Flynn Allen J. (Orcid ID: 0000-0002-3471-3063)

The Knowledge Object Reference Ontology (KORO): A Formalism to Support Management and Sharing of Computable Biomedical Knowledge for Learning Health Systems

Allen J. Flynn, PharmD[1,2], Charles P. Friedman, PhD[1,2,3],
Peter Boisvert, BA, MFA[2], Zachary Landis-Lewis, PhD[2], Carl Lagoze, PhD[1]
[1]School of Information; [2]School of Medicine; [3]School of Public Health,
University of Michigan, Ann Arbor, MI

Corresponding Author:

Allen J. Flynn, PharmD
Division of Learning and Knowledge Systems
Department of Learning Health Sciences
Medical School
University of Michigan
T: 734.615.0839
E: ajflynn@med.umich.edu

Article type: Technical Report

Word counts:

Abstract (Including the word 'Abstract'): 249
Body (Introduction through Acknowledgements): 7051
Total: 7300

Abstract

Introduction

Health systems are challenged by care underutilization, overutilization, disparities, and related harms. One problem is a multi-year latency between discovery of new best practice knowledge and its widespread adoption. Decreasing this latency requires new capabilities to better manage and more rapidly share biomedical knowledge in computable forms. Knowledge Objects package machine-executable knowledge resources in a way that easily enables knowledge-as-a-service. To help improve knowledge management and accelerate knowledge sharing, the Knowledge Object Reference Ontology (KORO) defines what Knowledge Objects are in a formal way.

Methods

Development of KORO began with identification of terms for classes of entities and for properties. Next, we established a taxonomical hierarchy of classes for Knowledge Objects and their parts. Development continued by relating these parts via formally defined properties. We evaluated the logical consistency of KORO and used it to answer several competency questions about parthood. We also applied it to guide Knowledge Object implementation.

Results

As a realist ontology, KORO defines what Knowledge Objects are and provides details about the parts they have and the roles they play. KORO provides sufficient logic to answer several basic but important questions about Knowledge Objects competently. KORO directly supports creators of Knowledge Objects by providing a formal model for them.

Conclusion

KORO provides a formal, logically consistent ontology about Knowledge Objects and their parts. It exists to help make computable biomedical knowledge findable, accessible, interoperable, and reusable. KORO is currently being used to further develop and improve computable knowledge infrastructure for Learning Health Systems.

Keywords

Ontology, Knowledge Management, Knowledge Object, KORO, BFO, IAO

Introduction

A movement to create Learning Health Systems (LHSs) is ongoing[1-3]. One important goal of a LHS is to greatly decrease the current latency of getting knowledge from bench to bedside, accelerating its movement from discovery into practice. This paper describes a formal, logically-consistent knowledge object reference ontology to improve management, enable archiving, and accelerate sharing of actionable, machine-executable biomedical knowledge.

We postulate that the LHS "learns" through a cyclical process that engages a community in empirical analysis of data relevant to a biomedical problem, which leads to discovery of new knowledge. The **learning cycle** is completed by direct application of that knowledge to change practice[3,4]. Changed practice generates new data, driving the next iteration of the cycle, with improvement occurring via successive iterations. To support learning cycles at scale, the LHS requires a shared technology infrastructure-platform to support three information flows: (1) **data-to-knowledge**, (2) **knowledge-to-practice**, and (3) **practice-to-data** (Figure 1).

We are building an LHS infrastructure-platform called the **Knowledge Grid** (www.kgrid.org)[5]. It specifically supports the knowledge-to-practice flow spanning steps 5, 6, and 7 in Figure 1. The Knowledge Grid (**KGrid**) is built upon mature and widely used technologies[6]. It is for managing and rapidly sharing machine-executable (i.e., "computable") biomedical knowledge. KGrid allows its users to create, manage, and steward digital **Knowledge Objects**[7], which are structured packages holding instances of *computable biomedical knowledge*. KGrid includes a digital **Library** component, to store and manage Knowledge Objects[5], and a separate **Activator** component, to enable remote invocation of the computable knowledge they hold. In this way, KGrid's **Activator** provides knowledge-as-a-service. Specifically, instances of the **Activator** "activate" computable knowledge by (a) loading Knowledge Objects stored as digital files, (b) providing a means to execute the computable knowledge held in those files, (c) making those means of execution available to external systems via webservices, and, (d) tracking the utilization of those webservices.

This paper describes the design, development, and initial use of the Knowledge Object Reference Ontology (KORO) in support of KGrid. KORO extends the Information Artifact Ontology (IAO)[8] to formally define what are Knowledge Objects (KOs). Hence, although motivated by the need to more readily share biomedical knowledge,

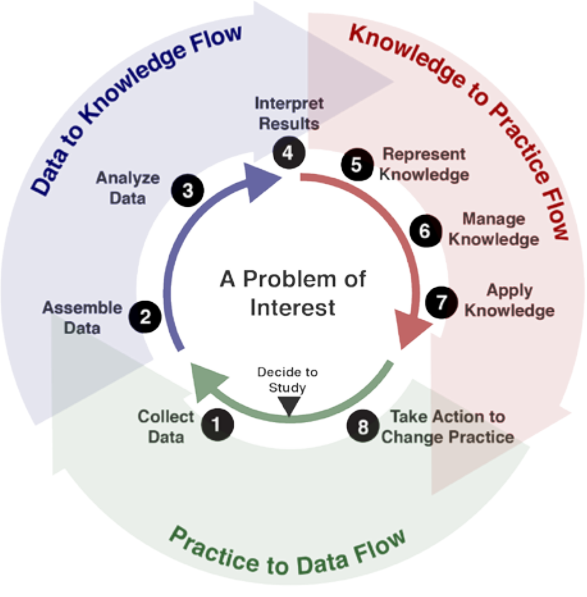

Figure 1. The Learning Health Cycle of the LHS with Three Information Flows and Eight Steps

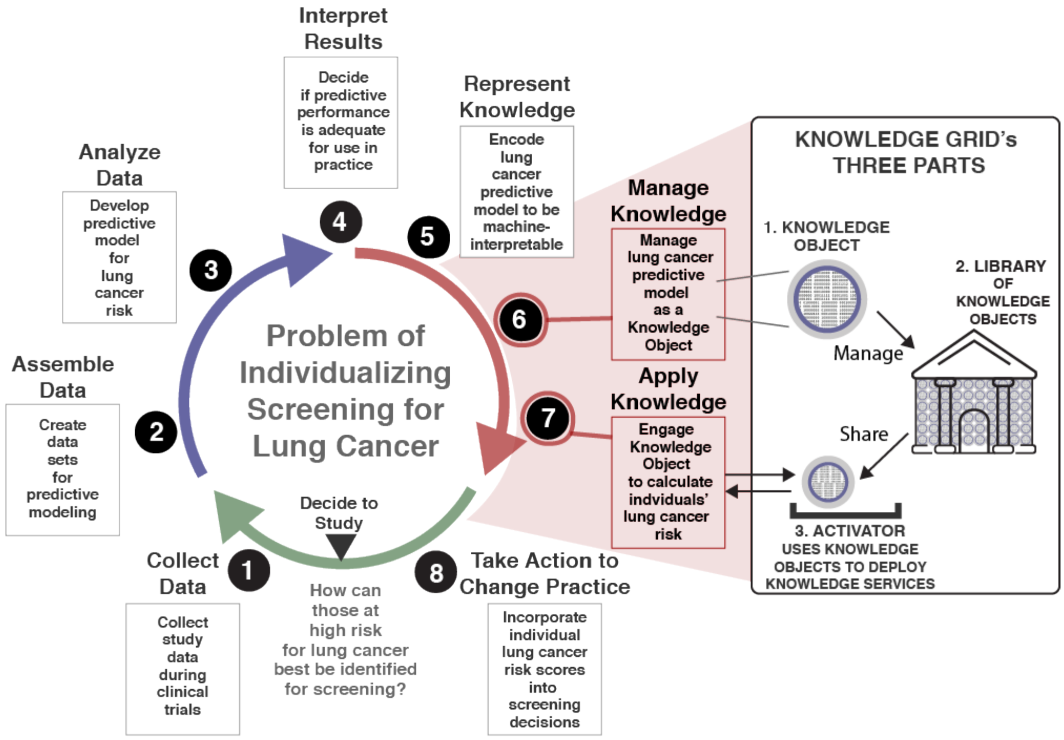
Interpret Results
Decide if predictive performance is adequate for use in practice
Represent Knowledge
Encode lung cancer predictive model to be machine-interpretable
Analyze Data
Develop predictive model for lung cancer risk
Manage Knowledge
Manage lung cancer predictive model as a Knowledge Object
KNOWLEDGE GRID's THREE PARTS
1. KNOWLEDGE OBJECT
2. LIBRARY OF KNOWLEDGE OBJECTS
Manage
Share
4
5
3
6
2
7
1
8
Problem of Individualizing Screening for Lung Cancer
Assemble Data
Create data sets for predictive modeling
Apply Knowledge
Engage Knowledge Object to calculate indviduals' lung cancer risk
Decide to Study
Collect Data
Collect study data during clinical trials
How can those at high risk for lung cancer best be identified for screening?
Take Action to Change Practice
Incorporate individual lung cancer risk scores into screening decisions
3. ACTIVATOR USES KNOWLEDGE OBJECTS TO DEPLOY KNOWLEDGE SERVICES

Figure 2. Learning Cycle Example portraying support from Knowledge Objects and KGrid

KORO is a *general formalism* for a “package”, called a Knowledge Object, that holds functional, computable knowledge from any domain. KORO illuminates the required and optional parts that KOs have and the relationships among those parts.

Unlike biomedical application ontologies, whose purposes are to support specific applications or to annotate existing records[9], KORO is a reference ontology that defines KOs as information artifacts[10]. However, since KORO supports practical reasoning about KOs, it has the character of an application ontology to some degree[10]. KORO’s purpose is to serve as a formal specification for the design of Knowledge Objects, thereby supporting computable knowledge stewardship[11] by helping to make computable knowledge findable, accessible, interoperable, and reusable[12].

Background and Significance

This section begins by describing the LHS context for which KORO was developed. Then it explains how KGrid and KOs support routine learning. Next, perspectives on knowledge and knowledge management systems are highlighted, and KGrid is differentiated from similar platforms. Then, Learning Objects and Research Objects are briefly reviewed to differentiate them from KOs. Finally, the significance of KORO is stated plainly.

