The Knowledge Grid: A Platform to Increase the Interoperability of Computable Knowledge and Produce Advice for Health

by

Allen J. Flynn

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information Science)
in the University of Michigan
2018

Doctoral Committee:

Professor Carl Lagoze, Chair
Professor Julia Adler-Milstein, University of California San Francisco
Professor Charles P. Friedman
Professor F. Jacob Seagull

Allen J. Flynn

ajflynn@umich.edu

ORCID iD: 0000-0002-3471-3063

DEDICATION

This dissertation is dedicated to all people who give advice about health to other people. May your health advice for others be formulated with the least possible amount of uncertainty.

ACKNOWLEDGEMENTS

David Barrett, the singer-songwriter who just happens to be my neighbor, once quipped that, “it takes a lot of help to make it on your own.” This truth could not be more apparent in the scientific work of this dissertation. This work has been a team effort.

Starting with my family, I would like to acknowledge the enduring emotional and intellectual support of my wife, Susan (Carbeck) Flynn. Next, I wish to thank my parents, Gordon and Beverly Flynn, for their sustaining support and encouragement. Further, for their patience and encouragement I am very thankful to my immediate family members: Joanne, Fred, Teddy, and Alice Keeling; Lisa, Erik, Annika, and Greta Zaar; Rick, Pam, Emma and Nick Flynn; Ava, and Addison Scigliano; and Diane Tooman. Also, I am blessed to have many extended family members, friends, and colleagues who have supported me in this effort and am very grateful for their support too.

Next, I acknowledge and thank the members of my dissertation committee. From Chuck Friedman, I have learned innumerable valuable scientific and life lessons. One of these is to take enough time to think things through carefully and thoroughly. From Carl Lagoze, I have learned about the amazing changes taking place in scholarly production and how to think about them scientifically. Julia Adler-Milstein has taught me the importance of asking good and incisive questions. Jake Seagull helped me refine many of my ideas by focusing on their scientific merits.

The work reported in this dissertation could not have been completed without the efforts and collaboration of the Knowledge Grid team, the staff, faculty, and students of the Schools of Information, Medicine, and Pharmacy, and many other members of the University of Michigan community. Thank you all!

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER III

CHAPTER IV

ABSTRACT

Here we demonstrate how more highly interoperable computable knowledge enables systems to generate large quantities of evidence-based advice for health. We first provide a thorough analysis of advice. Then, because advice derives from knowledge, we turn our focus to computable, i.e., machine-interpretable, forms for knowledge. We consider how computable knowledge plays dual roles as a resource conveying content and as an advice enabler. In this latter role, computable knowledge is combined with data about a decision situation to generate advice targeted at the pending decision.

We distinguish between two types of automated services. When a computer system provides computable knowledge, we say that it provides a *knowledge service*. When computer system combines computable knowledge with instance data to provide advice that is specific to an unmade decision we say that it provides an *advice-giving service*. The work here aims to increase the interoperability of computable knowledge to bring about better knowledge services and advice-giving services for health.

The primary motivation for this research is the problem of missing or inadequate advice about health topics. The global demand for well-informed health advice far exceeds the global supply. In part to overcome this scarcity, the design and development of Learning Health Systems is being pursued at various levels of scale: local, regional, state, national, and international. Learning Health Systems fuse capabilities to generate new computable biomedical knowledge with other capabilities to rapidly and widely use computable biomedical knowledge

to inform health practices and behaviors with advice. To support Learning Health Systems, we believe that knowledge services and advice-giving services have to be more highly interoperable.

I use examples of knowledge services and advice-giving services which exclusively support medication use. This is because I am a pharmacist and pharmacy is the biomedical domain that I know. The examples here address the serious problems of medication adherence and prescribing safety. Two empirical studies are shared that demonstrate the potential to address these problems and make improvements by using advice. But primarily we use these examples to demonstrate general and critical differences between stand-alone, unique approaches to handling computable biomedical knowledge, which make it useful for one system, and common, more highly interoperable approaches, which can make it useful for many heterogeneous systems.

Three aspects of computable knowledge interoperability are addressed: modularity, identity, and updateability. We demonstrate that instances of computable knowledge, and related instances of knowledge services and advice-giving services, can be modularized. We also demonstrate the utility of uniquely identifying modular instances of computable knowledge. Finally, we build on the computing concept of pipelining to demonstrate how computable knowledge modules can automatically be updated and rapidly deployed.

Our work is supported by a fledgling technical knowledge infrastructure platform called the Knowledge Grid. It includes formally specified compound digital objects called Knowledge Objects, a conventional digital Library that serves as a Knowledge Object repository, and an Activator that provides an application programming interface (API) for computable knowledge.

The Library component provides knowledge services. The Activator component provides both knowledge services and advice-giving services.

In conclusion, by increasing the interoperability of computable biomedical knowledge using the Knowledge Grid, we demonstrate new capabilities to generate well-informed health advice at a scale. These new capabilities may ultimately support Learning Health Systems and boost health for large populations of people who would otherwise not receive well-informed health advice.

INTRODUCTION

OVERVIEW

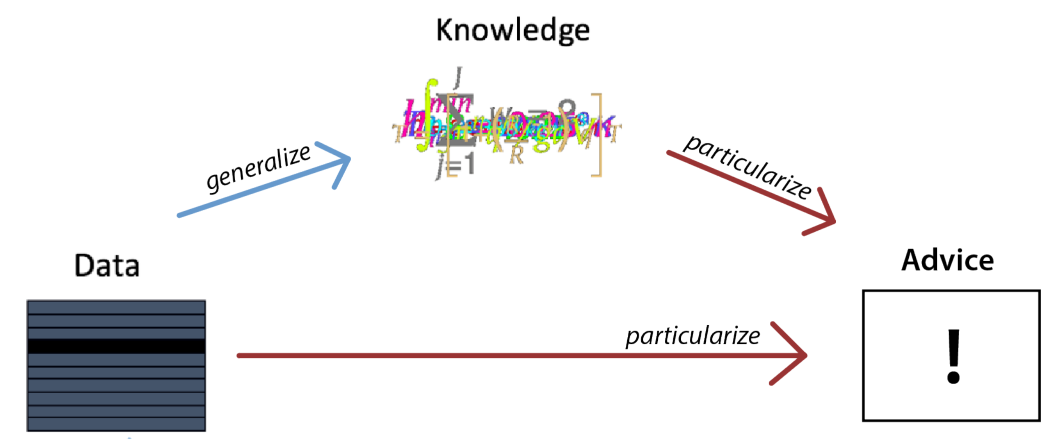
Throughout this dissertation, I describe moves towards a new technical infrastructure to generate large amounts of well-informed health advice. A bit later, the difficult problems motivating this work are discussed. Before that, I provide an overview for the reader.

This work is deeply rooted in the sciences of advice. What various sciences have to say about the concept of advice is the focus of Chapter I. In that chapter I establish a theoretical definition that a message of advice is an information object targeted at an unmade decision. In addition, I give a content-based framework for advice. This framework makes it possible to analyze advice-giving systems and advice-giving services in new ways.

The rest of this work, Chapters II, III, and IV, comprise a trilogy focused on increasing the interoperability of computable knowledge. But what is the link between advice and increasing the interoperability of computable knowledge?

Figure A below portrays data, knowledge, advice and certain links between them. I show in Chapter I that advice is any information targeted by an advisor at an unmade decision faced by a decision-maker. That is why Figure A depicts both data and knowledge being particularized to form advice. The relevant point here is that the process of producing advice is a process that particularizes, meaning fits or applies, data and/or knowledge to the specific context of an unmade decision. Precisely how this particularization happens is not relevant at the moment. What is critically important to note is that some advice derives from knowledge. This relationship between knowledge and advice is precisely the link that holds this dissertation together. The more the interoperability of computable knowledge is increased, the easier it should become to particularize knowledge computationally to produce messages of advice.

Figure A. Links between Data, Knowledge, and Advice



The trilogy of chapters on computable knowledge interoperability unfolds like this. In Chapter II, the architecture of a promising stand-alone advice-giving system is analyzed in hindsight to better understand some limitations that restrict the interoperability of its computable biomedical knowledge. In Chapter III, an account of the design and development of the Knowledge Grid platform is given. The technical components of this knowledge infrastructure platform provide potential means to increase the interoperability of computable knowledge. In Chapter IV, a modularized data-to-advice pipeline is designed, developed, and tested. This pipeline uses components of the Knowledge Grid to create, modularize, identify, update, and combine unique instances of computable biomedical knowledge automatically. The success of this data-to-advice pipeline trial shows that it is possible to use the Knowledge Grid to increase the interoperability of computable biomedical knowledge. In the end, this work moves us a step closer to a technical knowledge infrastructure that is capable of generating large amounts of well-informed health advice on demand.

While parts of this are my own work, other parts have been accomplished by the teams with whom I work. Therefore, statements of attribution shift from "me" to "we" throughout.

MAIN CONTRIBUTION AND MOTIVATION

For our main contribution, we show how the interoperability of computable knowledge can be increased in ways that facilitate shareable knowledge services and advice-giving services. The work we present here is domain agnostic. Any computable knowledge can be treated this way. However, our focus here is exclusively in the domain of biomedicine because our overarching goal is to improve human health. Therefore, when speaking generally, I frequently use the term *computable knowledge*, but, when speaking about the health use cases we are pursuing, I use the more specific term *computable biomedical knowledge*.

