the service metadata repository
308

FIG. 3

SYSTEM AND METHOD FOR MANAGING ONTOLOGIES AS SERVICE METADATA ASSETS IN A METADATA REPOSITORY

CLAIM OF PRIORITY

[0001] The present application claims the benefit of priority under 35 U.S.C. §119(e) to U.S. Provisional Patent Application No. 60/956,614 entitled "SYSTEM AND METHOD FOR SEMANTIC ASSET SEARCH IN A METADATA REPOSITORY," filed on Aug. 17, 2007, which application is incorporated herein by reference.

CROSS REFERENCE TO RELATED APPLICATION

[0002] The following U.S. patent application is cross-referenced and incorporated herein by reference:

[0003] U.S. patent application Ser. No. 12/193,651 entitled "SYSTEM AND METHOD FOR SEMANTIC ASSET SEARCH IN A METADATA REPOSITORY," filed Aug. 18, 2008.

COPYRIGHT NOTICE

FIELD OF THE INVENTION

[0005] This invention is related to Semantic Technology and Metadata Repositories in the Information Technology field, particularly with regard to utilizing semantic technology to search for assets in a metadata repository.

BACKGROUND

[0006] Service-Oriented Architecture (SOA) is based on the deconstruction of yesterday's monolithic applications and information technology infrastructure into a matrix of discrete, standards-based, network-accessible services. The process of transformation requires the organization, identification, and repurposing of applications and business processes of the existing information technology (IT) infrastructure. The transformation to SOA begins with an analysis of the IT infrastructure to identify applications, business processes, and other software assets that become services, or otherwise support the SOA.

[0007] Metadata is data about data, or more specifically, information about the content of the data; service metadata is information about the services in an SOA. Service producers use service metadata to describe what service consumers need to know to interact with the service producers. Service metadata is stored in a metadata repository by service producers and then accessed by service consumers. A metadata repository provides visibility into the portfolio of assets, the traceability of the assets within that portfolio, the relationships and interdependencies that connect the assets to each other, the policies that govern use of the assets, and the projects that produce the assets and consume the assets.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. **1** shows a system for managing ontologies produced by ontology development tools as assets in a service metadata repository.

[0009] FIG. **2** shows one embodiment of a method for performing a semantic asset search in a metadata repository.

[0010] FIG. **3** shows a method for transforming an ontology in an ontology development tool into a service metadata asset in a service metadata repository.

DETAILED DESCRIPTION

[0011] Service-Oriented Architecture (SOA) is a new approach to information technology that connects people, process, and technology in a dynamic, distributed environment. Although SOA provides the agility required to innovate and compete in today's economy, it also increases system complexity. To mitigate this risk, organizations control and track information technology investments to ensure alignment with corporate objectives. A service metadata repository enables SOA governance that provides comprehensive insight into the business impact of SOA. A service metadata repository can enable SOA governance to span the SOA lifecycle and unite resources from across divisions and geographies in a collaborative holistic approach to corporate decision-making and compliance by providing the automated exchange of metadata and service information among service consumers, providers, policy decision points, and other governance tools.

[0012] A service metadata repository provides role-based visibility into all SOA assets, regardless of source, through a centralized repository for business processes, services, applications, components, models, frameworks, policies, and data services. Visibility into assets under development minimizes redundancy and promotes service collaboration and reuse. A service metadata repository could also graphically display and navigate asset-to-asset and asset-to-project relationships and interdependencies to simplify impact analysis and ensure business alignment by enabling users to organize and link SOA assets to associated business processes.

[0013] Semantic technology encodes meanings separately from data and content files, and separately from application code. Semantic technologies include tools for auto-recognition of topics and concepts, information and meaning extraction, and categorization. Semantic technologies provide an abstraction layer above existing technologies that enables interconnection of data, content, and processes.

[0014] In order to enhance reuse of service metadata in a service metadata repository, users need to be able to search for service metadata assets. What is needed is a way to perform a semantic asset search on service assets in a service metadata repository.