Context of Providing Support for the Learning Health System

In LHSs, after a community decides to study a problem, discrete steps can be repeated sequentially to learn (Figures 1 and 2). An example learning cycle is illustrated in Figure 2. The cycle’s goal is to learn, “How can those at higher risk for lung cancer best be identified for screening?” To begin, data are collected (Step 1), assembled (Step 2), and analyzed (Step 3), resulting in a potentially useful predictive model for estimating individuals’ lung cancer risk over the next six years[13]. This predictive model is actionable and can be represented as machine-executable knowledge (Step 5). An implementation of the predictive model can be held in a Knowledge Object, which in turn can be managed in a KGrid Library (Step 6). From there the predictive model can be deployed to support a prediction calculating service using a KGrid Activator. When the lung cancer risk predictive model is remotely invoked by an EHR or other health IT system, it is applied as knowledge (Step 7). This happens by combining the computable predictive model with facts about a person to generate a prediction for that person's lung cancer risk that may be useful in practice.

The KO plays two important roles in the example above. As shown in Figure 2 at Step 6, a KO is used to manage a working implementation of the lung cancer risk predictive model as a persistent resource. At Step 7, the computable predictive model held in a KO is accessed and made serviceable via an Activator. As previously described above, an Activator "activates" computable knowledge in way that brings about and tracks the utilization of a webservice which can be called on by other systems. Early efforts to build KGrid indicate that KOs are workable and potentially useful. To date we have (a) implemented a conceptual model of a KO[7], (b) developed and described a digital Library to manage KOs[5], and (c) built an Activator to apply the knowledge held in KOs via knowledge services. Our work anticipates a future where *computable biomedical knowledge* is created, used in practice, and archived as an integral part of the scientific record[11]. These capabilities provide LHSs with the means to recall and use what they learn.

Perspectives on Knowledge and Knowledge Management Systems

In our work on knowledge infrastructure for LHSs, we adopt two perspectives of the Computer Supported Cooperative Work (CSCW) community. Since the 1980s, CSCW researchers have examined how computer systems can be used to support and coordinate collaborative group activities[14]. An annual ACM conference on CSCW showcases this research (http://cscw.acm.org).

The first CSCW perspective we adopt is that knowledge can be a result of insights gained from empirical analytic or deliberative processes that are situated in a social context occurring at a particular time and place[15]. This epistemological perspective is very relevant to groups of researchers and practitioners collaborating to generate new health knowledge within Learning Health Systems[16]. To support collaborative learning for health, KORO anticipates annotating KOs with a wide variety of descriptive metadata to document KO provenance. This includes metadata about the social context within which KOs are created.

The second CSCW perspective we adopt is that knowledge repositories are necessary but not sufficient to achieve effective knowledge sharing[15]. In published examples[17-19], we find that knowledge repositories are augmented with capabilities for communication, annotation, and search. KGrid's Library is also augmented with capabilities that help users to annotate and deploy computable biomedical knowledge. Furthermore, KGrid may differ some from these previous efforts by assigning to the Knowledge Object the dual roles of **resource** and **service-enabler**. These dual roles are defined in KORO.

Distinguishing Learning Objects from Knowledge Objects

Digital objects for managing knowledge within information systems are not new. In support of e-learning and instructional systems, Learning Objects (LOs) have a significant history. Yet it is hard to understand precisely what a Learning Object is. An overview in 2003 noted numerous "confusing and arbitrary" definitions of LOs[20]. Learning Object content includes text, images, audio, and video files, and software applications. Similar to websites[21], the scope of LOs is so broad that it is difficult to formally define them in a way that can be widely accepted[20]. Even so, for the purpose of achieving interoperability, the technical Shareable Content Object Reference Model (SCORM) standard has been adopted and used in many e-learning systems to date[22]. A "next generation" version of SCORM, based on service-oriented architectures, is planned[23]. Taking note of this history, our approach to formalizing the definition of a Knowledge Object is to develop the KORO ontology and to share it as widely as possible. As indicated in KORO, what distinguishes KOs from Learning Objects is that KOs exist specifically to help manage discrete instances of *computable knowledge* in ways that facilitate their rapid and widespread deployment and use via knowledge services.

Distinguishing Research Objects from Knowledge Objects

We have also explored, in a preliminary way, what makes Knowledge Objects different from Research Objects. There exists a Research Object ontology[24]. (It should not be confused with "RO", the Relation Ontology[25].) Per the Research Object ontology, a Research Object is an *evolving aggregation* of heterogeneous resources for describing and reproducing research workflows, along with annotations about those resources[26]. In contrast, a Knowledge Object packages a *discrete instance of functional computable knowledge* so that the computable knowledge it holds can be shared as a static resource, implemented as a reliable service, and archived as part of the global scientific record. As noted above, KOs play dual roles as resources and service-enablers, and include both resource and service metadata.

Via personal correspondence with the editors of the Research Object ontology, we are aware of the potential to embed KOs inside Research Objects. From the point of view of the Research Object ontology, embedded KOs coexist within Research Objects as 'Resources' that package computable workflow-related 'Scripts' in a formal way. Embedding KOs inside Research Objects could make it easier for researchers to put each other's scripts to use, thereby facilitating evaluation of scripts. Doing this may also facilitate publication of computable analytic results, e.g., predictive models.

Related Ontologies

We reviewed the ontologies from the National Center for Biomedical Ontology online Bioportal (http://bioportal.bioontology.org/) and the OBO Foundry (http://www.obofoundry.org/). We were unable to identify an existing ontology for digital objects intended to help manage and share computable biomedical knowledge. We did find the Software Ontology (SWO), which supports annotation of software tools by type, manufacturer, inputs, outputs, and uses. SWO does not provide enough detail to model a Knowledge Object and its parts[27].

This work raised the question of which, if any, upper level ontology to use for KORO. We first selected the Basic Formal Ontology (BFO) as an upper level ontology for KORO. BFO was chosen at first because it enables ontologies to coordinate and interoperate[28,29]. However, we later discovered the Information Artifact Ontology (IAO). IAO incorporates upper level classes from BFO, and then extends it. This extension of BFO by IAO includes classes for information entities that are useful for KORO, especially the classes *information content entity* and *directive information entity*[8]. For this reason, we ultimately based KORO directly on IAO, which in turn is based on BFO.

We recognize that IAO, which is subject to ongoing refinement, will eventually need to be embedded into a wider framework of domain ontologies, including the Mental Functioning Ontology (MFO)[30]. However, for KORO we attempt only to slightly extend the scope of IAO to cover a new type of information artifact, the Knowledge Object.

Significance of this Work

Learning Health Systems surface a complex set of knowledge management and archiving requirements. These include a requirement to directly connect knowledge to practice within the context of ongoing learning. In LHSs, communities of interest need to continuously learn and then share what they learn easily and widely via existing EHRs and other health IT systems[3,4,16]. In the KGrid approach, KOs play dual roles as archival knowledge resources, for knowledge management, and as service-enablers, for knowledge sharing. By so doing, Knowledge Objects have the potential to help people meet the knowledge management, archiving, and sharing needs of Learning Health Systems[4,5,7].

Research Questions

The following four Research Questions, labeled RQ1 – RQ4, were investigated during the development of KORO.

RQ1. What are required and optional parts of a Knowledge Object Content Package, and how may those parts and their relations be represented and described, in a logically consistent way, as entities in an ontology?

RQ2. What classes, relations and constraints pertain among the whole and the parts of a Knowledge Object, when concretized and materialized, and how may they be represented and made logically consistent in an ontology?

RQ3. What questions specifically about Knowledge Object parthood can KORO answer competently to demonstrate its logic and potential utility?

RQ4. What are necessary implementation decisions that have to be made to successfully implement a Knowledge Object that bears a concretization of computable knowledge within its required instance of a 'Knowledge Object Payload Item' part?

Methods

Ontology Development

A multidisciplinary team at our site, comprised of faculty, graduate students, and developers, collaborated to create KORO. At the outset, we determined that KORO needed to serve as a formal way to describe all of the parts of KOs, and how those parts relate. KORO had to reflect the dual roles of KOs as information resources and knowledge service-enablers. We started with the top-level Basic Formal Ontology (BFO 2.0) and the Information Artifact Ontology (IAO 1.0) merged with BFO[31], each represented in OWL format[29,31]. We followed the steps to develop ontologies using BFO given by Arp, Smith, and Spear[29]. In addition, when incorporating terms from BFO and IAO into KORO, we used the Minimum Information to Reference an External Ontology Term (MIREOT) method[32]. However, we also included some labels, definitions, and examples to help clarify the meanings of BFO and IAO terms used for KORO. When other external sources inspired KORO terms, these sources were explicitly cited in the ontology.

To address Knowledge Object parthood, we focused mostly on BFO continuants. We modeled a Knowledge Object Content Package and its *information content entity* parts. We also modeled material Knowledge Objects and their parts. KORO provides Aristotelian definitions for all KORO classes and properties[29]. After incorporating BFO and IAO classes and their relations (Figure 3), we added KORO-specific content through a process of term identification. We gathered established terms for the parts of KOs from the digital library and information technology worlds. For example, the terms "Fact Sheet", for a document comprised of statements that describe key facts and "Log", for a document resulting from automatic recording of activity by a computer. After assigning existing terms to most parts of KOs, we formalized what is meant by the novel terms in the ontology, e.g., the term 'Knowledge Payload Item.'

Next, we ordered the terms for KORO's entities in a taxonomical hierarchy using the '*is_a*' relation to denote subtypes. Then we added to KORO the minimum number of additional relations needed to represent parthood. We then debated and reordered KORO's entities multiple times through an iterative, collaborative process over ten months. We finally arrived at a logically consistent version 1.3 of KORO, serialized in OWL. After refining KORO using the feedback of experts, we uploaded our original and two improved versions of KORO to the National Center for Biomedical Ontology Bioportal at the following location on the World Wide Web: http://purl.bioontology.org/ontology/KORO.

Evaluation of KORO Using Competency Questions

We developed three Competency Questions (CQs) about knowledge objects and their parts to help illuminate the scope and use of KORO[33]. KORO enables inferences to be made to answer the following three competency questions:

CQ1. According to its parts, is a given instance of an arbitrary entity a Knowledge Object? (Answer: Yes or No)

CQ2. If an arbitrary entity IS an instance of Knowledge Object, what Knowledge Object part items does it have?