This work is motivated by a shortage of well-informed health advice.[1–4] Here, advice is *any* information object targeted at an unmade decision. This perspective is elaborated in Chapter 1. By well-informed advice, we mean advice that reflects current best practices, which in turn are based on empirical scientific knowledge. Hence, well-informed advice is information, based on some knowledge in the world, that is given as a message to inform an unmade decision.[4,5]

Manual methods of giving health advice are insufficient. There are too few health experts with too little available work-time to provide enough well-informed health advice for all people.[6,7] This problem is compounded by rapid growth in the quantity of scientific biomedical knowledge.[8] For these reasons, automated methods are needed.[9] Medical experts indicate that automation is the key to meeting the global demand for expert advice to improve health.[10–12]

MAJOR CHALLENGES

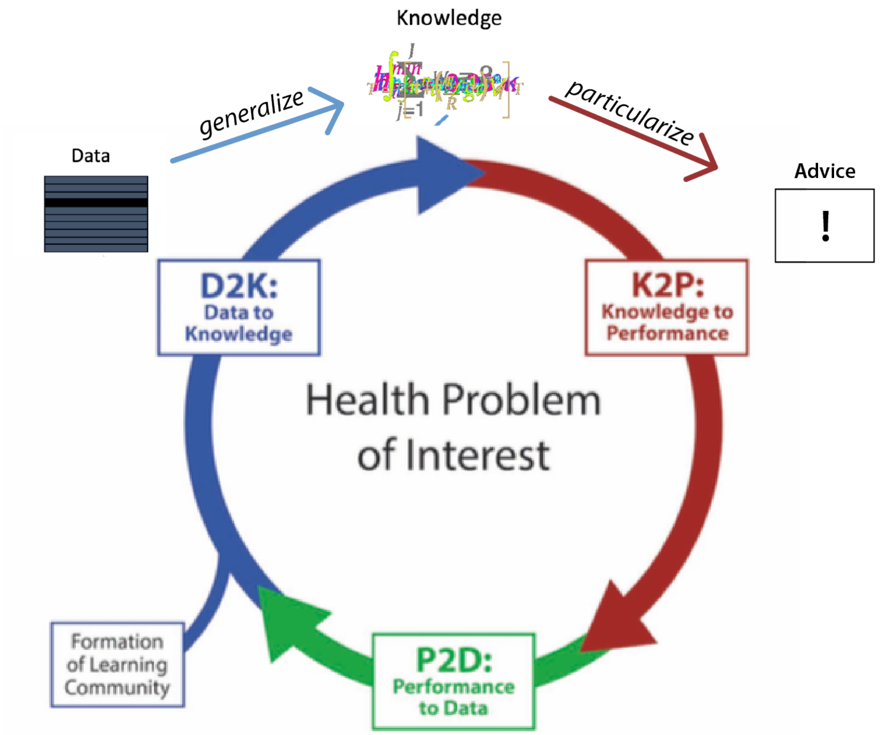
There are major challenges that have to be overcome to successfully automate advice-giving for health on a large scale. Some of these major challenges are conceptual. Science has yet to settle the questions of precisely what advice is or how advice relates to data, information, and knowledge (see Figure A).[13] Other major challenges are technical and infrastructural. One is to devise knowledge infrastructure that can support knowledge services and advice-giving services capable of generating sufficient quantities of health advice to meet demand. Embedded in this infrastructural challenge is the need to increase the interoperability of computable biomedical knowledge, which is the primary goal of this work.

According to Edwards, *knowledge infrastructure* is, "a sociotechnical system that collects data, models processes, tests theories, and ultimately generates a widely shared understanding."[14] Edwards, a historian of climate change science, focuses on societal learning, where the goal is to achieve a widely shared understanding among the public.[14] As a sociotechnical phenomenon, an instance of *knowledge infrastructure* unites technology with socially-determined norms and standards and with related technical and social processes to make knowledge, in any form, findable, accessible, interoperable, and reusable (the FAIR principles).[15]

Another major challenge we face is that health knowledge, in particular, often needs to be understood by only a few people whose health depends on it. For better population health, the goal of using knowledge infrastructure is sometimes to achieve a widely shared understanding[14], but just as often it is to achieve an *easily shareable understanding*. An *easily shareable understanding* is one that is kept updated and quickly found and accessed on-demand by individuals just when they need it. This difference can be clarified with a health example. Consider the case of multiple-system atrophy (MSA), a rare and fatal disease.[16] The few who are afflicted with MSA, and their families, have an obvious need for well-informed health advice about it. Yet for most other people there is little or no need to reach a shared understanding about MSA. So, another major challenge here is to be able to have the most current biomedical knowledge – spanning most or all diseases, tests, treatments, and health behaviors – in a ready-to-be-applied format in order to provide best practice advice to individuals on demand.

Finally, this work addresses another major challenge, which is to develop technical infrastructure to support Learning Health Systems (LHSs). LHSs learn about human health in a systematic way.[17] They rely on communities of interest to generate and apply new biomedical knoweldge.[12,18–20] Here we explore how to achieve potentially better knowledge services and

Figure B. The Learning Cycle Model



advice-giving services for LHSs. Taking a macro view, these services are parts of a much larger sociotechnical infrastructure needed by LHSs to support ongoing processes of learning.[19,20]

At a micro level, it is postulated that LHSs instantiate *learning cycles*, which unite four activities that learning communities do to achieve continuous learning about a problem of interest to them (Figure B).[19] The four steps of a learning cycle are illustrated in Figure B above.

First, on the lower left, the learning cycle begins when a learning community forms around a problem of interest to them. Second, the community analyzes or deliberates over data relevant to their problem of interest and generates results as new knowledge. This process is labeled "D2K", which stands for "Data to Knowledge." Third, to improve performance, the community implements and acts on its new knowledge. This process is labeled "K2P", meaning "Knowledge to Performance." Fourth, once new knowledge has been implemented and acted on, the learning community collects additional data to study the impacts of implementing its new knowledge on the problem of interest. This is a process of data collection labeled "P2D", which stands for "Performance to Data." By repeatedly doing the work of D2K, K2P, and P2D, performance is iteratively improved.

Embedded in the learning cycle model portrayed in Figure B are the same links between data, knowledge, and advice that were introduced before. LHSs need infrastructure to take data and generalize it to get knowledge. They also need infrastructure to take knowledge and particularize it to get advice. Here we address the major challenge of first devising and then starting to develop the knowledge infrastructure needed by LHSs to systematically particularize computable biomedical knowledge and generate health advice.

BACKGROUND AND SIGNIFICANCE

Lack of well-informed health advice

Problems resulting from a lack of well-informed health advice are plain to see. A current example is the epidemic of opioid-related drug-overdose deaths in the United States.[21] Some of these avoidable deaths result from misunderstanding best practices for the use of opioids to treat pain.[22] Others result from the use of illicit "street drugs."[23] Still others result from an incapacity of the health system to systematically treat drug addiction as expert addictionologists do.[24] The absence of ready advice, predicated on expert knowledge about opioid drugs and their use to treat pain and overcome addiction, hinders many physicians, nurses, pharmacists, dentists, and others in their efforts to use opioids safely and effectively.[23] The ongoing opioid epidemic in the United States highlights, in a woeful way, our need for better knowledge infrastructure to improve the flows of reliable, valid knowledge and the health advice we derive from it.

There are, unfortunately, many more indicators of the lack of sufficient, well-informed health advice. These indicators include clinical *knowledge gaps* that are apparent in everyday medical practice, *information overload* in clinical practice, evidence that predictable *cognitive biases* can negatively influence medical decision-making, and a lack of *care coordination*. Based on the totality of this evidence, it is reasonable to conclude that there is a substantial unmet need for well-informed, readily-available health advice. The scientific evidence supporting this conclusion is reviewed in greater detail next.

In a study of *knowledge gaps* in clinical practice from 1985, Covell et al. described the unmet information needs of physicians.[25] They found that only 30% of physicians' information needs were being met. This study measured the proportion of answers available to "highly specific" questions about "patient management" arising in the minds of clinicians while providing care.[25] Some of the questions that arose in practice were matters of fact, others of opinion. In 2007, in a study of 3,511 patient consultations, González-González estimated that only 20% of primary care physician information needs were being met.[26] These findings suggest that clinical *knowledge gaps* appear routinely and persist in everyday clinical practice.[27]

Information overload occurs when the quantity of potentially useful information received by an individual becomes a hindrance because the burden of processing it cannot be overcome.[28] Beasley et al. claim primary care physicians regularly experience information chaos, which includes *information overload*, underload, scatter, conflict, and falsity.[29] Markman describes how the number of relevant inputs needed to choose an appropriate oncology treatment overwhelms practitioners' ability to identify and assimilate them.[30]

Advice has the potential to counter information overload only if it brings forward actionable, contextualized information that is better than the information clinicians already have.[9,11] Toward this end, Stead et al. envision intuitive, assistive computer systems that work to coordinate the thinking of multiple experts by providing them with "patient centered cognitive support."[11] Such systems would be capable of rendering well-informed, relevant health advice using commonly understood, highly visual, easy-to-grasp, anatomic, physiologic, and pathologic models of people and disease.[11] Computable biomedical knowledge has to be very well organized to do this.

Probability neglect is a known cognitive bias that affects medical practice and could be counteracted by well-informed health advice. Sunstein defines *probability neglect* as, "emotional interference with judgment about risk", causing some risks to be overlooked and others to be overestimated.[31–33] To counter *probability neglect*, advice could indicate how to incorporate probabilities into a decision-making process,[31] or advice could convey the gist of the available evidence in a manner that overcomes emotional interference.[34]

Outcomes bias is another known cognitive bias. It happens when decision makers judge the quality of their decisions according to the outcomes that result from them.[35] Advice could be used to counteract this by communicating how to evaluate decisions regardless of outcome.

The need for more and better health advice can be inferred from the number of preventable adverse events (PAEs) occurring due to medical errors. PAEs are unintended injuries caused by medical management resulting in measurable disability. Leape et al. estimated the incidence and preventability of adverse events effecting hospital patients.[36] Using 30,195 randomly selected hospital patient records from 1984, they estimated that nearly 4% of patients suffered an adverse event.[36] Of these events, 42% were PAEs unassociated with negligence. Another 6% were potentially preventable, save for lack of expertise. There is further evidence that PAEs continue to occur at similar rates today.[37]

Adding to these data about medical errors, Lesar, Briceland, and Stein demonstrated that the most common factors associated with medication prescribing errors involve missing knowledge, or the misapplication of knowledge, in 30% of cases.[38] These findings suggest a significant role for advice in preventing medication errors. These results were cited in the Institute of Medicine's seminal report To Err is Human.[39] This report called on organizations to (1) "avoid reliance on memory" by adopting "protocols and checklists", (2) to "use constraints and forcing functions", and (3) to "improve access to accurate, timely information."[39] In other words, the Institute of Medicine recommends that well-informed health advice be made more readily and widely available. In this report, they emphasize medication safety, just as we do here.