[0015] One embodiment is a system that manages ontologies as assets in a service metadata repository. FIG. **1** shows an architecture for managing ontologies produced by development tools as service metadata assets in a service metadata repository. Ontology development tools **100** include tools that can produce a web ontology language (OWL) file **102**, including Collaboration Portals, Enterprise Architecture tools, and Configuration Management Databases **110**. In order to develop an ontology using OWL, a taxonomy **104**,

asset types **108**, and relationships **106** need to be identified. A plug-in **112** in the ontology development tool **100** communicates to an API **128** in the service metadata repository **116** in order to automatically transform **114** an OWL file **102** into a service metadata repository asset **130** stored in the service metadata repository **116**. The process further requires mapping the OWL taxonomy **104** into a service metadata repository taxonomy **122**. Finally, the process maps the OWL asset type **108** into a domain asset type **126** and maps the OWL relationships **106** into metadata repository relationships **124**.

[0016] In one embodiment, a data transformation utility transforms **114** an OWL file **102** into a format for storing in the service metadata repository **116** as a service metadata asset **130**. In one embodiment, the data transformation utility is part of the plug-in **112**. In one embodiment, the data transformation utility is part of the API **128**. In one embodiment, the data transformation utility is separate from the plug-in **112** and the API **128**. In one embodiment, the API **128** is a service metadata repository extensibility framework API that provides programmatic access to service metadata repository functionality.

[0017] One embodiment comprises a method for performing a semantic asset search in a service metadata repository, shown in FIG. **2**. In step **200**, the service metadata repository receives search terms for an asset stored in the metadata repository. In step **202**, the service metadata repository refines the search with keywords associated with assets stored in the metadata repository. In step **204**, the service metadata repository identifies assets with similar metadata, relationships, and categorizations to the assets with the most relevant keywords. In step **206**, the service metadata repository displays a list of assets, sorted by relevancy ranking. In step **208**, the service metadata repository receives one or more selected service metadata assets from the list of service metadata assets. In step **210**, the service metadata repository identifies service metadata assets that have been used in conjunction with the one or more selected assets. In step **212**, the service metadata repository displays the one or more selected assets and the list of assets that have been used in conjunction with the one or more selected assets.

[0018] One embodiment comprises a method for transforming an ontology into an asset in a service metadata repository, shown in FIG. **3**. In step **300**, a data transformation utility communicates information from an ontology development tool plug-in to an application programming interface for a service metadata repository. In step **302**, the data transformation utility maps a taxonomy from an ontology in the ontology development tool into a taxonomy in the service metadata repository. In step **304**, the data transformation utility maps asset types from the ontology in the ontology development tool into domain asset types in the service metadata repository. In step **306**, the data transformation utility maps relationships from the ontology in the ontology development tool into relationships in the service metadata repository. In step **308**, the data transformation utility transforms an ontology into a service metadata asset in the service metadata repository.

[0019] A metadata repository is the enterprise system of record for the software asset portfolio. The metadata repository federates metadata from individual registries and repositories. It manages all software assets and their relationships, including business processes, applications, patterns, frameworks, services and components (corporate, packaged or open source). The metadata repository has links to content in other systems (source code management systems, project portfolio management systems, document management systems, etc.) and exposes a web-service-based API for integration. The metadata repository promotes enterprise visibility and access. In one embodiment, additional features include web-based accessibility with role-based access control, a "transparent" development experience, and project portfolio management integration.

[0020] The metadata repository enables governance and compliance initiatives within an organization. A metadata repository enables automated tracking of compliance with architecture standards and regulatory requirements. A metadata repository provides for prescriptive reuse and policy, validation, certification and approvals for registered content.

[0021] In one embodiment, the metadata repository provides an analytics capability, wherein users can automatically calculate the value of the software asset portfolio, savings from use and reuse of assets, software investment return on investment (ROI), and impact analysis for IT planning decisions.