CQ3. If an arbitrary entity IS NOT an instance of Knowledge Object, does it partially fulfill the requirements of Knowledge Object, and, if so, what specific Knowledge Object Part Items does it have?

Using Protégé[34] and the Pellet reasoner[35], we created instances of 'Knowledge Object' and 'Knowledge Object Part Item' suitable to demonstrate that KORO properly infers answers to these three competency questions.

Creation of Exemplar Knowledge Objects to Illuminate Implementation Decisions

To create KOs, a variety of implementation-specific decisions have to be made. One goal we have is to future-proof KORO, as much as possible, so that as technologies and capabilities evolve, it will still provide a useful formalism for a package of computable knowledge. For this reason, KORO sets only a general pattern for all KOs. It does not constrain the format(s) of Knowledge Object parts, leaving these to be determined by their creators. Hence, to help identify and illuminate the implementation decisions that have to be made when creating KOs, we have created exemplar KOs of various types. The process of creating actual Knowledge Objects allowed us to list an initial set of implementation questions that need to be answered about each part of a KO when creating new Knowledge Objects.

Results

Table 1 includes counts of KORO's classes and properties. To begin this results section, the definitions for key terms from BFO, IAO, and KORO are provided next as background to assist in reporting the remainder of the results.

Fundamentally, BFO separates all 'Entities' in the world into the following two classes:

> CONTINUANT = def. an *entity* that continues or persists through time
>
> OCCURRANT = def. an *entity* that occurs or happens, such as an event or process with a beginning and an end

Further, BFO separates 'Continuants' into three classes, all three of which are included in KORO:

> GENERICALLY DEPENDENT CONTINUANT = def. a *continuant* that is dependent for its existence on one or other *independent continuants* that can serve as its bearer
>
> INDEPENDENT CONTINUANT = def. a *continuant* that is the bearer of *qualities* such that *qualities* inhere in it
>
> SPECIFICALLY DEPENDENT CONTINUANT = def. a *continuant* that is dependent for its existence on one or more specific *independent continuants* that serve as its bearer

The difference between generic and specific dependence is that generic dependence accounts for exact copies or clones of an 'Entity.' Some 'Entities' can migrate from bearer to bearer, as when copies of the same digital file migrate from one hard drive to another hard drive. A 'Specifically Dependent Continuant' is an 'Entity' that cannot migrate in this manner and instead depends on a specific 'Independent Continuant' bearer for its existence.

IAO extends BFO, starting with the term for this key subclass of 'Generically Dependent Continuant':

> INFORMATION CONTENT ENTITY (ICE) = def. an *entity* which is generically dependent on some *material entity* and which stands in a relation of aboutness to some *entity*

The term 'Material Entity', used in the definition of ICE above, is a subclass of 'Independent Continuant' from BFO that is defined in this way:

> MATERIAL ENTITY = def. an *entity* that has some portion of matter as part

KORO incorporates the term ICE and several terms for ICE subclasses from IAO, with one noted modification:

DATA ITEM = def. an *ICE* that is intended to be a truthful statement about some thing

DIRECTIVE INFORMATION ENTITY = def. an *ICE* whose concretizations indicate to their bearer how to realize some *entity* in a process*

* Note: The underlined portion of this definition differs from the current definition given in IAO, which reads, "how to realize *them* in a process." This change is made in KORO because the referent for "them" in the IAO definition of a 'Directive Information Entity' was not clear to us.

DOCUMENT = def. an *ICE* that is a collection of other *ICEs* intended to be understood together as a whole

KORO then adds a number of new terms. It uses them to help define a key new subclass of 'Document', which has the term 'Knowledge Object Content Package.' That subclass is comprised of instances of five other 'ICE' subclasses. Its definition, and the definition of one of its most important parts, 'Knowledge Object Payload', are given next:

KNOWLEDGE OBJECT CONTENT PACKAGE = def. a *document* including or containing, at a minimum:
some *knowledge object primary identifier*
some *knowledge object resource metadata fact sheet*
some *knowledge object service specification*
some *knowledge object lifecycle log*
some *knowledge object payload*

KNOWLEDGE OBJECT PAYLOAD = def. a *document* comprised of one or more *knowledge content entities*

This brings us to a core term in KORO, which is the term 'Knowledge Content Entity.' It is defined as follows:

KNOWLEDGE CONTENT ENTITY = def. an *ICE* that describes a result, found to be meaningful to one or more person(s) or a community, such that it can be interpreted by them in ways that they value, and which arises from a systematic analytic and/or deliberative process of investigation and study of other *entities*

As noted above, this pragmatic definition of 'Knowledge Content Entity' is informed by prior work of the CSCW community[15]. The definition of a 'Knowledge Content Entity' reflects the contended epistemological notion that knowledge is information that has somehow been upgraded [36,37,38]. Instead of defining precisely how information is upgraded to knowledge, KORO only stipulates the involvement of a systematic process of study. Otherwise, KORO leaves the precise determination of what is knowledge to people and their communities. The following definitions of terms for subclasses of 'Knowledge Content Entity' serve as examples of these types of entities:

CONTENT ANALYSIS RESULT = def., a *knowledge content entity* that is the result of systematically coding and analyzing qualitative (non-numerical) *ICEs*

STATISTICAL ANALYSIS RESULT = def., a *knowledge content entity* that is the result of systematically coding and analyzing quantitative (numerical) *ICEs*

MODEL = def. a *knowledge content entity* comprised of a collection of two or more *data items* and their relationship(s) to each other

EMPIRICAL MODEL = def. a *model* that describes relationships, or correspondences, found to exist among *data items* that describe observations

> THEORETICAL MODEL = def. a *model* that describes, in a hypothetical way based on previous observation, potential relationships, or possible correspondences, among *data items* that have yet to be observed or demonstrated

By definition, then, ‘Models’ and other analytic or deliberative results are ‘Knowledge Content Entities.’ These ‘Knowledge Content Entities’ are parts of ‘Documents’ of the class ‘Knowledge Object Payload.’

In addition to having as one of its parts a ‘Document’ that is a ‘Knowledge Object Payload’, a ‘Knowledge Object Content Package’ also has a ‘Primary Identifier’, a ‘Resource Metadata Fact Sheet’, a ‘Lifecycle Log’, and a ‘Knowledge Object Service Specification’ as its parts.

Next, turning the focus to subclasses of ‘Material Entity’, KORO adopts the definitions of terms for an ‘Artifact’ and an ‘Information Artifact’ given by Smith and Ceusters[30]:

> ARTIFACT = def. a *material entity* created or modified or selected by some *agent* to realize a certain function or *role*
>
> INFORMATION ARTIFACT = def. an *artifact* whose function is to bear an *information carrier*

KORO provides a definition for the above term ‘Agent’ (not shown) and IAO provides the following definition for the above term ‘Information Carrier’:

> INFORMATION CARRIER = def. a *specifically dependent continuant* that is a *quality* of an information bearer that imparts the information content

By defining and relating entities, KORO finally arrives at these formal definitions of a ‘Knowledge Object’ and ‘Knowledge Object Part Item’:

> KNOWLEDGE OBJECT = def. an *information artifact* with *knowledge object part items* that bears a concretization of a *knowledge object content package*
>
> KNOWLEDGE OBJECT PART ITEM = def. an *information artifact* that bears a concretization of an *ICE* that exists within a *knowledge object content package*

Although the definitions of the classes ‘Knowledge Object’ and ‘Knowledge Object Part Item’ above do not require it, we anticipate that most actual Knowledge Objects and Knowledge Object Part Items in the world will take the form of digital files. When they take a digital form, instances of these two subclasses of ‘Information Artifact’ better enable the rapid, widespread sharing, deployment, and use of computable knowledge that we intend to achieve.

Having reviewed this background material, we can now proceed to report the remainder of the results. What follows are results organized as responses to the four research questions, RQ1 – RQ4, given above.

RQ1. ***What are required and optional parts of a Knowledge Object Content Package, and how may those parts and their relations be represented and described in a logically consistent way as entities in an ontology?***

According to KORO, a ‘Knowledge Object Content Package’ is a ‘Document’, which is an ‘Information Content Entity’, which is a ‘Generically Dependent Continuant.’ Figure 3 below depicts the portion of KORO’s ‘*is_a*’ relation hierarchy that shows this. It illustrates the parts of a ‘Knowledge Object Content Package’, thereby answering RQ1.

In Figure 3, the key class, ‘Knowledge Object Content Package’, appears toward the center. There are five “1st Degree” parts that have to be included for a ‘Document’ to be an instance of the class ‘Knowledge Object Content Package.’ Subclasses for these five “1st degree” parts are marked with a red oval and connected to the ‘Knowledge Object Content Package’ class in Figure 3 by a thick black line portraying the ‘*has_part*’ relation. (Note: Other, optional subclasses for parts of an instance of ‘Knowledge Object Content Package’ are marked by purple-pink ovals. To avoid clutter, the ‘*has_part*’ relation to these optional part subclasses is not portrayed in Figure 3.)

In addition, because some ‘Knowledge Object Content Package’ parts have their own required parts, four more “2nd Degree” required parts of a ‘Knowledge Object Content Package’ are indicated with pink ovals. These are connected to the classes of the parts that require them by thin, red, curved lines. Optional parts of a ‘Knowledge Object Content Package’ include those parts indicated by the purple-pink ovals. Finally, in KORO inverse relationships exist between the classes ‘Knowledge Content Entity’ and ‘Information Content Entity.’ These are the ‘*accounts_for’* and ‘*is_accounted_for_by’* inverse relationships. They are defined in the following way:

> ACCOUNTS FOR = def. a question-answer relationship between an *information content entity* and a *knowledge content entity* whereby some *knowledge content entity* accounts for, meaning explains, the answer to a question about the *information content entity*
>
> IS ACCOUNTED FOR BY = def. an answer-question relationship between a *knowledge content entity* and an *information content entity* whereby an answer to a question about some *information content entity* is accounted for by, meaning is explained by, some *knowledge content entity*

These relationships address a known issue with IAO, which is the need to refine or extend IAO in some manner to include the concepts of truth and/or knowledge[30]. In this regard, we assert that it is not the case that all Information Content Entities are truth-bearing. Indeed, the ‘Directive Information Entity’ class is defined in a way that instances of procedures may be members of this class. Yet, according to Floridi[36], procedures, e.g., “lock the door at 4pm”, do not “qualify alethically”, meaning they cannot be correctly qualified as true or false. We have been able to create early versions of KORO while recognizing particular issues like this with IAO that remain unsettled.