If, as Jureta asserts[13], "advice is a tool of coordination", then, where advice is lacking, some degree of discoordination should be expected. This is the case in health care. Sadly, reports of bad outcomes resulting from poor care coordination are plentiful.[40,41] The Institute of Medicine reviewed a large body of evidence and concluded, "care delivery has become increasingly fragmented leading to coordination and communication challenges for patients and clinicians."[42] Therefore it is reasonable to attribute a measure of poor coordination of health care activities to a lack of well-informed health advice.

All of this evidence probably underestimates the need for well-informed health advice because of the rapid growth in the quantity of available biomedical knowledge. This topic is discussed next.

Rapid growth in biomedical knowledge

In 1976, Pauker et al. estimated family-practice physicians must have one million facts at their disposal for medical decision-making.[43] They speculated that medical specialists require access to two million facts.[43] These past estimates of the number of facts individual practitioner's must command to support their medical practices may be gross underestimates today.

A simple way to document the growth in our overall corpus of biomedical knowledge is to examine the number of scientific biomedical journal articles being indexed each year in the National Library of Medicine's online digital library, PubMed. Table A indicates that, overall, with fluctuations, the number of new articles indexed in PubMed annually has been increasing steadily over the previous two decades, since 1997 (2.7% average growth per year).[44]

Table A. Annual Number of Scientific Biomedical Articles Indexed in PubMed
(Source: https://www.nlm.nih.gov/bsd/index_stats_comp.html)

1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
519,012	411,921	434,000	442,000	463,000	502,000	526,000	571,000	606,000	623,089
2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
670,943	671,904	712,675	699,420	724,831	760,903	734,052	765,850	806,326	869,666

Meanwhile, new open access *knowledgebases* about health are also appearing from reputable sources.[45,46] Here, a *knowledgebase* is defined as a store of computable knowledge. One example of a new public *knowledgebase* comes from the Clinical Pharmacogenetics Implementation Consortium (CPIC).[47] Included amongst CPIC's published online files are spreadsheets describing the genetic determinants of drug-gene associations. These spreadsheets contain approximately 2,000 drug selection and dosing rules.[45] These numerous drug-gene rules represent 0.2% of the number of facts that Paulker estimated physicians must know to practice medicine effectively in 1976[43], but they cover only a miniscule amount of the actionable drug knowledge that is available today.

Not only has the overall corpus of scientific knowledge about biomedicine grown rapidly since 1976, but health facts are now subject to more rapid change.[48] While patients and their families expect doctors to know what to do to meet their medical needs - and how to do it - Weed commented several decades ago in 1983 that, "the unaided mind cannot meet such expectations."[49] Therefore, it is evidently well-established that computer systems and information technology are now required to aid people by providing health advice.

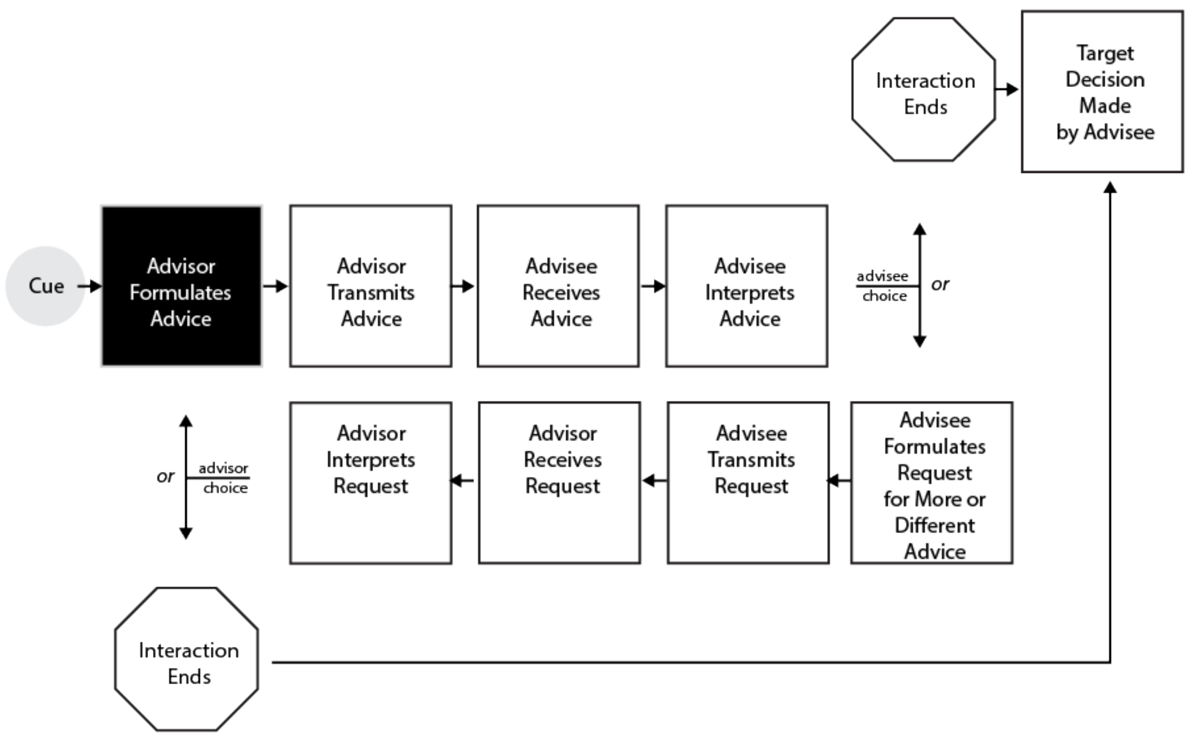
Next, we introduce the fundamental concepts of an *advice interaction*, an *advice-giving system*, and an *advice-giving service*, and then, to complete this dissertation, we briefly review some of the science pertaining to advice-giving systems for health.

Model for an advice interaction

Figure C below portrays a conceptual model of a two-way, advisor-advisee advice interaction. I developed this model in 2014 after reviewing the scientific literature on advice. This model is directly informed by the works of Jureta[13], Salacuse[50], Friedman[51], and Shannon.[52] It includes a series of actions prompted by a cue, which is perceived by the advisor and serves to trigger the advisor to begin formulating advice for an advisee. Formulation of advice is a critical step in an advice interaction preceding the transmission (or communication) of advice. Together these acts of formulation and transmission of advice are referred to throughout this work as *generating or producing* advice. Note that, according to the model in Figure C below, advice interactions can cycle back and forth between advisor and advisee before halting.

Stepping through this advice interaction model in Figure C from the beginning, first the advisor is prompted by an initial cue (the cue may or may not be an initial request for advice from the advisee). The cue triggers the advisor to formulate advice. Next the advice about a target decision has to be transmitted by communicating a message. This is how advice is

Figure C. Model of an Advice Interaction with the Potential to Cycle Before Ending



"given" to an advisee. Advice is then received and interpreted by the advisee. Next, the advisee chooses whether to proceed and make the target decision, while applying or ignoring the advice received, or whether to request more or different advice from the advisor.

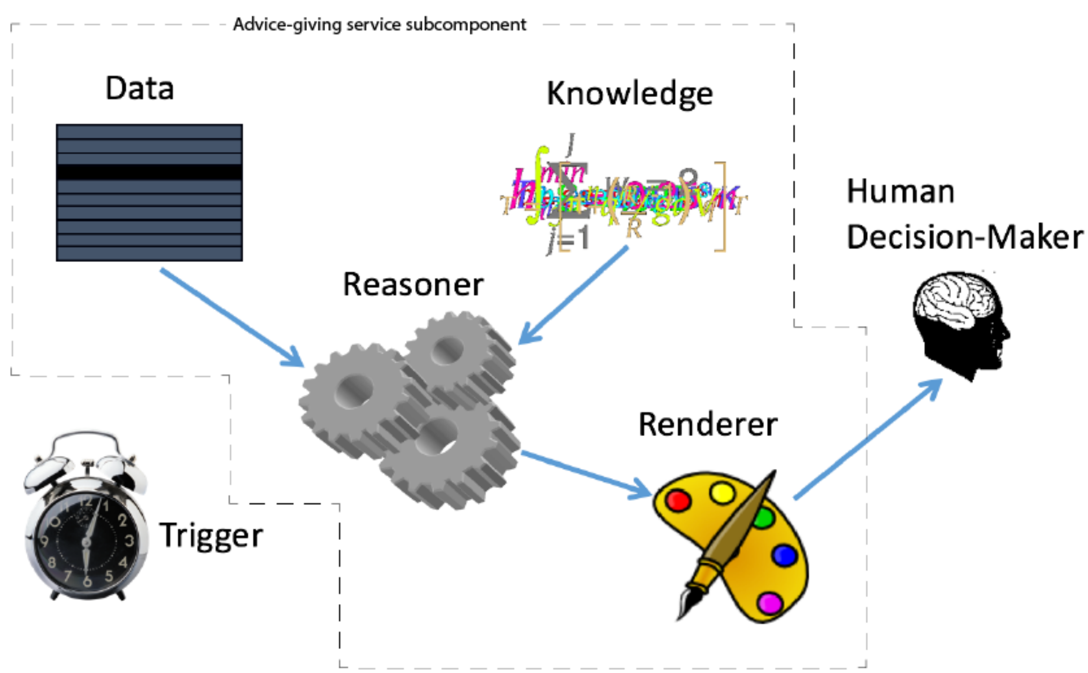
In response to a request for more or different advice, which is a cue, the advisor may choose whether or not to provide more or different advice. According to the model, an advice interaction ends when either the advisee or advisor halts it at the choice point under their control. After the advisor-advisee interaction ends, the advisee's yet unmade decision, at which the advice given is targeted, finally gets made.