[0022] In one embodiment, the metadata repository is an enterprise repository that links multiple repositories together. The metadata repository integrates with project portfolio management systems such as CA Clarity™ and Mercury ITG. The metadata repository brings visibility of the asset portfolio and aligns it with the project portfolio. The metadata repository integrates with document management systems because frequently documentation, requirements, etc. are managed within document management repositories but need to be accessed from the asset entry within the metadata repository. The metadata repository integrates with source code management systems for code and artifacts, with build tools to harvest assets and update the status of assets. The metadata repository captures business processes and metadata through Business Process Execution Language (BPEL) introspection. The metadata repository supports the Application Programming Interfaces (APIs) for Universal Description, Discovery and Integration (UDDI) for interoperability. For organizations that have UDDI registries, the metadata repository inter-operates with UDDI registries. The metadata repository can sit on top of Systinet or any other UDDI-compliant registry and bring service metadata into the metadata repository where it can be augmented and tied into the balance of the information technology ecosystem. One embodiment of the metadata repository integrates with modeling tools including Microsoft Visio®. One embodiment integrates with modeling tools such as Aris™, Popkin, and Troux.

[0023] In one embodiment, the metadata repository provides greater capabilities than just storing metadata. The metadata repository functions as a software asset management platform. It enables organizations to improve governance of the asset portfolio, enable compliance with both enterprise architecture and project-specific objectives, and provide a platform to reduce and manage complexity to lower costs and increase business agility.

Search

[0024] Semantic technology can search across multiple instances of a metadata repository, i.e. federated repositories. This technology can serve as a "corporate dictionary" or "global thesaurus" for a metadata repository, such that portfolio managers can select relevant asset keywords for their

assets/projects, and users can find and reuse existing functionality, regardless of their search terminology.

[0025] Asset lifecycle tracking can be enhanced by semantic technology by searching across multiple repositories (metadata repositories, source code management systems, CMDBs, enterprise architecture repositories, metadata repositories, etc.), finding information about the same asset in planning, development, and deployed states.

[0026] Searching for artifacts associated with assets can be enabled by semantic technology in order to identify relevant assets. Examples of artifacts include design documents, requirements documents, development coding guidelines, etc.

[0027] A set of secondary search results related to the initial search criteria can be produced by semantic technology. This is similar to an extended affinity.

Searching Federated Repositories

[0028] Organizations may deploy multiple instances of a metadata repository as single purpose repositories. For example, there might be a separate instance for each organizational domain, or one instance might manage the organization's application portfolio, while another instance might be externally facing to provide services to third parties. In such instances, the ability to search across multiple instances of the metadata repository and return an aggregated search result set is desirable.

[0029] Users may only be authorized to view subsets of assets within federated repositories. User permissions may need to be identified through advanced role-based access control (RBAC) settings. Only valid users of the system with the appropriate permissions can enter asset search criteria. The user is then presented with an aggregated set of search results with links to the appropriate asset details within each repository. The user can then activate the search result links to access asset details.

Corporate Dictionary/Global Thesaurus

[0030] One of the reasons that organizations fail to identify existing functionality is because there is not a common vocabulary used to describe and categorize existing assets. This causes organizations to build redundant functionality, resulting in increased development, maintenance, and operations costs. Including a corporate dictionary or global thesaurus in a metadata repository will solve this problem. In the entry of asset metadata, project metadata or categorization entries, the dictionary/thesaurus supplies words to any drop-down menus in metadata templates to ensure consistency in terminology throughout the metadata repository. Consideration should be given to this functionality on automatic import of asset metadata from another system of record.

[0031] When the metadata repository is subsequently searched, the search keywords should resolve toward words included in the dictionary/thesaurus as follows. Users enter keywords into the search. If the keyword matches something in the dictionary/thesaurus, then the metadata repository presents asset or project hits on the dictionary entry. If the keyword does not match something in the corporate dictionary, then the system prompts the user to choose which word(s) in the dictionary most closely match the keyword. As an alternative, the system could automatically select the words that most closely match.

[0032] Only those users with permission to edit asset metadata/project information in the metadata repository would be able to select relevant keywords for assets/projects. Any user would be able to search the metadata repository leveraging the dictionary/thesaurus. The search criteria may not be an exact match of the terms included in the asset name, description, or keywords. The result is that the user is presented with a set of search results and relevancy ratings, sorted by relevancy rating, and including links to the appropriate asset details. The user can then activate the search result links to access asset details.