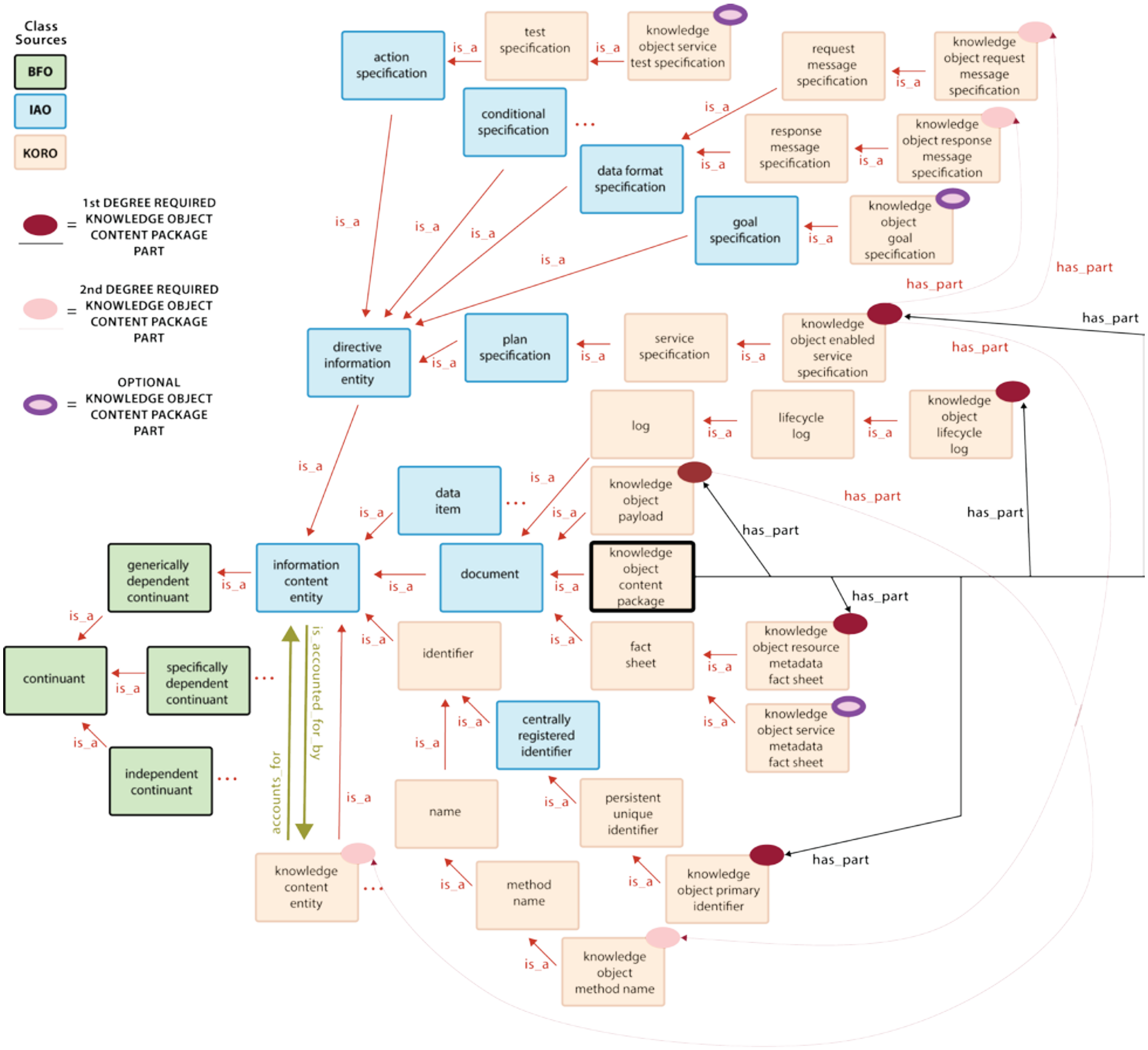
Class Sources
BFO
IAO
KORO
= 1st DEGREE REQUIRED KNOWLEDGE OBJECT CONTENT PACKAGE PART
= 2nd DEGREE REQUIRED KNOWLEDGE OBJECT CONTENT PACKAGE PART
= OPTIONAL KNOWLEDGE OBJECT CONTENT PACKAGE PART
action specification
test specification
knowledge object service test specification
request message specification
knowledge object request message specification
conditional specification
data format specification
response message specification
knowledge object response message specification
goal specification
knowledge object goal specification
directive information entity
plan specification
service specification
knowledge object enabled service specification
log
lifecycle log
knowledge object lifecycle log
data item
knowledge object payload
generically dependent continuant
information content entity
document
knowledge object content package
continuant
specifically dependent continuant
independent continuant
identifier
fact sheet
knowledge object resource metadata fact sheet
centrally registered identifier
knowledge object service metadata fact sheet
name
persistent unique identifier
knowledge content entity
method name
knowledge object primary identifier
knowledge object method name
is_a
has_part
is_accounted_for_by
accounts_for
...

Figure 3. Portion of KORO indicating the required and optional parts of a Knowledge Object Content Package

RQ2. ***What classes, relations and constraints pertain among the whole and the parts of a Knowledge Object, when concretized and materialized, and how may they be represented and made logically consistent in an ontology?***

By definition, a Knowledge Object is a 'Material Entity.' This means that all Knowledge Objects have some "portion of matter."[29] A potential ontological confusion arises here. In BFO and IAO, there are subclasses of 'Material Entity' termed 'Object' and 'Object Aggregate.' Yet in a more recent work, two of the creators of IAO have defined 'Artifact' and 'Information Artifact' as subclasses of 'Material Entity', without making it entirely clear how these subclasses relate to the 'Object' and 'Object Aggregate' subclasses[30]. For KORO, we opted to include all four of these subclasses of 'Material Entity.' We did this because the definitions of 'Artifact' and 'Information

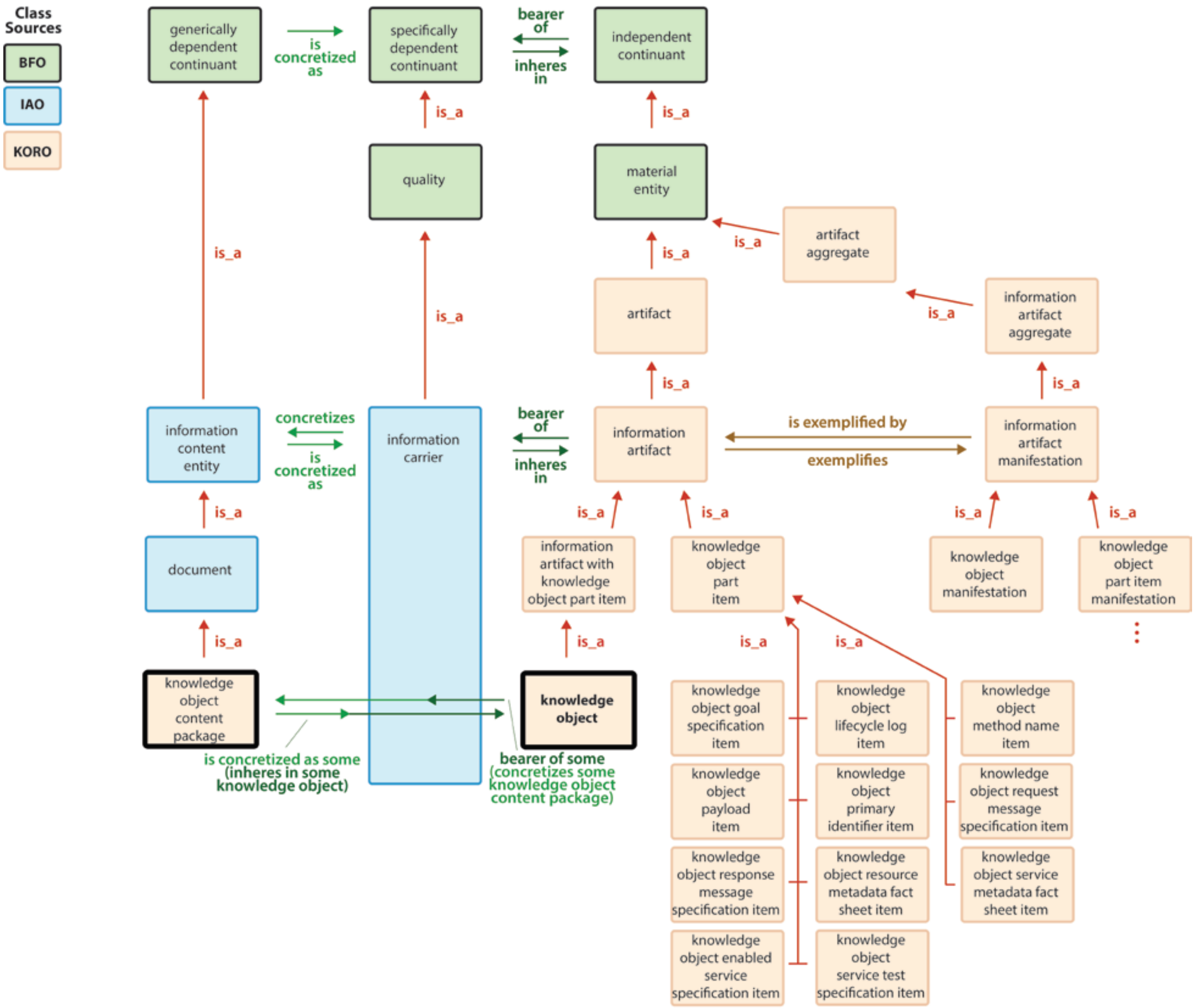
Class Sources
BFO
IAO
KORO
generically dependent continuant
is concretized as
specifically dependent continuant
bearer of
inheres in
independent continuant
is_a
quality
material entity
artifact aggregate
artifact
information artifact aggregate
information content entity
concretizes
is concretized as
information carrier
information artifact
is exemplified by
exemplifies
information artifact manifestation
document
information artifact with knowledge object part item
knowledge object part item
knowledge object manifestation
knowledge object part item manifestation
knowledge object content package
knowledge object
is concretized as some (inheres in some knowledge object)
bearer of some (concretizes some knowledge object content package)
knowledge object goal specification item
knowledge object lifecycle log item
knowledge object method name item
knowledge object payload item
knowledge object primary identifier item
knowledge object request message specification item
knowledge object response message specification item
knowledge object resource metadata fact sheet item
knowledge object service metadata fact sheet item
knowledge object enabled service specification item
knowledge object service test specification item

Figure 4. Portion of KORO indicating the whole and parts of a Knowledge Object with key relationships

Artifact' are more recent and more specific to our purposes[30]. As indicated above, this means that a 'Knowledge Object' is an 'Information Artifact.' To limit confusion about the two similar terms, 'Object' and 'Artifact', we reserve the term 'Object' in KORO for entities that do not bear concretizations of 'Information Content Entities.'