There is no limit to the amount of communication that can take place within an advice interaction according to the advice interaction model. The model illustrated in Figure C greatly simplifies advice-giving and advice-receiving. In the real-world, additional complexities pertain. Real advice interactions can be interrupted and restarted. Advisors and advisees can change roles or participate in multiple, concurrent advice interactions with different target decisions. Sometimes, decisions at which advice is targeted may be deferred or avoided altogether. The model in Figure C provides us with just enough detail to consider what it means to have an automated advice-giving system, or an advice-giving service, function in the role of advisor.

Model of an advice-giving system and of an advice-giving service

Throughout this dissertation, the terms *advice-giving system* and *advice-giving service* are used. Here, the term advice-giving system is an over-arching term for complete decision-support systems, recommendation systems, and expert systems, among others. For a system to be considered an advice-giving system it must automatically particularize some knowledge in the world, using contextual information about a decision situation, to generate messages of advice that are specific to some decision situation.

Figure D. Friedman's Model of an Advice-giving System



Advice-giving systems play the role of automated advisor. Via private communication, Friedman has provided a six-part conceptual model of an advice-giving system (Figure D). As depicted above, the first required capability is the Trigger capability. Advice-giving systems must include a Trigger capability to enable outside agents, either people or machines, to initiate advice-generation by the system. The Trigger capability is recognized among clinical decision support experts as a critical, scope-limiting component of advice-giving systems.[49,53,54] Therefore, a couple of additional points specifically about the Trigger capability are warranted.

In 1983, Weed and Hertzberg presaged context-aware advice-giving system Trigger capabilities in a paper about "Problem-Knowledge Couplers."[49] Their couplers enabled knowledge to be used to make advice appear automatically under predetermined conditions.[49] Since that time, efforts to develop and improve the Trigger capability for advice-giving systems have continued. One current example is the "CDS Hooks" project. CDS Hooks seeks both to expand and standardize automated triggers (also called "hooks") within health information systems, especially electronic health record (EHR) systems.[54]

Figure D shows how, once an advice-giving system is triggered to generate advice, it combines two different types of inputs, data and knowledge. In Friedman's model, input data (i.e., "Data") describe something related to a decision situation. These "instance data" may describe a single person, one biological sample, the current state of an organization, etc. These descriptions are the data which, when combined with computable knowledge, result in advice.

In addition, advice-giving systems must also draw on computable knowledge (i.e., "Knowledge"). Per Friedman, knowledge is a result from a previous analytic or deliberative process of study that is significant to a community.[19] Methods of knowledge representation are used to encode knowledge to make it computable so that advice-giving systems can use it.

All advice-giving systems apply computable knowledge to data, through processes of reasoning and computation, to generate advice. In Friedman's model, the Reasoner capability, depicted as a set of gears in the middle of Figure D, includes formal logical reasoning, pattern-matching, calculation using statistical prediction rules, and other computational functions.

Viewed from this most general perspective, advice-giving system Reasoners come in many forms and are called by many names, e.g., rules engine, reasoning engine, or interpreter.

The fifth required capability of all advice-giving systems is the Rendering capability, indicated by a painter's palette in Figure D. After data and knowledge are combined in the Reasoner and a new result is generated, advice-giving systems must then formulate messages of advice and communicate them in verbal and/or visual formats.

Finally, Friedman's model of an advice-giving system includes a sixth component, the Human Decision-maker. Including the Human Decision-maker in the model makes it clear that advice-giving systems provide advice to people. However, as we will see in Chapter I, advice-giving systems can also provide advice to other machines.

An *advice-giving service* is the subcomponent of an advice-giving system that does the work of formulating and transmitting messages of advice. In Friedman's general model of an advice-giving system (Figure D), the advice-giving service subcomponent encompasses only the "Data", "Knowledge", "Reasoner" and "Renderer" parts, as indicated by the dashed grey line in the figure. Advice-giving systems may have more than one advice-giving service subcomponent. As we will see, some of the work presented here enables the advice-giving service subcomponents of advice-giving systems to be modularized so they stand on their own.

There are many studies of health advice-giving systems built in accordance with Friedman's model (Figure D). Much of the science about these systems is organized under the rubric of Clinical Decision Support Systems or CDSS.[53] CDSS are one category of advice-giving systems, and they are an important category for health. However, in addition to CDSS, other advice-giving systems for health, notably those that generate advice for consumers, including patients, their family members and caregivers, have also been developed and studied.[55] Central findings about the benefits and limitations of both of these types of advice-giving systems, CDSS and consumer-oriented systems for health, are reviewed next.

Advice-giving systems for health

In a synthesis of 17 well-executed systematic reviews, Jasper et al. reported in 2011 that, "there is significant evidence that CDSS can positively impact healthcare providers' performance, with drug ordering and preventive care reminder systems as the most clear [sic] examples."[53] However, what these authors also make clear is that, as of 2011, few studies of CDSS to date have demonstrated any benefits to patient outcomes.[53]

There is even less evidence showing benefits from consumer-oriented advice-giving systems. In 2014, Payne et al. performed a systematic review of mobile applications (apps) used to support health behavior change, resulting in an analysis of 24 studies.[55] Most of these studies focused on increasing physical activity and/or weight loss, while a few focused on better mental health. Making it more difficult to determine the impact of these apps, many of the studies analyzed involved a multi-component strategy for health behavior improvement that included the use of advice-giving apps along with other interventions, such as personal counseling. Further limiting evidence of positive benefits from advice-giving health apps for consumers, most studies to date are pilot studies conducted with small samples showing only the potential for real-world health gains.[55]

An orthodox interpretation of the central finding about advice-giving systems for health, which is that these systems have yet to result in widespread health outcomes improvement, could be that advice-giving systems are of uncertain value and need to be further studied in larger populations for longer periods of time. This view is supported by evidence.

A somewhat less conservative interpretation of this central finding, made in light of the need for more well-informed health advice, is that the modest health benefits so far demonstrated from using advice-giving systems are enough to show that these systems are moderately effective today and likely to become more effective in the future. This interpretation is bolstered by the views of health care thought leaders[9–12] and reflected in national health IT policy in the United States, which promotes the use of advice-giving systems.[56]

Yet another interpretation of this central finding, the one that is most relevant here, is that knowledge services and advice-giving services capable of providing high-quality advice that is sophisticated enough to be helpful and made available at the scale needed to improve health outcomes, have yet to be designed, developed, and deployed. This interpretation rests on evidence that there are known limitations of current advice-giving systems that can potentially be overcome in part by increasing the interoperability of computable biomedical knowledge.[9,57,58]

Limitations of current advice-giving systems for health

Current advice-giving systems, including most CDSS, are limited in their impact by the narrowness of their scope, inadequate upkeep of knowledge, the large scope of work needed to implement them, the fact that they are naïve to what their users already know, and their cost.[59]

To date, many task-oriented, domain-specific advice-giving systems for health have been developed and deployed.[53] Examples include CDSS to guide antibiotic selection and dosing[60], CDSS to warn responders about gradual clinical worsening of patients in the hospital[61], and CDSS to guide the use of radiological imaging.[62] These domain-specific advice-giving systems have their own unique, often closed and proprietary, knowledgebases.

One of the ways that narrowly scoped advice-giving systems fail is by not accounting for common real-world complexities related to health. Examples of this are cases of multiple diseases, also called multiple comorbidities. Many advice-giving systems for health are purpose-built to advise about a single disease, condition, treatment, or intervention. Advice-giving systems with such narrow scope are incapable of reconciling the advice they give about one topic with advice about other related topics. To address this problem, some research has been done on automatically merging multiple disease-specific computable clinical practice guidelines.[63]

In contrast to advice-giving systems built using narrowly scoped knowledgebases, systems with a broader scope of computable biomedical knowledge have been envisioned. Tsafnat and Coiera propose modular, composable or combinable advice-giving services arising from individual instances of computable knowledge focused at the molecular, organelle, cellular, organ, or whole-body levels.[64] Similarly, Stead envisions computer-supported continuous learning and performance environments for medical providers.[11,12] Such environments would track and assess provider knowledge while also directly supporting provider performance with "just in time" access to well-informed advice provided to them in their clinical workflow."[12]

Another limitation of today's advice-giving systems is that the knowledgebases they use are difficult to keep current. Ash et al. reported in 2007 that keeping the knowledge in CDSS current is difficult, even for well-resourced organizations.[65] Reinforcing this point, Campbell et al. noted that CDSS upkeep is unending, requiring a permanent commitment of time and attention.[66] One strategy for achieving more efficient knowledge upkeep is to distribute the work of knowledge maintenance across a group of domain experts. Geissbuhler and Miller described this approach as the "Knowledge Library" model of distributing knowledge maintenance for CDSS in 1999.[67] Similarly, Learning Health Systems depend on communities of interest, focused on particular health problems, to steward computable knowledge assets that can be shared on a local, regional, national, or international scale.[19]

The effort, in terms of time and attention, required to implement advice-giving systems is another limitation to their use. The work of advice-giving system implementation can be broken down into two categories. The first category is the *technical work* needed to turn these systems on. The second category is the *process change work* needed to make use of these systems in practice. A large portion of the technical work involves engineering and testing interfaces between advice-giving systems and other systems.[68] Research and development efforts to reduce the burden of interface engineering and testing are ongoing.[69] Meanwhile, other efforts in implementation science address ways to improve and scale-up process change.[70]

Another limitation of today's advice-giving systems is that they can be difficult, if not impossible, to customize to meet local needs.[65] Some of this difficulty arises from a lack of granularity in the affordances that govern how the knowledgebases supporting advice-giving systems function. When the only available controls are to activate or deactivate knowledge categorically, instead of more granular controls over individual instances of computable knowledge, then advice-giving system users may face a trade-off. Either they enable the advice-giving system to provide wanted advice along with unwanted advice, or they forego wanted advice to avoid unwanted advice. Improved knowledge infrastructure could address this problem of granularity of control over the computable knowledge that supports advice-giving systems.[71]

Proprietary approaches to developing advice-giving systems that are scoped for narrow domains and use minimally interoperable computable knowledge thwart the scale-up of advice-giving capabilities that is needed today. The present work is significant because it addresses the need for better knowledge infrastructure, and especially for more highly interoperable computable biomedical knowledge, to scale-up automated advice-giving sufficiently to close the gap between the demand for well-informed health advice and the current inadequate supply of it.