Asset Lifecycle Tracking

[0033] Throughout its lifecycle, information about an asset may be represented in multiple tools and repositories across an organization. For example, asset planning information might appear in an architecture tool, a visual modeling tool, a business process management tool, within documents in a document management system, etc. During development, information about an asset might appear in integrated development environments (IDEs), version control and source control management systems (SCMs), testing tools, etc. Once deployed, information about an asset might appear in a service registry, a service bus, in web service management/monitoring (WSM) tools, in configuration management databases (CMDBs), in defect/change management systems, etc. Information about an asset may appear in the metadata repository at any point in its lifecycle. There is a need to identify all references to an asset, regardless of the reference location.

[0034] The most likely users are asset portfolio managers that will harvest and package the lifecycle information from these various sources into a single asset reference within the metadata repository. When a valid user of the system with the appropriate permissions enters asset search criteria, the user is presented with an aggregated set of search results with links to the appropriate asset details within each tool/repository. Users with permission to create/edit assets in the metadata repository will have the option of aggregating this information into a single asset reference within the metadata repository.

Searching Related Artifacts

[0035] Search terms entered by a user may appear in documents or links associated with the asset, rather than in the asset's metadata or keywords. Examples of artifacts include design documents, requirements documents, architecture diagrams, development coding guidelines, etc. It is necessary to search these associated files and links in order to return a more robust and relevant search result set. A valid user of the system with the appropriate permissions enters asset search criteria and the system searches asset metadata and associated asset artifacts (represented as links in the asset metadata). The user is subsequently presented with an aggregated set of Asset search results with links to the appropriate asset details. The user can then activate the search result links to access asset details.

Affinity

[0036] In cases where a user is unfamiliar with assets in the metadata repository, it is helpful not only to offer up sets of assets that directly match a user's search criteria, but also present secondary sets that might be of interest. In one

embodiment, the metadata repository identifies assets that other people have used in conjunction with the primary asset of interest. In one embodiment, the metadata repository produces a set of extended search results, such as a set of assets with similar metadata, common relationships, the same categorizations, etc. This is a "suggestive selling" approach for asset consumers, and it also assists asset portfolio managers in analyzing relationships within their asset portfolios. This enables synergy in the asset portfolio and across asset portfolios. A valid user of the system with the appropriate permissions enters asset search criteria, the system searches asset metadata (including metadata fields, relationships, categorizations, and associated artifacts) to identify a secondary set of search results with an affinity to the primary search results, and the user is presented with a primary and secondary set of asset search results with links to the appropriate asset details within the metadata repository. The user can then activate the search result links to access asset details.

Intelligence and Automation

[0037] Automated semantic technology can assist asset portfolio managers, domain experts, etc. in identifying redundant functionality and providing an automated solution for the asset organization tasks necessary to resolve a problem. Redundant functionality may exist within or outside of the metadata repository. Examples include: identifying that two or more assets deliver similar functionality and suggesting consolidation; and identifying commonalities among multiple sets of project requirements as opportunities to develop or harvest an asset to meet the multiple sets of project requirements.

[0038] Relationships between assets can be revealed by semantic technology. Categorizations and new taxonomies for groups of assets are easier to identify. This technology can also provide an ad hoc rollup of related asset information (such as metadata from enterprise repositories, documents stored in a document management system, code stored in a version control system, defects and change requests in a defect tracking system, etc.), and provide an option to automatically package related information into a single asset in the metadata repository.

[0039] Assets in the metadata repository that might be relevant to a project, based on the project's requirements, can be identified. Requirements might be articulated through a project profile in the metadata repository, a requirements document, a requirements management system, etc. Project profiles can be built based on pre-defined information. A repository "clean-up" function can be provided, suggesting assets within the metadata repository that should be retired or decommissioned.

Asset Consolidation

[0040] Redundant functionality results in increased maintenance and operations costs. Redundancy might occur as early as project planning, at which time common requirements and opportunities to develop common functionality go unrecognized. Organizations rely upon asset portfolio managers, domain experts, architects, etc. to identify redundant functionality and suggest consolidation opportunities.