To answer RQ2 it is first necessary to recount how, in BFO, 'Generically Dependent Continuants' relate to 'Specifically Dependent Continuants.' Further, it is also necessary to recount how 'Specifically Dependent Continuants' in turn relate to 'Independent Continuants.' Here are definitions from BFO for these relationships:

> IS CONCRETIZED AS = def. a relationship between a *generically dependent continuant* and a *specifically dependent continuant*, in which the *generically dependent continuant* depends on some *independent continuant* in virtue of the fact that the *specifically dependent continuant* also depends on that same *independent continuant*

INHERES IN = def. a relation between a *specifically dependent continuant* (the dependent) and an *independent continuant* (the bearer), in which the dependent specifically depends on the bearer for its existence

The two relationships defined immediately above have the following two corresponding inverse relationships that are also defined in BFO:

CONCRETIZES = def. a relationship between a *specifically dependent continuant* and a *generically dependent continuant*, in which the *generically dependent continuant* depends on some *independent continuant* in virtue of the fact that the *specifically dependent continuant* also depends on that same *independent continuant*

BEARER OF = def. a relation between an *independent continuant* (the bearer) and a *specifically dependent continuant* (the dependent), in which the dependent specifically depends on the bearer for its existence

At the top of Figure 4, the inverse relationships between the three subclasses of 'Continuant' are illustrated. Next, toward the middle of Figure 4, the 'Specifically Dependent Continuant' subclass of 'Quality' is shown. It has a key subclass from IAO, 'Information Carrier', which is also defined above. Instances of 'Information Carrier' are essentially intermediating entities between ICEs and material 'Information Artifacts.' As shown in Figure 4, through instances of 'Information Carrier', instances of 'Knowledge Object Content Package' are concretized and then made to inhere in instances of material 'Knowledge Objects.' Hence, all Knowledge Objects are instances of 'Material Entity', which in turn are instances of 'Independent Continuant.'

To give further detail, the following key terms for six of KORO's classes of 'Independent Continuant' are defined below and portrayed on the right in Figure 4:

ARTIFACT AGGREGATE = def. a *material entity* consisting exactly of a collection of *artifacts* as its member parts at all times at which it exists

INFORMATION ARTIFACT AGGREGATE = def. an *artifact aggregate* that is made up of a collection of *information artifacts* bearing concretizations of *information content entities*

KNOWLEDGE OBJECT PART ITEM = def. an *information artifact* that is a bearer of a concretization of a required or optional part of a *knowledge object*

INFORMATION ARTIFACT MANIFESTATION = def. an *information artifact aggregate* comprised of all instances in reality of some particular *information artifact*

KNOWLEDGE OBJECT MANIFESTATION = def. an *information artifact aggregate* comprised of all instances in reality of a particular and uniquely identifiable *knowledge object*

KNOWLEDGE OBJECT PART ITEM MANIFESTATION = def. an *information artifact aggregate* comprised of all instances in reality of a particular part item of a *knowledge object*

In addition, the following key terms for inverse relationships between 'Information Artifact' and 'Information Artifact Aggregate' are also portrayed in Figure 4:

IS EXEMPLIFIED BY = def. a particular *information artifact manifestation* is exemplified by a particular *information artifact* when the particular *information artifact* is one of potentially many material bearers of the same *information content entity*, and the particular *information artifact manifestation* is the collection of all material bearers of that particular *information content entity*

EXEMPLIFIES = def. a particular *information artifact* exemplifies a particular *information artifact manifestation* when the particular *information artifact* is one of potentially many material

> bearers of the same *information content entity*, and the particular *information artifact manifestation* is the collection of all material bearers of that particular *information content entity*

These inverse '*Is exemplified by*' and '*Exemplified*' relations are directly inspired by the Functional Requirements for Bibliographic Records (FRBR) conceptual model of the International Federation of Library Associations[39]. We find that the creators of IAO have also been influenced by FRBR[31]. The IAO definition of a 'Textual entity' indicates this:

> TEXTUAL ENTITY = def. a part of a *manifestation* (FRBR sense) that is a *generically dependent continuant* whose concretizations are patterns of glyphs intended to be interpreted as words, formulas, etc.

However, in contrast to the definition of 'Textual Entity' above from IAO, because in FRBR manifestations and items embody 'Material Entities', we interpret a manifestation otherwise to be an 'Independent Continuant' and not a 'Generically Dependent Continuant.' Our interpretation is reinforced by the FRBR-aligned Bibliographic Ontology (FaBIO), wherein the relation 'has manifestation' has as a sub-property 'has embodiment' to indicate that a manifestation embodies (i.e., makes tangible) some expression[40].

The answer to RQ2 can now be summarized. By definition, every Knowledge Object is a 'Material Entity' that bears a concretization of an instance of 'Knowledge Object Content Package', which in turn is an instance of 'Document.' Thus, the content of every Knowledge Object is constrained by the definition of 'Knowledge Object Content Package' above, which stipulates that every instance of 'Knowledge Object Content Package' has five required parts. When the defined parts of a 'Knowledge Object Content Package' are concretized as instances of 'Information Carrier', then they must inhere in a material instance of 'Knowledge Object Part Item.' As shown in Figure 4, the class 'Knowledge Object Part Item' has 11 subclasses to account for this, one for each type of 'Knowledge Object Part Item' in KORO.

Finally, as Figure 4 also indicates, because instances of 'Knowledge Object' and 'Knowledge Object Part Item' are also instances of 'Information Artifact', they exemplify some 'Information Artifact Manifestation.' Hence there are subclasses of 'Information Artifact Manifestation' corresponding to the class 'Knowledge Object', and also to the class 'Knowledge Object Part Item' and its subclasses (not shown).

RQ3. ***What questions specifically about Knowledge Object parthood can KORO answer competently to demonstrate its logic and potential utility?***

When given an instance of an arbitrary entity, described in the form of OWL axioms, KORO supports automated reasoning to infer things about the makeup and parts of the arbitrary entity. By so doing, it enables effective answering of three competency questions, CQ1, CQ2, and CQ3. The results of KORO's performance in this regard are organized by competency question and given in the following text:

CQ1. *According to its parts, is a given instance of an arbitrary entity a Knowledge Object? (Answer: Yes or No)*

To answer CQ1, KORO provides sufficient logic to the Pellet reasoner to infer from axioms specifying the parts that an arbitrary entity has, whether or not the arbitrary entity is a member of the class 'Knowledge Object.' KORO actually enables a machine to answer this question in two different ways. In one way, KORO supports reasoning over axioms that indicate an entity has parts that are instances of 'Knowledge Object Part Item' to determine whether or not those instances include all five required parts comprising a 'Knowledge Object.' In another way, KORO supports reasoning over axioms that indicate an entity has parts that are instances of a 'Knowledge Object Content Package' to determine whether or not those instances include all five required parts comprising a 'Knowledge Object Content Package.' In the latter case, KORO supports further reasoning to correctly infer that a 'Knowledge Object' entity exists when it is asserted that an arbitrary entity is a bearer of a concretization of an instance of 'Knowledge Object Content Package.'

CQ2. *If an arbitrary entity IS an instance of Knowledge Object, what Knowledge Object part items does it have?*

To answer CQ2, KORO provides sufficient logic to infer an arbitrary entity has parts that are instances of ‘Knowledge Object Part Item’ and to identify precisely what particular types of ‘Knowledge Object Part Item’ the arbitrary entity has. To do this, KORO takes advantage of the logic of the inverse relations ‘has part’ and ‘is part of.’

CQ3. *If an arbitrary object instance IS NOT an instance of Knowledge Object, does it partially fulfill the requirements of Knowledge Object, and, if so, what specific Knowledge Object Part Items does it have?*

Building on the logic used to answer CQ1 and CQ2, KORO includes a class ‘Information Artifact with Knowledge Object Part Item’ to enable reasoning that results in answers to CQ3. Using this class, KORO provides sufficient logic to infer from axioms that assert an arbitrary entity has parts that are instances of ‘Knowledge Object Part Item’ whether or not that entity is an instance of ‘Knowledge Object.’ When, an entity having instances of ‘Knowledge Object Part Item’ is determined NOT TO BE an instance of ‘Knowledge Object’, then KORO enables inferences about the part items it does have. This reasoning supports automated determination of the parts of a Knowledge Object the entity is missing by comparison to a predefined list of all required ‘Knowledge Object Part Items.’

RQ4. ***What are necessary implementation decisions that have to be made to successfully implement a Knowledge Object that bears a concretization of computable knowledge within its required instance of a ‘Knowledge Object Payload Item’ part?***

To answer *RQ4*, we have built a simple working example of an instance of ‘Knowledge Object.’ It is referred to from hereon as KO^1. KO^1 has, as one of its necessary parts, an instance of ‘Knowledge Object Payload Part Item.’ It is referred to as payload-of-KO^1. Payload-of-KO^1 bears a concretization of a very simple mathematical model, which is an example based on a previously published lung cancer risk predictive model[13].

For this simple example, required Knowledge Object parts for KO^1 are listed in Table 2, column 1. Corresponding examples of the information content constituting the parts of KO^1 are listed in Table 2, column 2.