COMPUTABLE KNOWLEDGE INTEROPERABILITY

As indicated in the Overview, we primarily address advice and computable knowledge interoperability in this work. Following a conceptual analysis of advice, the preponderance of this work focuses on increasing the interoperability of computable knowledge, especially computable biomedical knowledge, for the purpose of generating more and better health advice.

Definition of interoperability

Before going further, it is necessary to define, as precisely as possible, what is meant throughout this work by the term *interoperability*. David Chen and his colleagues discuss the definition of *interoperability* in the context of information system engineering.[72] Fundamentally, they note that *interoperability* is the ability for two or more arbitrary things to function jointly.[72] This definition of *interoperability* implies three important things that are detailed next.

First, *interoperability* is an *ability* that things may have which varies by degrees. For scientific purposes, to say that a thing is or is not interoperable is insufficient. Unfortunately, however, precise measures of *interoperability* are wanting. Therefore, *interoperability* is discussed here in categorical ways to convey a scale from low to high.

Second, the definition of *interoperability* above implies that the degree to which two things are *interoperable* depends on multiple factors and not on just one factor. Features of each thing involved, and aspects of their joint action, help determine the degree of *interoperability* in any particular case.

Third, the definition of *interoperability* above further implies that the *interoperability* of many things, including instances of computable biomedical knowledge, can potentially be increased or augmented in a variety of ways, not just in one way.

Previous scientific work on information artifact interoperability

Scientific work towards making digital information systems more highly interoperable is longstanding. We note that Cerf and Kahn published their protocol for packet-based network communication in 1974.[73] They were motivated by the need to increase the interoperability of computer systems enough to support sharing of information artifacts (or resources) by disparate systems over digital computer networks. Their seminal work has evolved into the *TCP/IP protocol suite* that enables computers to share information across the modern global Internet.

Scholarly information artifact interoperability. Past scientific work intended to augment the interoperability of scholarly information artifacts is most relevant here. In this regard, Lagoze and Van de Sompel provide a review of some of the earliest moves of the Open Archives Initiative (OAI).[74] The OAI set out to build an interoperability framework to improve scholarly artifact sharing among federated archives. These early moves included efforts, which were deliberately limited in scope, to expose the metadata describing scholarly information artifacts held in various different online archives. Work to expose and share these metadata stimulated the development of value added services that consumed and further processed these metadata describing scholarly artifacts. An important outcome of early work on the OAI was the protocol for metadata harvesting (PMH).[74] To enhance its interoperability, the PMH was based on the common Hypertext Transfer Protocol (HTTP) methods GET and POST.

As implied by their names, the HTTP GET method supports requests to get artifacts or resources from an archive while the POST method supports posting artifacts or resources to an archive. Using these HTTP methods, which take advantage of the World Wide Web (WWW) as a common infrastructure for information interoperability, it becomes possible to improve the interoperability of technical component-to-component interactions, to reduce interaction latency between technical components, and to facilitate technical solutions that address cross-cutting concerns, such as information system security. For these reasons, there has been substantial growth in the number of HTTP-based webservice application programming interfaces (APIs) available for use via the WWW. Work reported here to increase the interoperability of computable knowledge takes advantage of these same HTTP methods.

Web-centric scholarly information artifact interoperability. More recently, Van de Sompel and Nelson provided an update on the evolution of the Open Archives Initiative for increasing the interoperability of what they have come to call *web-based scholarship*.[75] They note that the PMH has achieved broad adoption since 2001. Metadata about scholarly artifacts is now routinely shared among scholarly archives using computer networks. However, they view the successful uptake of the PMH with some concern because it enabled archive-centric, instead of web-centric, interoperability gains. Archive-centric approaches emphasize archive-level interoperability over potentially more desirable artifact-level interoperability enabled by the web-centric approach.[75] In this regard, with respect to artifacts, during the period between 2001 and 2015, it became apparent that future attempts to improve the interoperability of artifacts of scholarly communication using the WWW could be even more successful if they focused on improving the interoperability of individual units of communication in the form of *compound digital objects* instead of units of communication that are scientific papers or monographs[76].

Compound digital objects. According to Kahn and Wilensky, who pioneered this idea, a *compound digital object* is a digital object with a unique identifier, or handle, that is composed of one or more instances of typed, structured "data" plus descriptive metadata.[77] We emphasize that when Kahn and Wilensky refer to "data", they are referring to any binary content and generally not to the data comprising a scientific data set. We are exclusively interested in increasing the interoperability of the computable knowledge class of scholarly artifacts. To be clear, the "data" of interest here, in the Kahn and Wilensky sense of the term "data"[77], is computable knowledge. We will refer to this "data" as computable knowledge from hereon to limit any confusion.

Van de Sompel and Nelson's insight about the importance of *compound digital objects* as units of scholarly communication has greatly influenced our approach to increasing the interoperability of computable knowledge.[75] Applying their insight, it was Lagoze who first introduced the idea of formally specifying a new class of compound digital objects to hold, steward, share, disseminate and facilitate deployment of instances of computable biomedical knowledge. Details about our work to formally specify a new class of compound digital objects appear in Chapter III.

This dissertation focuses on how the interoperability of computable knowledge may be increased for the purpose of producing more and better health advice. The previous three paragraphs have highlighted how a host of strong standards (TCP/IP, HTTP, PMH) have been used to augment the interoperability of scholarly resources via the WWW. Yet there are other ways to augment information artifact interoperability that must be considered too.

Other approaches to increasing information artifact interoperability. To augment information artifact interoperability, a shift away from repository-centric approaches towards web-centric and/or object-centric approaches is notable.[75] This shift is facilitated by the methods of HTTP, typed links, and compound digital objects.[75] Besides these methods, there are even more potential solutions to consider for increasing information artifact interoperability.

Paepcke et al. have described the information artifact interoperability solution space in an expansive way.[78] They delineate five clusters of approaches for augmenting information resource interoperability. These five clusters of approaches are *strong standards* (e.g., HTTP), *families of standards* (e.g., the family of scripting computer languages with members such as PERL and Python), *external mediation* (e.g., a text file to PDF converter), *specification-based interaction* (e.g., a RESTful webservice), and *mobile functionality* (e.g., a JavaScript module that can be executed by any modern Internet browser). The work we report here takes advantage of solutions from all five of these clusters of approaches.

Interest in interoperability within the greater health community

Meanwhile, over approximately the past two decades in the health community, the term "interoperability" has become a buzzword. Recently, a group of experts in health care delivery and health informatics convened by the National Academy of Medicine in 2016 published a clarion call for greater interoperability of health information technology.[79] They assert it is vital for the health system to have a more flexible and adaptable health information technology infrastructure. They promote two specific approaches for this: Enable end-user software applications to become readily "substitutable" and enable software-as-a-service by taking advantage of the HTTP-based methods of the WWW.

Also in 2016, another large group of experts, members of the Electronic Data Methods (EDM) forum, published a report calling for better interoperability of health information resources.[80] The EDM forum was convened by Academy Health under a cooperative agreement with the United States Agency for Healthcare Research and Quality (AHRQ). Their report describes a complex model of interoperability with four supporting layers: a *technical interoperability layer*, enabled by communication protocols, a *syntactic interoperability layer*, enabled by common structures for sharing information, a *semantic interoperability layer*, enabled by common interpretations of the meaning of shared information, and a *process interoperability layer*, enabled by methods that link health-related processes of care and activity to shared information with a common interpretation. Their perspective reinforces the notion that approaches to augmenting the interoperability of computable biomedical knowledge for health advice-giving systems have to be multiple and varied.

Definition of computable knowledge

Up until this point, we defined knowledge in its own right but have used the term *computable knowledge* loosely. We require a precise definition of *computable knowledge*. To arrive at this definition, we begin by stating what *computable knowledge* is not. Computable knowledge is not an arbitrary instance of freetext, even if the arbitrary instance of freetext conveys knowledge to a human reader. Computable knowledge is something other than that. But what is it? To define computable knowledge, we borrow conceptually from Soare and his predecessors in the mathematical domain of computability theory, primarily Turing. They equate what is computable with what is known – *in advance of executing any computational process* – to be calculable.[81] There are, of course, undecidable problems that are known in advance not to be calculable. But the reason that an arbitrary instance of freetext is not computable is because it cannot be known *in advance* whether or not an arbitrary instance of freetext is calculable (decidable). So, for the purpose of this work, *computable knowledge* is knowledge represented so that it is known to be calculable *before* it is used in a computational process. The practical implications of this definition are that any instance of *computable knowledge* must be able to be formatted so it becomes machine-interpretable. It follows that any instance of *computable knowledge* must ultimately be reducible to a set of instructions that a computing machine can execute which provide an algorithmically decided result (in contrast to a result that is undefined).