[0041] During project planning, redundant requirements might appear in project portfolio management tools, requirements management tools, etc. During development, redundant functionality might appear in integrated development environments (IDEs), source code management, version control and source control management systems (SCMs), testing tools, etc. Once deployed, redundant functionality might appear in a service registry, a service bus, in web service management/monitoring (WSM) tools, in configuration management databases (CMDBs), in defect/change management systems, etc. Redundant functionality may also appear in the metadata repository. An automated method for identifying redundant functionality and suggesting consolidation opportunities would be of great benefit.

[0042] Typical users are asset portfolio managers, domain experts, architects, project portfolio managers, operations managers, etc. The users can schedule the redundancy identification function to run on automatic intervals. The users are then presented with a list of redundant functionality/opportunities for consolidation. The users can select groups of assets to consolidate, which creates a "proposed" asset in the metadata repository with relationships to all of the redundant functionality to be replaced. The users can identify assets that are not consolidation targets, and these assets can be removed from future result sets.

Identify Asset Relationships

[0043] As the metadata repository gathers assets from software applications, such as enterprise service buses, data service platforms, service registries, business process management systems, etc., there will be a need to identify the way in which these assets relate to each other, and to allow a user to easily establish these relationship links in the metadata repository.

[0044] The typical users are asset portfolio managers who will need to establish meaningful relationships among assets in the metadata repository. The users can schedule the relationship identification function to run on automatic intervals. Users are presented with a list of assets and suggested relationships. The users select the desired asset relationships. Users can identify asset relationships that are not meaningful, and those asset relationships can be removed from future result sets.

Identify Asset Categorizations and Taxonomies

[0045] As the metadata repository gathers assets from software applications, such as enterprise service buses, data service platforms, service registries, business process management systems, etc., there will be a need to organize these assets into meaningful taxonomies within the metadata repository. There will also be a need to suggest new taxonomies and organizational constructs as the inventory of assets grows.

[0046] The most likely users are asset portfolio managers who will need to organize their assets in the metadata repository. Users may schedule the categorization/taxonomy identification function to run on automatic intervals. Users are presented with a list of assets and suggested categorizations or new taxonomies. Users may select the desired asset categorizations/new taxonomy. Users may identify asset categorizations/taxonomies that are not meaningful, and those asset categorizations/taxonomies can be removed from future result sets.

Automated Packaging

[0047] An asset in the metadata repository consists of metadata and supporting documentation (which is stored in other

systems and appears as a Universal Resource Indicator (URI) in the metadata repository). If semantic technology can be used to identify all references to instances of the same asset, and present these to the metadata repository, then the user should have the option to generate an asset in the metadata repository that populates the metadata from these other sources and creates links to all related information and instances of the asset.

[0048] Typical users are asset portfolio managers, domain experts, etc. Users may schedule the search function to run on automatic intervals. Users are presented with a list of information pertaining to an asset. Users may select the information that will populate the asset metadata and be linked to the asset. Users may identify information that is not relevant. This information should be removed from future result sets.

Asset Suggestion

[0049] Project requirements might be captured in a project portfolio management tool, a project management tool, a requirements management tool, in a project requirements document within a document management system, as metadata in a project profile in the metadata repository, etc. There is a need to automatically identify sets of assets in the metadata repository based on a project's requirements, to provide the user with a means of selecting relevant assets, and to package selected assets into the metadata repository project profile.

[0050] Typical users are project architects, business analysts, and project technical leads. Users identify the source of the project requirements. Users are presented with a list of assets in the metadata repository that may meet the project's requirements. Users select the assets that are relevant. Relevant assets are added to the metadata repository project profile, if the profile already exists. A new project profile is created in the metadata repository if there is no pre-existing project profile.