Creators of Knowledge Objects have innumerable options for formatting the information content concretized and born by the material parts of their KOs. For each of the five 1st degree required ‘Information Content Entity’ parts making up a KO (and for all other parts besides), three common implementation questions pertain:

1. What will be the information model or ontology used to represent the content contained in each KO part?
2. How will the information in each KO part be serialized so that it can be managed as a resource in an instance of the Knowledge Grid’s digital Library component[5]?
3. How will the information in each KO part be serialized for use as a service-enabler by the Knowledge Grid’s Activator component, which allows its users to quickly stand up KO-based webservices?

Since three common questions pertain to each KO part, to create the five 1st degree required KO parts for any given KO involves answering 15 different implementation questions. For our simplified example of KO^1, answers to these 15 implementation questions are provided in the 3rd, 4th, and 5th columns of Table 2.

We are currently testing the performance of the actual lung cancer risk prediction model packaged as a resource and made available to support a webservice by using a KO. Using an instance of the KGrid’s Activator, we load our Knowledge Object into the Activator and then the Activator uses it to engender a webservice. The Activator exposes the computable payload in a way that allows external systems to call on it. Eventually, we plan to deploy Knowledge Objects to provide knowledge-as-a-service. We hope this method will provide a highly scalable way to generate consistent, actionable health information for decision-makers.

Discussion

We developed KORO to overcome issues that arose during early development of the Knowledge Grid infrastructure-platform. Before KORO, we were unable to consistently answer simple questions about Knowledge Objects, such as, "What are the mandatory and optional parts of a Knowledge Object?" Now, with KORO, these and other questions about Knowledge Objects and their parts can be answered in a consistent way.

In addition, Knowledge Objects, as we conceive of them, are capable of bearing any form of concretized Knowledge Content Entity. There is no restriction as to the computer language used to encode the information in a Knowledge Object Payload Item. The Knowledge Grid infrastructure-platform is being built in ways that make it computer language agnostic where Knowledge Object Payload Items are concerned. As much as possible, KGrid Activator technology is being made extensible so that webservices arise from computable biomedical knowledge encoded in a wide array of languages including Python, R, Java, and Javascript.

We envision future work that will leverage KORO to help us build more capable Knowledge Grid components. Soon we plan to further upgrade the Knowledge Grid's digital Library in accordance with KORO. Among other things, this digital Library upgrade will focus on better support for Knowledge Object versioning and archiving. We also look forward to further investigating how the W3C Open Annotation Model may inform our work to add and manage resource and service related metadata to Knowledge Objects.

Because the scope of the present paper is limited to defining and describing Knowledge Objects and their parts, here we have not covered the definitions of the dual roles Knowledge Objects play as resource and service-enablers. We have also ignored for now any defined processes that involve Knowledge Objects, especially instances of the class 'Service Interaction' in KORO. We look forward to further work on KORO that will focus more attention on formalizing roles and processes involving Knowledge Objects.

Finally, we believe that the development of KORO is one of many necessary steps towards computable knowledge interoperability. By formally defining what Knowledge Objects are in an ontology, future knowledge management systems can gain the capability to determine automatically whether or not arbitrary entities are Knowledge Objects, and to delineate whether the parts those entities have are formally defined Knowledge Object Part Items.

Conclusion

KORO is a realist ontology built using the Basic Formal Ontology, BFO, and the Information Artifact Ontology, IAO. It supports automated reasoning to determine whether or not an entity is a Knowledge Object. A Knowledge Object is a material information artifact that bears a concretized instance of 'Knowledge Object Content Package.' Knowledge Objects have a number of required parts, including an instance of 'Knowledge Object Payload Item.' In turn, an instance of 'Knowledge Object Payload Item' is a material information artifact that bears a concretized instance of 'Knowledge Content Entity.' Thus, KORO specifies a workable package for managing and sharing any instance of computable knowledge, including instances of computable biomedical knowledge that are needed to sustain and support Learning Health Systems.

Acknowledgments

The following people at the University of Michigan helped with the development of the Knowledge Object Reference Ontology: Rocky Fischer, Frank Manion, and Wei Shi, from Cancer Center Informatics; Andrew Cullen, Lisa Ferguson, Oliver Gadabu, Kristen McGarry, George Meng, and Maria G. Trinidad from the Department of Learning Health Sciences; and Jeff Cowall, Bob Riddle, and John Walsh of Health Information Technology Services.

References

1. Olsen L, Aisner D, McGinnis JM. Institute of Medicine (US). Roundtable on Evidence-Based Medicine. The Learning Healthcare System: workshop summary. 2007; doi:10.17226/11903.

2. Friedman CP, Allee NJ, Delaney BC, Flynn AJ, Silverstein JC, Sullivan K, Young KA. The science of learning health systems: foundations for a new journal. Learning Health Systems. 2017 Jan 1.
3. Friedman CP, Rubin J, Brown J, Buntin M, Corn M, Etheredge L, Gunter C, Musen M, Platt R, Stead W, Sullivan K. Toward a science of learning systems: a research agenda for the high-functioning learning health system. Journal of the American Medical Informatics Association. 2014 Oct 23.
4. Friedman CP, Rubin JC, Sullivan KJ. Toward an Information Infrastructure for Global Health Improvement. IMIA Yearbook. 2017.
5. Flynn AJ, Bahulekar N, Boisvert P, Lagoze C, Meng G, Rampton J, Friedman CP. Architecture and Initial Development of a Digital Library Platform for Computable Knowledge Objects for Health. Studies in health technology and informatics. 2017.
6. Lagoze C, Payette S, Shin E, Wilper C. Fedora: an architecture for complex objects and their relationships. International Journal on Digital Libraries. 2006 Apr 1;6(2):124-38.
7. Flynn AJ, Shi W, Fischer R, Friedman CP. Digital knowledge objects and digital knowledge object clusters: unit holdings in a learning health system knowledge repository. In System Sciences (HICSS), 2016 49th Hawaii International Conference on 2016 Jan 5 (pp. 3308-3317). IEEE.
8. Ruttenberg A, Courtot M, Peters B, Rees J, Malone J. The information artifact ontology. 2008. Accessed 5 Jan 2017 at: http://www.obofoundry.org/ontology/iao.html
9. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA. Gene Ontology: tool for the unification of biology. Nature genetics. 2000 May 1;25(1):25-9.
10. Menzel C. Reference Ontologies-Application Ontologies: Either/Or or Both/And?. In KI Workshop on Reference Ontologies and Application Ontologies 2003 Sep 16 (Vol. 16).
11. Van de Sompel H, Treloar A. A perspective on archiving the scholarly web. IPRES 2014 proceedings. 2014:194.
12. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J. The FAIR guiding principles for scientific data management and stewardship. Scientific data. 2016;3.
13. Tammemägi MC, Katki HA, Hocking WG, Church TR, Caporaso N, Kvale PA, Chaturvedi AK, Silvestri GA, Riley TL, Commins J, Berg CD. Selection criteria for lung-cancer screening. New England Journal of Medicine. 2013 Feb 21;368(8):728-36.
14. Carstensen PH, Schmidt K. Computer supported cooperative work: New challenges to systems design. InIn K. Itoh (Ed.), Handbook of Human Factors 1999.
15. Ackerman MS, Dachtera J, Pipek V, Wulf V. Sharing knowledge and expertise: the CSCW view of knowledge management. Computer Supported Cooperative Work (CSCW). 2013 Aug 1;22(4-6):531-73.
16. Faden RR, Kass NE, Goodman SN, Pronovost P, Tunis S, Beauchamp TL. An ethics framework for a learning health care system: a departure from traditional research ethics and clinical ethics. Hastings Center Report. 2013 Jan 1;43(s1).
17. Ackerman MS, McDonald DW. Answer Garden 2: merging organizational memory with collaborative help. In Proceedings of the 1996 ACM conference on Computer supported cooperative work 1996 Nov 16. ACM.
18. Dourish P, Edwards WK, LaMarca A, Lamping J, Petersen K, Salisbury M, Terry DB, Thornton J. Extending document management systems with user-specific active properties. ACM Transactions on Information Systems (TOIS). 2000 Apr 1;18(2):140-70.
19. Laqua S, Sasse MA, Greenspan S, Gates C. Do you know dis?: a user study of a knowledge discovery tool for organizations. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2011 May 7 (pp. 2887-2896). ACM.
20. Polsani PR. Use and abuse of reusable learning objects. Journal of Digital information. 2006 Feb 27;3(4).
21. Brügger N. Website history and the website as an object of study. New Media & Society. 2009 Feb;11(1-2):115-32.
22. Sinclair J, Joy M, Yau JY, Hagan S. A practice-oriented review of learning objects. IEEE Transactions on learning technologies. 2013 Apr;6(2):177-92.

23. Poltrack J, Hruska N, Johnson A, Haag J. The next generation of SCORM: innovation for the global force. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) 2012 (Vol. 2012, No. 1). Orlando: National Training System Association.
24. Bechhofer S, De Roure D, Gamble M, Goble C, Buchan I. Research objects: Towards exchange and reuse of digital knowledge, 2010.
25. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mungall CJ, Leontis N. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature biotechnology. 2007 Nov 1;25(11):1251-5.
26. Soiland-Reyes, S., Bechhofer S., Eds., Research Object ontology, Release 2016-01-28 Available at: http://wf4ever.github.io/ro/2016-01-28/ Accessed on: December 10, 2017.
27. Malone J, Brown A, Lister AL, Ison J, Hull D, Parkinson H, Stevens R. The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. Journal of biomedical semantics. 2014 Jun 2;5(1):25.
28. Smith B, Kumar A, Bittner T. Basic formal ontology (BFO) for bioinformatics. Journal of Information Systems. 2005:1-6.
29. Arp R, Smith B, Spear AD. Building ontologies with basic formal ontology. MIT Press; 2015 Jul 31.
30. Smith B, Ceusters W. Aboutness: Towards foundations for the information artifact ontology. In International Conference on Biomedical Ontology, Lisbon, Portugal 2015 Jul (pp. 47-51).
31. Information Artifact Ontology. Available at: https://raw.githubusercontent.com/information-artifact-ontology/IAO/master/releases/2017-03-24/iao-merged.owl Accessed on: December 10, 2017.
32. Courtot M, Gibson F, Lister AL, Malone J, Schober D, Brinkman RR, Ruttenberg A. MIREOT: The minimum information to reference an external ontology term. Applied Ontology. 2011 Jan 1;6(1):23-33.
33. Ren Y, Parvizi A, Mellish C, Pan JZ, Van Deemter K, Stevens R. Towards competency question-driven ontology authoring. In European Semantic Web Conference 2014 May 25 (pp. 752-767). Springer.
34. Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Tu SW. The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human-computer studies. 2003 Jan 31;58(1):89-123.
35. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web. 2007 Jun 30;5(2):51-3.
36. Floridi L. Semantic information and the correctness theory of truth. Erkenntnis. 2011 Mar 1;74(2):147-75.
37. Floridi L. Semantic information and the network theory of account. Synthese. 2012 Feb 1;184(3):431-54.
38. Weinberger D. Too Big to Know: Rethinking Knowledge Now That the Facts Aren't the Facts, Experts Are Everywhere, and the Smartest Person in the Room Is the Room. Basic Books; 2011.
39. Tillett B. Functional requirements of bibliographic records (FRBR): final report. International Federation of Library Associations and Institutions (IFLA) 1998.
40. Shotton D, Peroni S. FaBiO, the FRBR-aligned bibliographic ontology. Available at: http://www.sparontologies.net/ontologies/fabio/source.html Accessed on: December 12, 2017.
41. Kunze J. Towards electronic persistence using ARK identifiers. In Proceedings of the 3rd ECDL Workshop on Web Archives 2003 Aug 21.
42. Weibel S. The Dublin Core: a simple content description model for electronic resources. Bulletin of the American Society for Information Science and Technology. 1997 Oct 1;24(1):9-11.
43. Missier P, Belhajjame K, Cheney J. The W3C PROV family of specifications for modelling provenance metadata. In Proceedings of the 16th International Conference on Extending Database Technology 2013 Mar 18. ACM.