Towards perfect computable knowledge interoperability

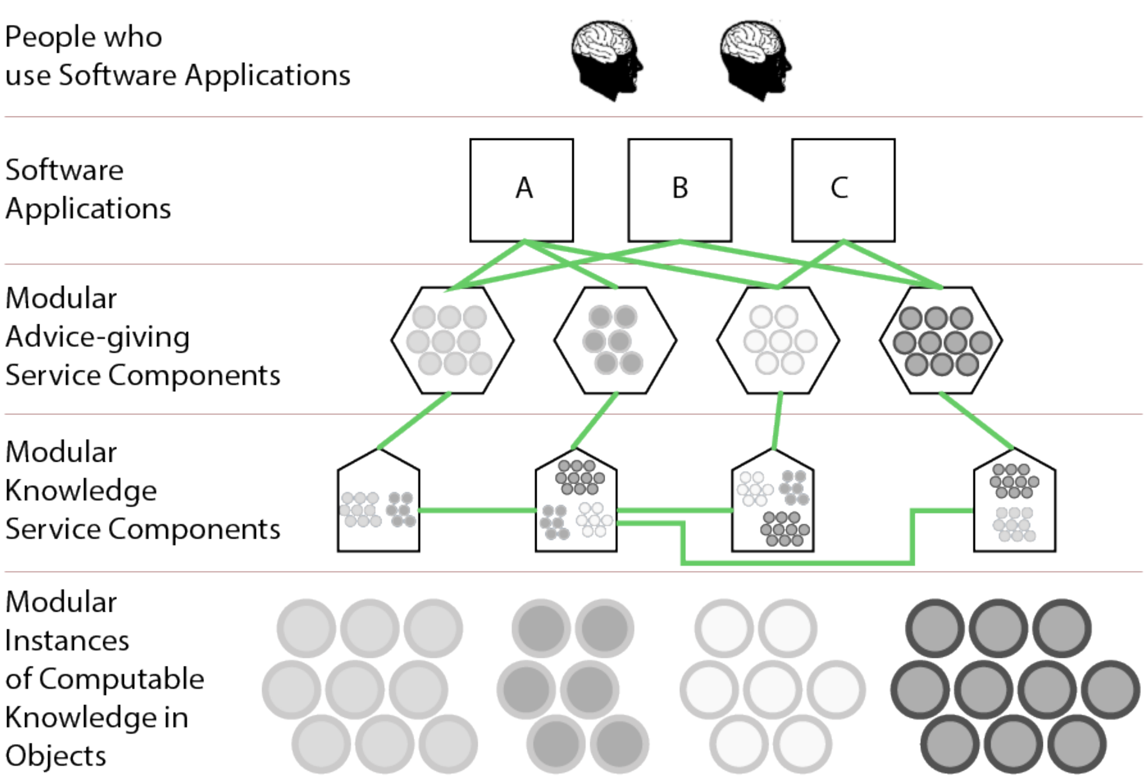
Next, we introduce a theoretically ideal state where computable knowledge, and the knowledge services and advice-giving services it enables, are perfectly interoperable. Perfect interoperability means that two disparate information systems can perform a desired function suitably without any further work either to make it or keep it so. In this case, the desired set of functions are to preserve computable knowledge, to distribute computable knowledge, and to use computable knowledge, in conjunction with information about the context of a decision, to generate advice.

Figure E below portrays the types of things that must interoperate perfectly to achieve an ideal state of computable knowledge preservation, distribution, and use to generate advice. We review these things in this theoretically perfect way just to set ourselves on a path to work *towards* perfect computable knowledge interoperability, even though we know it is impossible to achieve it.

Five layers are portrayed in Figure E below. Starting on the bottom layer we find modular instances of computable knowledge organized into four different collections of individual digital objects. In this portrayal, each circular digital object is a small system containing a discrete instance of computable knowledge. A discrete instance of computable knowledge could be a statistical prediction rule, a set of "IF-THEN" production rules, or some other single unit of computer-processable knowledge. Objects are formed into differently shaded collections.

On the next layer up, we find five-sided software modules. Each one of these is an individual system dedicated both to preserving computable knowledge in digital forms and to providing knowledge services which distribute computable knowledge. Each of the four modules in this layer serve as a repository and hold at least two of the collections of computable knowledge from the layer below. Having multiple copies of each collection is an important digital preservation mechanism that systems-of-systems like this enable.[82] Note that within this layer, green lines indicate that these four five-sided repository modules interconnect over a

Figure E. Ideal State of Perfect Computable Knowledge Interoperability



computer network. This is so that they can share the digital objects they hold with each other. These technical connections can be made in a variety of different ways.

Moving up another layer, we find four hexagonal modules that support four different and discrete advice-giving services. As indicated by the green lines, these advice-giving service modules are independent systems that connect over a network with the modules in the layer immediately below. Each of these advice-giving service modules has the capacity to combine data with computable knowledge to produce advice. In addition, the advice-giving service modules in this layer interact over a computer network with the software applications in the layer immediately above. Therefore, as depicted in Figure E, the advice-giving service modules on this layer are a kind of *middleware*, which is software that performs a bridging role on a computer network.[83] In this case, these four advice-giving service modules bridge the knowledge service system layer below to the application layer above.

On the next layer up, we find three software applications, generically labeled A, B, and C. These applications engage multiple advice-giving services to perform their roles. In Figure E, each application engages two or more advice-giving service modules in the layer immediately below. These applications use those advice-giving service modules to meet the needs of their human users, who are depicted in the top layer of Figure E.

There is reason to believe that, while not perfect because ongoing work would always be required, a high degree of computable knowledge interoperability could be achieved if the system-of-systems approach in Figure E could even partly be realized.[84] Such an approach, where discrete software modules are consistently used from the bottom up, can potentially give rise to a loosely coupled system-of-systems that can meet the actual demand for health advice.

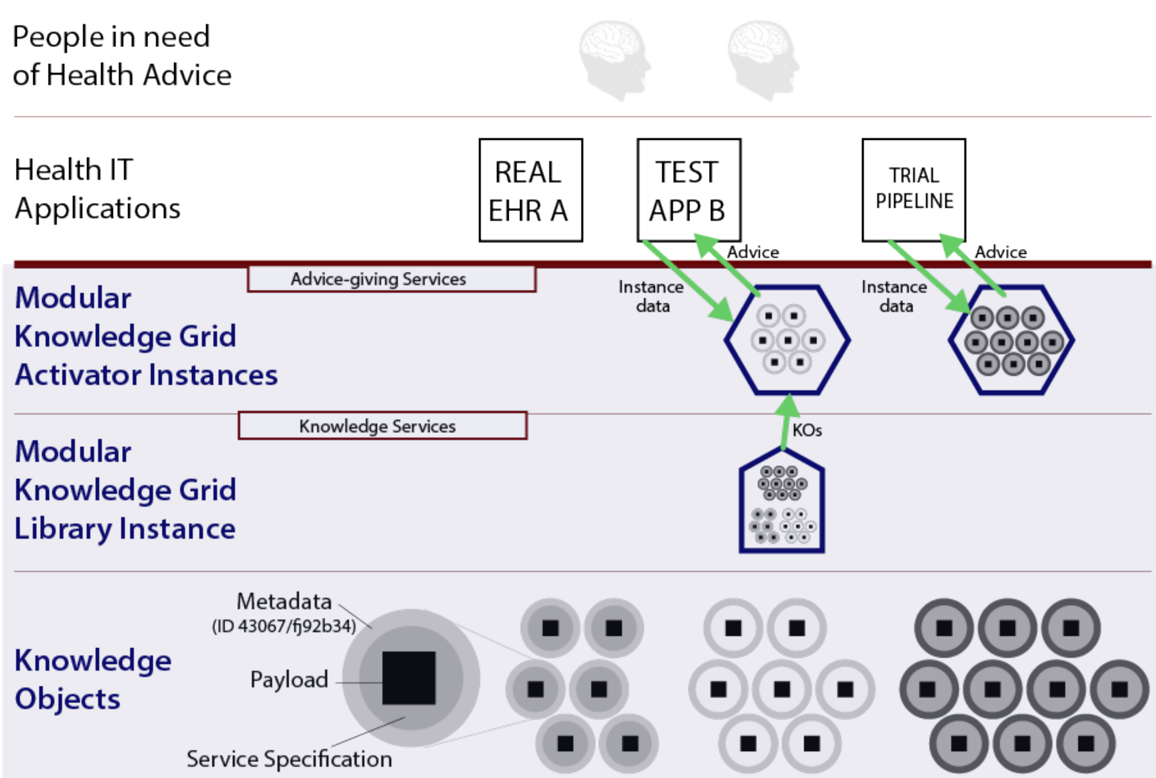
While it may not be obvious in Figure E, among the other things that it affords, the system-of-systems approach illustrated facilitates replacement of individual objects at the bottom, and knowledge service modules, and advice-giving service modules in the middle layers, without having to replace or change any other parts. A high degree of modularity like this is one of the known properties of infrastructure.[85]

The Knowledge Grid

The Knowledge Grid, which was first mentioned above in the Overview, is introduced here in brief and covered in depth in Chapter III. Figure F below portrays the Knowledge Grid in its current developmental form. The Knowledge Grid has not been deployed in practice. Therefore, no people in need of health advice are benefitting from it yet. Notice that its architecture generally follows the ideal system-of-systems architecture approach for increasing the interoperability of computable knowledge (see Figure E).

The Knowledge Grid is a novel, prototype, fledgling knowledge infrastructure (Figure F). It is intended to make computable knowledge as interoperable as possible. Components of the Knowledge Grid have first been built and then iteratively improved, starting in January of 2016, by a team of developers and researchers, including staff, students, and faculty, at the University of Michigan School of Information and Medical School.

Figure F. Technical Components of the Knowledge Grid



The Knowledge Grid is capable of supporting computable knowledge curation, organization, and distribution. It is also capable of supporting the application of computable knowledge to produce advice. Its three primary technical components exist as mature prototypes. These technical components are: (1) Knowledge Objects[86], (2) a conventional digital Library repository for curating, organizing, and distributing Knowledge Objects[87], and (3) an Activator for deploying Knowledge Objects in support of advice-giving services. These three technical components appear below the thick red horizontal line in the blue-shaded area of Figure F. Each component is briefly described next.

Knowledge Objects. As described conceptually by Flynn et al.[86], a "Knowledge Object", abbreviated "KO", is a structured *compound digital object*[77] suitable for holding and managing instances of computable knowledge. A KO is a specific type of compound digital object containing three types of information. As depicted in Figure F, every KO has: (1) descriptive information about the KO, i.e., its *metadata*, which always include a unique KO identifier, (2) a machine-interpretable representation of an instance of computable knowledge, called a *payload*, and (3) a representation of an interface or service specification, which describes the format and character of the input parameters needed to engage the computable knowledge held in the KO and the format and character of the advice that can be generated by engaging it.