Asset Cleanup

[0051] In order to keep the metadata repository populated with the most relevant sets of assets, the asset inventory should be periodically examined and cleansed of assets that are not being used, assets that have poor user ratings, assets that have incomplete metadata, etc. The most likely users are asset portfolio managers. The users may schedule the clean-up function to run on automatic intervals. The users are presented with a list of assets in the metadata repository that are candidates for cleansing. The users select the assets and determine a course of action. Courses of action may include: Retirement—moved to a "retired" state in the metadata repository and asset users are notified. Deletion—moved to a "deleted" state in the metadata repository. Enhancement—asset remains at the current version and additional metadata, documentation, etc. is added. Reengineering—a new version of the asset is created in the metadata repository in a "proposed" state.

Ontology Definitions

[0052] Collection: Any grouping of items into a single container. The metadata repository equivalent is a Taxonomy.

[0053] Class: A set of things with common characteristics. The metadata repository equivalent is an AssetType.

[0054] Instance: Occurrence or specific member of a class. The metadata repository equivalent is an Asset.

[0055] Ontology: A formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. Ontologies are used in artificial intelligence, the Semantic Web, software engineering, biomedical informatics, library science, and information architecture as a form of knowledge representation about the world or some part of it

[0056] Subclass or Slot: Properties of each class describing various features and attributes of the class. The metadata repository equivalent is a Metadata Field.

[0057] Ontology Asset Type—Metadata Fields. Metadata Fields describe the domain covered by the ontology, who uses the ontology, and who maintains the ontology.

Ontologies

[0058] A configured instance of the metadata repository represents one or more ontologies. There is a programmatic mapping between the OWL Web Ontology Language and the metadata repository XML formats. A data transformation utility can be written to convert between formats. Ontologies can be developed in other tools, such as collaborative portals, enterprise architecture tools, CMDBs, etc. If these tools use OWL, then ontologies generated by those tools can be shared between the metadata repository and such tools. Ontologies (represented in OWL or other file formats) can be managed as assets within the metadata repository. An ontology is defined as an explicit formal specification of the terms in the domain and relations among them.

Managing OWL Files in a Metadata Repository

[0059] The metadata repository manages OWL files. Managing OWL files comprises: managing the lifecycle of OWL files, searching for and evaluating OWL files as reuse candidates, tracking OWL file usage within and external to an organization, tracking the relationship between OWL files and other assets, and collaborating on the concepts, properties, and relationships in a domain that will be captured in an OWL file. The goal is to promote visibility, accessibility, traceability, notification, and collaboration. The approach is to build an OWL asset type in the metadata repository, including relevant taxonomies and relationships.

Deriving Domain Metamodels from OWL files

[0060] Domain meta-models (consisting of taxonomies and common domain relationships) are captured in an inconsistent format within various tools throughout an organization (project portfolio management systems, configuration management databases, version control systems, financial accounting systems, metadata repositories, enterprise architecture repositories, etc.). Typically, there is no easy means of sharing taxonomies between tools and repositories. Therefore, there is a need to use domain meta-models as an organizing principal for managing assets.

[0061] OWL can be used as a standard for sharing domain taxonomies and commonly defined relationships between tools and repositories. The metadata repository ontology may be modified in several ways—through the metadata repository API, through an ontology editor such as Knoodl, through the metadata repository administrative functions—so the ontology asset in the metadata repository might not match the actual ontology representation in the metadata repository. In one embodiment, architecture tools, CMDB tools, etc. generate OWL files. One embodiment uses Semantic Annotation

for WSDL (SAWSDL). One embodiment uses Simple Knowledge Organizational Systems (SKOS) instead of OWL. SKOS is a set of specifications and standards to support the use of knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web. SKOS is developed and maintained by the W3C Semantic Web Deployment Working Group.

[0062] One embodiment may be implemented using a conventional general purpose computer or a specialized digital computer or microprocessor(s) programmed according to the teachings of the present disclosure, as will be apparent to those skilled in the computer art. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art. The invention may also be implemented by the preparation of integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

[0063] One embodiment includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the features present herein. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, micro drive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, flash memory of media or device suitable for storing instructions and/or data stored on any one of the computer readable medium (media). The present invention can include software for controlling both the hardware of the general purpose/specialized computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the present invention. Such software may include, but is not limited to, device drivers, operating systems, execution environments/containers, and user applications.