Table 1. KORO Summary Statistics

	NUMBER OF CLASSES	NUMBER OF PROPERTIES
Total	110	19
From BFO	16	12
From IAO	16	3
Unique to KORO	78	4

Table 2. Implementation decisions made to create an instance of 'Knowledge Object' called KO[1]

REQUIRED PART	EXAMPLE CONTENTS	INFORMATION MODEL	LIBRARY SERIALIZATION for use as a RESOURCE	ACTIVATOR SERIALIZATION to enable a SERVICE
Knowledge Object Primary Identifier Item	ark:/99999/fk4jh3tk9s	Archival Resource Key[41]	As RDF† triple	As JSON◆
Knowledge Object Payload Item	Risk score (%) $=V1^{0.4}+V2^{1.4}-V3^{2}$	Python 2.7	As binary file	As JSON
Knowledge Object Resource Metadata Fact Sheet	Title: Risk Model Description: Predictive Model Creator: A. Person	Dublin Core Metadata Set[42]	As RDF triples	As JSON
Knowledge Object Enabled Service Specification	Accepts: V1:integer; V2:integer; V3:integer Returns Y:float	Custom	As RDF/XML★ file	As key-value pairs in JSON
Knowledge Object Lifecycle Log	Created on: 1-1-2017	PROV-O Ontology[43]	As RDF triples	~ N/A ~

†RDF stands for Resource Description Framework. ◆JSON stands for Javascript Object Notation. ★XML stands for eXtensible Markup Language.

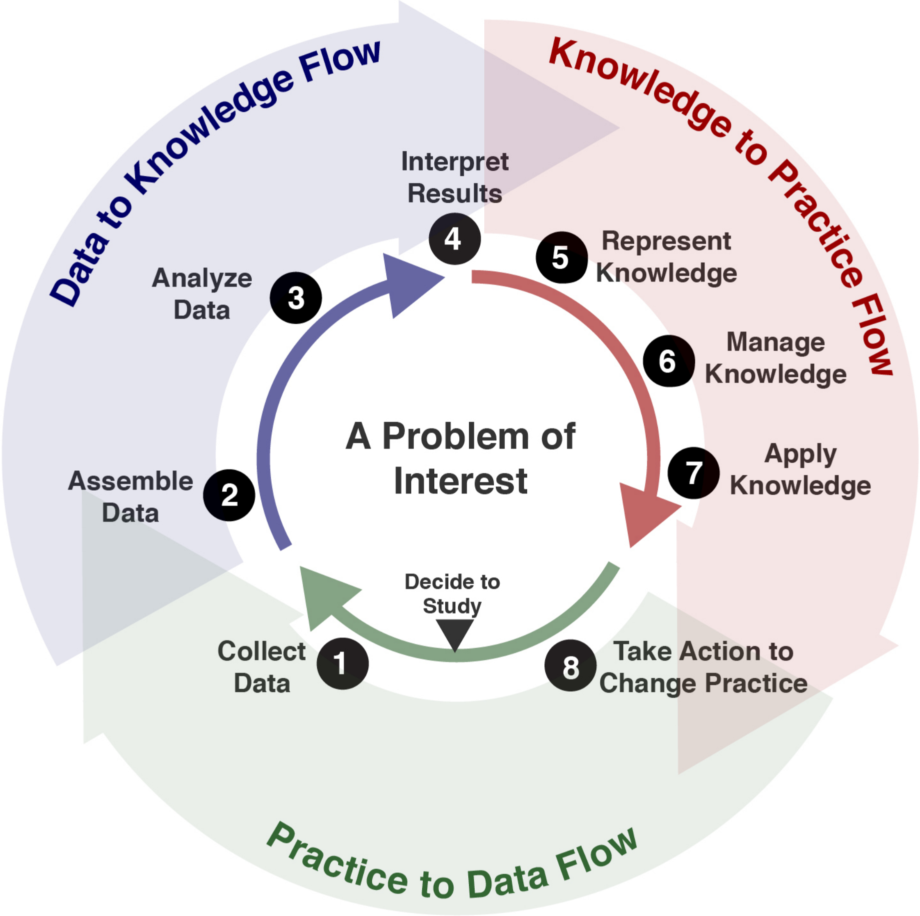

Figure 1. The Learning Health Cycle of the LHS with Three Information Flows and Eight Steps

LRH2_10054_F1.jpg

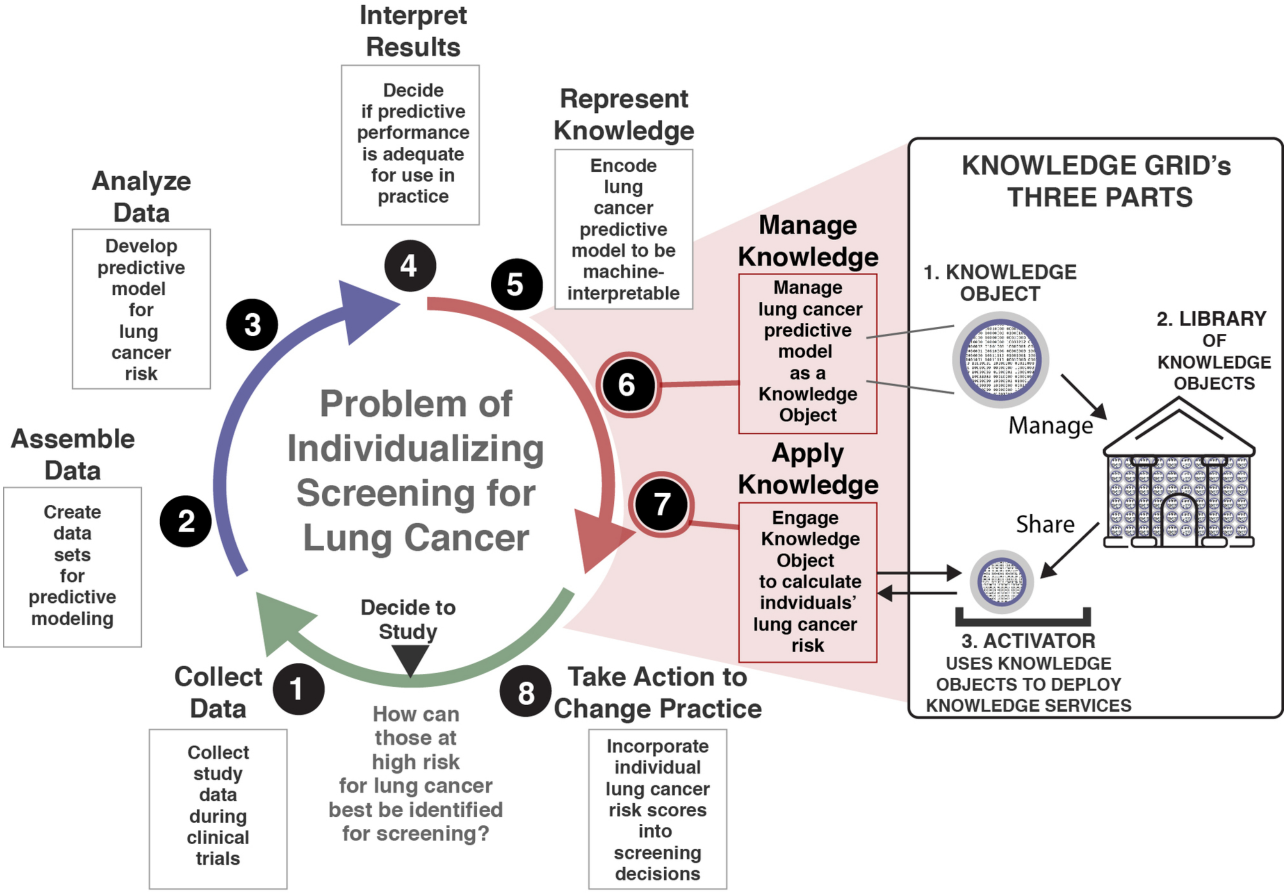
Interpret Results
Decide if predictive performance is adequate for use in practice
Represent Knowledge
Encode lung cancer predictive model to be machine-interpretable
Analyze Data
Develop predictive model for lung cancer risk
Assemble Data
Create data sets for predictive modeling
Collect Data
Collect study data during clinical trials
Manage Knowledge
Manage lung cancer predictive model as a Knowledge Object
Apply Knowledge
Engage Knowledge Object to calculate indviduals' lung cancer risk
Take Action to Change Practice
Incorporate individual lung cancer risk scores into screening decisions
1
2
3
4
5
6
7
8
Problem of Individualizing Screening for Lung Cancer
Decide to Study
How can those at high risk for lung cancer best be identified for screening?
KNOWLEDGE GRID's THREE PARTS
1. KNOWLEDGE OBJECT
2. LIBRARY OF KNOWLEDGE OBJECTS
Manage
Share
3. ACTIVATOR USES KNOWLEDGE OBJECTS TO DEPLOY KNOWLEDGE SERVICES