The Knowledge Grid Library. A conventional library provides a set of core functions for storing, discovering, and curating its holdings.[87] A conventional digital library adequately supports people wanting to store, find, access, and curate digital artifacts. The system architecture for the Knowledge Grid Library, which is reviewed in Chapter III, was designed with these three functional categories in mind.[87] Currently, the Knowledge Grid Library component is a conventional digital library prototype that can be used to organize, curate, and distribute KOs. This component has its own API that enables it to provide knowledge services to other systems (Figure F). Automated knowledge services include KO discovery services, for finding KOs, and KO distribution services, for sharing KOs with other computer systems over a network. We have successfully tested a knowledge service that shares KOs with the Knowledge Grid Activator over a computer network (Figure F).

The Knowledge Grid Activator. The Knowledge Grid Activator is primarily an advice-giving service component. It uses Knowledge Objects to generate advice. The Knowledge Grid Activator is a modular system with its own API. Its API can expose the technical service specification of any KO in a way that external systems can process. By doing so, the Knowledge Grid Activator enables other systems to send it requests for advice, requests that carry instance data (Figure F). Upon receiving such requests, the Knowledge Grid Activator combines the instance data it receives over a computer network with the computable knowledge payload of the KO from which advice is sought to generate new messages of advice. The Knowledge Grid Activator then sends the advice it generates back to the requesting system (Figure F). Note that to date, instances of the Knowledge Grid Activator have been used primarily with test applications. However, in Chapter IV we report a trial involving the Knowledge Grid Activator as a key component in a new data-to-advice pipeline. We anticipate future work to use the Knowledge Grid to provide health advice to health IT applications used by people. This has not happened yet.

GOAL AND SPECIFIC AIMS

Goal

My goal in this dissertation is show how to increase the interoperability of computable biomedical knowledge sufficiently to potentiate the production of well-informed health advice.

Specific aims

The four specific aims of this dissertation are:

1. To define and categorize advice more precisely by building on existing advice theory found in the published record of the sciences of advice
2. To explain the ways in which proprietary knowledgebases limit the interoperability of the computable knowledge they hold and, by extension, limit the interoperability of the advice-giving systems they support

3. To determine how to increase the interoperability of the computable biomedical knowledge that is used by advice-giving systems through the design and development of new knowledge infrastructure components
4. To demonstrate that increasing the interoperability of computable biomedical knowledge enables modular, uniquely identifiable knowledge services and advice-giving services that are capable of generating useful health advice on demand and that can be updated automatically

This work unfolds from here in the following way. Each chapter addresses a single specific aim. Chapter I provides a scientific conceptual analysis of advice. It is followed by a trilogy of chapters about computable knowledge interoperability.

Chapter II shares a study of a promising but stand-alone advice-giving system for improving medication adherence called MedMinify. More to the point, Chapter II provides a retrospective analysis of the interoperability of the computable biomedical knowledge used by MedMinify. What results from this analysis is a better understanding of some of the things that limit the interoperability of computable knowledge in general.

Chapter III covers the design and development of the Knowledge Grid. It introduces the Knowledge Object Reference Ontology, KORO, which formally specifies what a Knowledge Object is. It also reviews the system architectures of the Knowledge Grid Library and the Knowledge Grid Activator.

The final chapter, Chapter IV, shares a study of a promising modularized data-to-advice pipeline called ScriptNumerate. ScriptNumerate makes use of the Knowledge Grid. It systematically analyzes raw medication prescription data, bundles the results of its analyses into Knowledge Objects, deploys the Knowledge Objects it generates using the Knowledge Grid Activator, and then uses the Knowledge Grid Activator API to generate prescription advice. By doing all of these things – automatically – ScriptNumerate points the way towards a future state when sufficient quantities of health advice can be generated using knowledge infrastructure built from the ground up to optimize the interoperability of computable knowledge.

References

1. Corrigan, J. in *Building a Better Delivery System: A New Engineering/Health Care Partnership* 95–98 (2005). doi:10.17226/11378
2. Whittington, J. W., Nolan, K., Lewis, N. & Torres, T. Pursuing the Triple Aim: The First 7 Years. *Milbank Q.* **93,** 263–300 (2015).
3. Higgins, S. T. Behavior change, health, and health disparities: An introduction. *Prev. Med. (Baltim).* 1–4 (2014). doi:10.1109/TMI.2012.2196707.Separate
4. Diem, G., Brownson, R. C., Grabauskas, V., Shatchkute, A. & Stachenko, S. Prevention and control of noncommunicable diseases through evidence-based public health: implementing the NCD 2020 action plan. *Glob. Health Promot.* **23,** 5–13 (2016).
5. Bouraga, S., Jureta, I., Faulkner, S. & Herssens, C. Knowledge-Based Recommendation Systems: A Survey. *Int. J. Intell. Inf. Technol.* **10,** 1–19 (2014).
6. Linzer, M. *et al.* The End of the 15–20 Minute Primary Care Visit. *J. Gen. Intern. Med.* **30,** 1584–1586 (2015).
7. Thompson, C. *et al.* The Effects of Time Pressure and Experience on Nures' Risk Assessment Decisions. *Nusring Res.* **57,** 302–311 (2008).
8. Bornmann, L. & Mutz, R. Growth Rates of Modern Science : A Bibliometric Analysis Based on the Number of Publications and. *J. Assoc. Inf. Sci. Technol. Technol.* **66,** 2215–2222 (2015).
9. Grossmann, C., Powers, B., Mcginnis, J. M. & Care, S. H. *Digital Infrastructure for the Learning Health System*. (2011). doi:10.17226/12912
10. Miller, B. M., Moore, D. E., Stead, W. W. & Balser, J. R. Beyond Flexner: a new model for continuous learning in the health professions. *Acad. Med.* **85,** 266–72 (2010).
11. Stead, W. W., Searle, J. R., Fessler, H. E., Smith, J. W. & Shortliffe, E. H. Biomedical informatics: changing what physicians need to know and how they learn. *Acad. Med.* **86,** 429–34 (2011).
12. Olsen, L., Aisner, D., McGinnis, J. M., Editors & Medicine, R. on E. *The learning healthcare system: workshop summary*. (2007).
13. Jureta, I. *Analysis and Design of Advice*. *Vasa* (Springer Berlin Heidelberg, 2011). doi:10.1007/978-3-642-18059-0
14. Edwards, P. N. *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*. (MIT Press, 2010).
15. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3,** 160018 (2016).
16. Fanciulli, A. & Wenning, G. K. Multiple-System Atrophy. *N. Engl. J. Med.* **372,** 249–263 (2015).
17. Flynn, A. J., Patton, J. & Platt, J. Tell it Like it Seems : Challenges Identifying Potential Requirements of a Learning Health System. *Proc. 48th Hawaii Int. Conf. Syst. Sci.* (2015). doi:10.1109/HICSS.2015.381
18. Etheredge, L. M. A rapid-learning health system. *Health Aff.* **26,** (2007).
19. Friedman, C. P., Rubin, J. C. & Sullivan, K. J. Toward an Information Infrastructure for Global Health Improvement. *IMIA Yearb. Med. Informatics* 1–8 (2017).
20. Friedman, C. *et al.* Toward a science of learning systems: a research agenda for the high-functioning Learning Health System. *J. Am. Med. Informatics Assoc.* 1–6 (2014).

doi:10.1136/amiajnl-2014-002977
21. Centers for Disease Control and Prevention. *Addressing prescription drug abuse in the United States:Current activities and future opportunities.* (2013).
22. Von Korff, M., Kolodny, A., Deyo, R. A. & Chou, R. Long-Term Opioid Therapy Reconsidered. *Ann. Intern. Med.* **160,** 567–570 (2014).
23. Centers for Disease Control and Prevention. *Annual Surveillance Report of Drug-Related Risks and Outcomes — United States, 2017. Surveillance Special Report 1*. (2017).
24. Kampman, K. & Jarvis, M. American Society of Addiction Medicine (ASAM) national practice guideline for the use of medications in the treatment of addiction involving opioid use. *J. Addict. Med.* **9,** 358–367 (2015).
25. Covell, D. G., Uman, G. C. & Manning, P. R. Information needs in office practice: are they being met? *Ann. Intern. Med.* **103,** 596–9 (1985).
26. González-González, A. Information needs and information-seeking behavior of primary care physicians. *Ann. Fam. ...* 345–352 (2007). doi:10.1370/afm.681.Jos
27. Smith, R. What clinical information do doctors need? *BMJ* **313,** 1062–8 (1996).
28. Clarke, M. A. *et al.* Information needs and information-seeking behaviour analysis of primary care physicians and nurses: a literature review. *Health Info. Libr. J.* **30,** 178–90 (2013).
29. Beasley, J. W. *et al.* Information Chaos in Primary Care: Implications for Physician Performance and Patient Safety. *J Am Board Fam Med* **24,** 745–751 (2011).
30. Markman, M. Information overload in oncology practice and its potential negative impact on the delivery of optimal patient care. *Curr. Oncol. Rep.* **13,** 249–51 (2011).
31. Sunstein, C. R. Probability neglect: Emotions, worst cases, and law. **112,** 61–107 (2002).
32. Berry, D. Interpreting information about medication side effects: differences in risk perception and intention to comply when medicines are prescribed for adults or young children. *Psychol. Health Med.* **9,** 227–234 (2004).
33. Poses, R. M. & Anthony, M. Availability, Wishful Thinking, and Physicians' Diagnostic Judgments for Patients with Suspected Bacteremia. *Med. Decis. Mak.* **11,** 159–168 (1991).
34. Reyna, V. F. A theory of medical decision making and health: fuzzy trace theory. *Med. Decis. Making* **28,** 850–65 (2008).
35. Gruppen, L., Margolin, J., Wisdom, K. & Grum, C. Outcome bias and cognitive dissonance in evaluating treatment decisions. *Acad. Med.* **69,** S57–S59 (1994).
36. Leape, L. L., Lawthers, A. G., Brennan, T. A. & Johnson, W. G. Preventing medical injury. *QRB. Qual. Rev. ...* **19,** 144–149 (1993).
37. Expert Panel Convened by The National Patient Safety Foundation, Berwick, D. & Shojania, K. G. *Free from Harm*. (2015).
38. Lesar, T. S., Briceland, L. & Stein, D. aniel S. Factors related to errors in medication prescribing. *JAMA* **277,** 312–317 (1997).
39. Kohn, L. T., Corrigan, J. & Donaldson, M. S. *To err is human : building a safer health system*. (National Academy Press, 2000).
40. Kelley, T. Poor Coordination Among Care Providers Contributed to Child Deaths, Report Says. *New York Times* (2005).
41. McGlynn, E. A. *et al.* Quality of health care delivered to adults in the United States. *N Engl J Med* **348,** 2635–2645 (2003).
42. Smith, M., Saunders, R., Stuckhardt, L. & Mcginnis, J. M. in *National Academy Press* (2012).