[0064] Embodiments of the present invention can include providing code for implementing processes of the present invention. The providing can include providing code to a user in any manner. For example, the providing can include transmitting the code to a user; providing the code on a physical media to a user; or any other method of making the code available.

[0065] Embodiments of the present invention can include a computer-implemented method for transmitting code which can be executed at a computer to perform any of the processes of embodiments of the present invention. The transmitting can include transfer through any portion of a network, such as the Internet; through wires, or any other type of transmission. The transmitting can include initiating a transmission of code, or causing the code to pass into any region or country from another region or country. For example, transmitting includes causing the transfer of code through a portion of a network as a result of previously addressing and sending data including the code to a user. A transmission to a user can include any transmission received by the user in any region or country, regardless of the location from which the transmission is sent.

[0066] Other features, aspects and objects of the invention can be obtained from a review of the figures and the claims. It is to be understood that other embodiments of the invention can be developed and fall within the spirit and scope of the invention and claims. The foregoing description of preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

1. A method for transforming an ontology in an ontology development tool into a service metadata asset in a service metadata repository, comprising:

communicating from an ontology development tool plug-in to an application programming interface for a service metadata repository;

mapping a taxonomy from an ontology in the ontology development tool into a taxonomy in the service metadata repository;

mapping asset types from the ontology in the ontology development tool into domain asset types in the service metadata repository;

mapping relationships from the ontology in the ontology development tool into relationships in the service metadata repository; and

transforming an ontology into a service metadata asset in the service metadata repository.

2. The method of claim **1**, wherein a configured instance of the service metadata repository represents one or more ontologies.

3. The method of claim **1**, wherein there is a programmatic mapping between the web ontology language and extensible markup language formats supported by the service metadata repository.

4. The method of claim **1**, wherein a data transformation utility converts files between OWL and XML.

5. The method of claim **1**, wherein the service metadata repository manages the lifecycle of one or more Ontologies.

6. The method of claim **1**, wherein the service metadata repository searches for and evaluates Ontologies as reuse candidates.

7. The method of claim **1**, wherein the service metadata repository tracks Ontology usage within and external to an organization.

8. The method of claim **1**, wherein the service metadata repository tracks relationships between web ontology language files and other service metadata assets.

9. The method of claim **1**, wherein the service metadata repository uses domain meta-models as an organizing principal for managing assets.

10. A computer-readable storage medium, including instructions stored thereon which when read and executed by a computer cause the computer to perform steps comprising:

communicating from an ontology development tool plug-in to an application programming interface for a service metadata repository;

mapping a taxonomy from an ontology in the ontology development tool into a taxonomy in the service metadata repository;

mapping asset types from the ontology in the ontology development tool into domain asset types in the service metadata repository;

mapping relationships from the ontology in the ontology development tool into relationships in the service metadata repository; and

transforming an ontology into a service metadata asset in the service metadata repository.

11. The computer-readable storage medium of claim **10**, wherein a configured instance of the service metadata repository represents one or more ontologies.

12. The computer-readable storage medium of claim **10**, wherein there is a programmatic mapping between the web ontology language and extensible markup language formats supported by the service metadata repository.

13. The computer-readable storage medium of claim **10**, wherein a data transformation utility converts files between OWL and XML.

14. The computer-readable storage medium of claim **10**, wherein the service metadata repository manages the life-cycle of one or more Ontologies.

15. The computer-readable storage medium of claim **10**, wherein the service metadata repository searches for and evaluates Ontologies as reuse candidates.

16. The computer-readable storage medium of claim **10**, wherein the service metadata repository tracks Ontology usage within and external to an organization.

17. The computer-readable storage medium of claim **10**, wherein the service metadata repository tracks relationships between web ontology language files and other service metadata assets.

18. The computer-readable storage medium of claim **10**, wherein the service metadata repository uses domain meta-models as an organizing principal for managing assets.

19. A system comprising:

a service metadata repository;

a plug-in to an ontology development tool;

an application programming interface that exposes functionality for the service metadata repository; and

a data transformation utility that transforms web ontology language files into extensible markup language.

* * * * *