Figure 2. Learning Cycle Example portraying support from Knowledge Objects and KGrid

LRH2_10054_F2.jpg

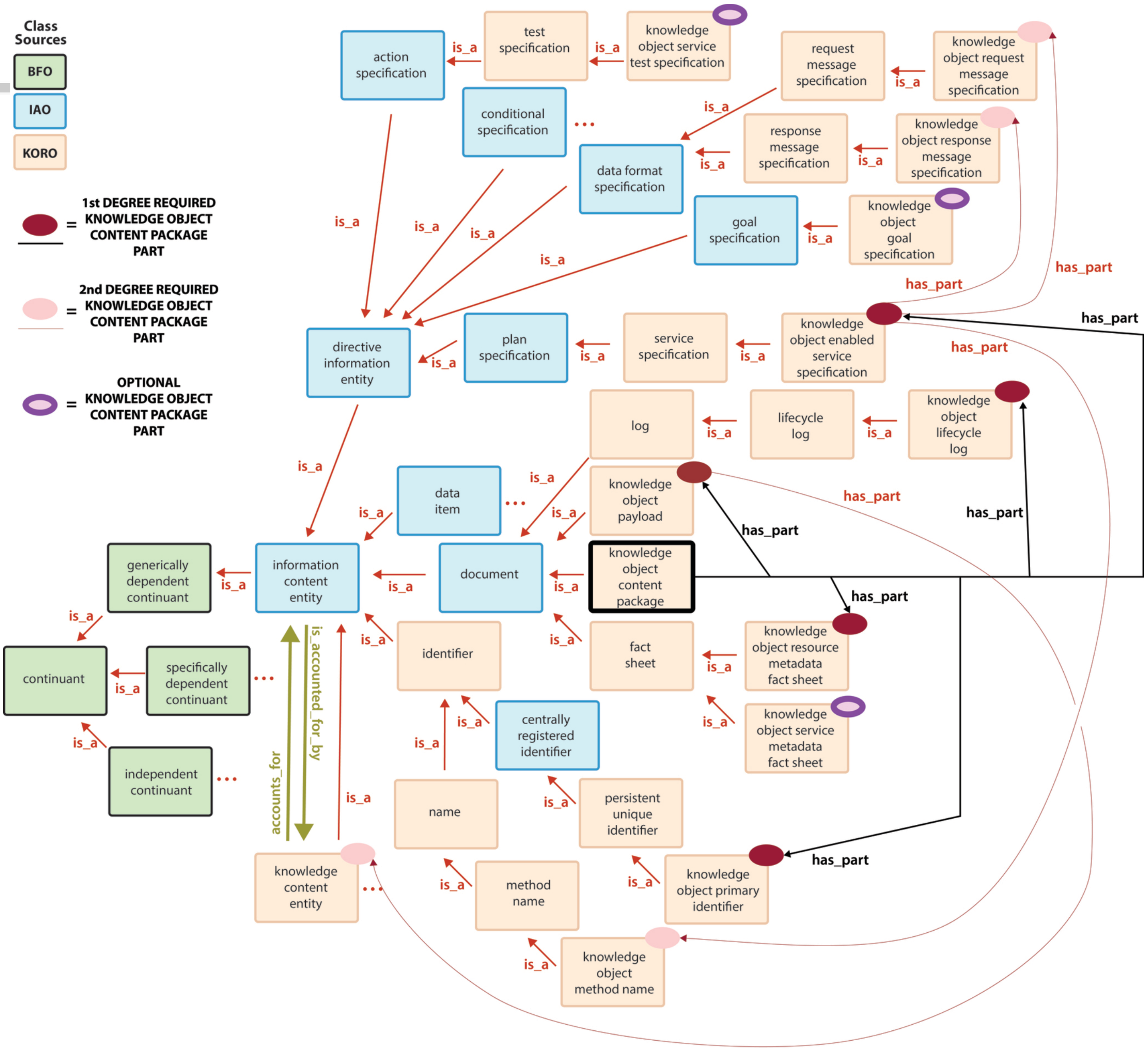
Class Sources
BFO
IAO
KORO
1st DEGREE REQUIRED KNOWLEDGE OBJECT CONTENT PACKAGE PART
2nd DEGREE REQUIRED KNOWLEDGE OBJECT CONTENT PACKAGE PART
OPTIONAL KNOWLEDGE OBJECT CONTENT PACKAGE PART
action specification
test specification
knowledge object service test specification
request message specification
knowledge object request message specification
conditional specification
data format specification
response message specification
knowledge object response message specification
goal specification
knowledge object goal specification
directive information entity
plan specification
service specification
knowledge object enabled service specification
log
lifecycle log
knowledge object lifecycle log
data item
knowledge object payload
generically dependent continuant
information content entity
document
knowledge object content package
continuant
specifically dependent continuant
identifier
fact sheet
knowledge object resource metadata fact sheet
knowledge object service metadata fact sheet
centrally registered identifier
independent continuant
name
persistent unique identifier
knowledge content entity
method name
knowledge object primary identifier
knowledge object method name
is_a
has_part
accounts_for
is_accounted_for_by
...

Figure 3. Portion of KORO indicating the required and optional parts of a Knowledge Object Content Package

LRH2_10054_F3.jpg

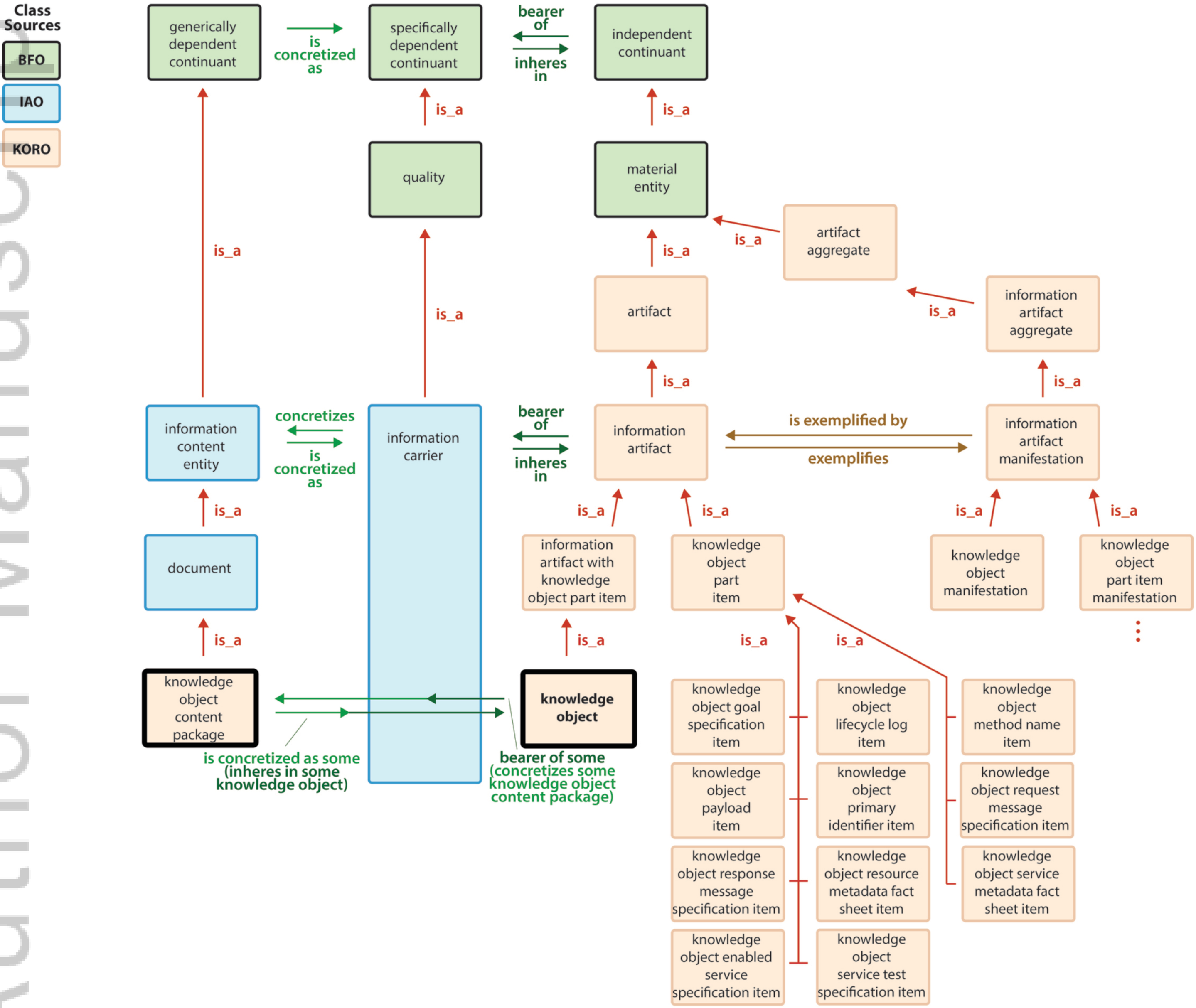
Class Sources
BFO
IAO
KORO
generically dependent continuant
is concretized as
specifically dependent continuant
bearer of
inheres in
independent continuant
is_a
quality
material entity
artifact aggregate
artifact
information artifact aggregate
information content entity
concretizes
is concretized as
information carrier
bearer of
inheres in
information artifact
is exemplified by
exemplifies
information artifact manifestation
document
information artifact with knowledge object part item
knowledge object part item
knowledge object manifestation
knowledge object part item manifestation
knowledge object content package
knowledge object
is concretized as some (inheres in some knowledge object)
bearer of some (concretizes some knowledge object content package)
knowledge object goal specification item
knowledge object lifecycle log item
knowledge object method name item
knowledge object payload item
knowledge object primary identifier item
knowledge object request message specification item
knowledge object response message specification item
knowledge object resource metadata fact sheet item
knowledge object service metadata fact sheet item
knowledge object enabled service specification item
knowledge object service test specification item

Figure 4. Portion of KORO indicating the whole and parts of a Knowledge Object with key relationships

LRH2_10054_F4.jpg