43. Pauker, S. G., Gorry, G. A., Kassirer, J. P. & Schwartz, W. B. Towards the simulation of clinical cognition. Taking a present illness by computer. *Am. J. Med.* **60,** 981–96 (1976).
44. National Library of Medicine. *Detailed Indexing Statistics 1965-2016, U.S. National Institutes of Health.* (2017).
45. Jiang, G., Solbrig, H. R. & Chute, C. G. ADEpedia: a scalable and standardized knowledge base of Adverse Drug Events using semantic web technology. *AMIA Annu. Symp. Proc.* **2011,** 607–16 (2011).
46. De Leon, J. Highlights of drug package inserts and the website dailymed: The need for further improvement in package inserts to help busy prescribers. *J. Clin. Psychopharmacol.* **31,** 263–265 (2011).
47. Relling, M. & Klein, T. CPIC: Clinical Pharmacogenetics Implementation Consortium of the Pharmacogenomics Research Network. *Nature* **89,** 464–467 (2011).
48. Giuse, N. B. *et al.* Information needs of health care professionals in an AIDS outpatient clinic as determined by chart review. *J. Am. Med. Inform. Assoc.* **1,** 395–403 (1994).
49. Weed, L. L. & Hertzberg, R. The use and construction of problem-knowledge couplers, the knowledge coupler editor, knowledge networks, and the problem-oriented medical record for the microcomputer. ... *Comput. Appl. Med.* ... 831–836 (1983).
50. Salacuse, J. W. *The Wise Advisor - What every professional should know about consulting and counseling*. (Praeger, 2000).
51. Friedman, C. P. A 'fundamental theorem' of biomedical informatics. *J. Am. Med. Inform. Assoc.* **16,** 169–70 (2009).
52. Shannon, C. A mathematical theory of communication. *Bell Syst. Tech. J.* **27,** 379–423 (1948).
53. Jaspers, M. W., Smeulers, M., Vermeulen, H. & Peute, L. W. Effects of clinical decision-support systems on practitioner performance and patient outcomes: a synthesis of high-quality systematic review findings. *J Am Med Inf. Assoc* **18,** 327–334 (2011).
54. Raths, D. What Is CDS Hooks ? An Interview with Josh Mandel About FHIR and Clinical Decision Support. *Healthcare Informatics* (2016).
55. Payne, H. E., Lister, C., West, J. H. & Bernhardt, J. M. Behavioral Functionality of Mobile Apps in Health Interventions: A Systematic Review of the Literature. *JMIR mHealth uHealth* **3,** e20 (2015).
56. Blumenthal, D. & Tavenner, M. The 'Meaningful Use' Regulation for Electronic Health Records. *N. Engl. J. Med.* **363,** 501–504 (2010).
57. Sittig, D. F. *et al.* Grand challenges in clinical decision support. *J. Biomed. Inform.* **41,** 387–92 (2008).
58. Greenes, R. A., Peleg, M., Rector, A. & Osheroff, J. Reusable knowledge for best clinical practices: why we have difficulty sharing and what we can do. *Stud Heal. Technol Inf.* **192,** 1237 (2013).
59. Greenes, R. A. *et al.* Clinical Decision Support Models and Frameworks: Seeking to Address Research Issues Underlying Implementation Successes and Failures. *J. Biomed. Inform.* (2017). doi:10.1016/j.jbi.2017.12.005
60. Evans, R. S. *et al.* A computer-assisted management program for antibiotics and other antiinfective agents. *N Engl J Med* **338,** 232–238 (1998).
61. Finlay, G. D., Rothman, M. J. & Smith, R. A. Measuring the modified early warning score and the Rothman Index: Advantages of utilizing the electronic medical record in an early warning system. *J. Hosp. Med.* **9,** 116–119 (2014).

62. Sheng, A. Y., Castro, A. & Lewiss, R. E. Awareness, Utilization, and Education of the ACR Appropriateness Criteria: A Review and Future Directions. *J. Am. Coll. Radiol.* **13,** 131–136 (2016).
63. Jafarpour, B. & Abidi, S. S. R. Merging Disease-Specific Clinical Guidelines to Handle Comorbidities in a Clinical Decision Support Setting. in *Artificial Intelligence in Medicine* (eds. Peek, N., Marín Morales, R. & Peleg, M.) 28–32 (Springer Berlin Heidelberg, 2013).
64. Tsafnat, G. & Coiera, E. W. Computational reasoning across multiple models. *J Am Med Inf. Assoc* **16,** 768–774 (2009).
65. Ash, J. S., Sittig, D. F., Campbell, E. M., Guappone, K. P. & Dykstra, R. H. Some unintended consequences of clinical decision support systems. *AMIA Annu Symp Proc* 26–30 (2007).
66. Campbell, E. M., Sittig, D. F., Ash, J. S., Guappone, K. P. & Dykstra, R. H. Types of unintended consequences related to computerized provider order entry. *J Am Med Inf. Assoc* **13,** 547–556 (2006).
67. Geissbuhler, a & Miller, R. a. Distributing knowledge maintenance for clinical decision-support systems: the 'knowledge library' model. *AMIA Annu Symp Proc* 770–4 (1999).
68. Walker, J. *et al.* The value of health care information exchange and interoperability. *Health Aff.* **Suppl Web,** 10–18 (2005).
69. Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S. & Ramoni, R. B. SMART on FHIR: A standards-based, interoperable apps platform for electronic health records. *J. Am. Med. Informatics Assoc.* **23,** 899–908 (2016).
70. Paulus, R. A., Davis, K. & Steele, G. D. Continuous innovation in health care: Implications of the Geisinger experience. *Health Aff.* **27,** 1235–1245 (2008).
71. Paterno, M. D. *et al.* Tiering drug-drug interaction alerts by severity increases compliance rates. *J Am Med Inf. Assoc* **16,** 40–46 (2009).
72. Chen, D., Doumeingts, G. & Vernadat, F. Architectures for enterprise integration and interoperability: Past, present and future. *Comput. Ind.* **59,** 647–659 (2008).
73. Cerf, V. G. & Kahn, R. E. A Protocol for Packet Network Intercommunication. *IEEE Trans. Commun.* **22,** (1974).
74. Lagoze, C. & Van de Sompel, H. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. *Proc. first ACM/IEEE-CS Jt. Conf. Digit. Libr. - JCDL '01* 54–62 (2001). doi:10.1145/379437.379449
75. De Sompel, H. Van & Nelson, M. L. Reminiscing about 15 years of interoperability efforts. *D-Lib Mag.* **21,** 1–1 (2015).
76. Erickson, J., Van de Sompel, H., Lagoze, C., Payette, S. & Warner, S. Rethinking Scholarly Communication. *D-Lib Mag.* **10,** 1–12 (2004).
77. Kahn, R. & Wilensky, R. A framework for distributed digital object services. *Int. J. Digit. Libr.* **6,** 115–123 (2006).
78. Paepke, A., Chang, C.-C. K., García-Molina, H. & Winograd, T. Interoperability for digital libraries workdwide. *Commun. ACM* **41,** 1–23 (1998).
79. Perlin, J. B., Baker, D. B., Brailer, D. J. & et al. Information Technology Interoperability and Use for Better Care and Evidence A Vital Direction for Health and Health Care. *Vital Dir.* (2016).
80. Padgham, D., Edmunds, M. & Holve, E. Interoperability in the United States Health System Health Information Interoperability in the United States Health System. (2016).
81. Soare, R. I. *Turing and the Art of Classical Computability*. (2012).

82. Maniatis, P., Roussopoulos, M., Giuli, T. J., Rosenthal, D. S. H. & Baker, M. The LOCKSS peer-to-peer digital preservation system. *ACM Trans. Comput. Syst.* **23,** 2–50 (2005).
83. Al-Jaroodi, J. & Mohamed, N. Middleware is STILL Everywhere!!! *Concurr. Comput. Pract. Exp.* **24,** 1919–1926 (2012).
84. Morris, E., Levine, L., Meyers, C., Place, P. & Plakosh, D. System of Systems Interoperability (SOSI): final report (No. CMU/SEI-2004-TR-004). *Carnegie-Mellon Univ Pittsburgh Pa Softw. Eng. Inst.* (2004).
85. Star, S. L. & Ruhleder, K. Steps Toward Design an Ecology and Access of Infrastructure : for Large Spaces Information. *Inf. Syst. Res.* **7,** 111–134 (1996).
86. Flynn, A., Shi, W., Fischer, R. & Friedman, C. P. Digital Knowledge Objects and Digital Knowledge Object Clusters: Unit Holdings in a Learning Health System Knowledge Repository. *Hawaii Int. Conf. Syst. Sci. 2016* (2016).
87. Flynn, A. J. *et al.* Architecture and Initial Development of a Digital Library Platform for Computable Knowledge Objects for Health. *Stud. Health Technol. Inform.* **235,** 496–500 (2017).

CHAPTER I

Theory of Advice as an Information Object Targeted at an Unmade Decision

> We must seek better ways to integrate our growing body of scientific knowledge and technical expertise with our needs as human beings living in an increasingly global society.
> – George E. Brown, Jr.[1]

INTRODUCTION

The purpose of this chapter is to bring about a better understanding of advice, and its relationship to documents, information, and knowledge, by synthesizing and extending existing advice theory. This chapter sets the stage for the rest of this dissertation, where increasing the interoperability of computable knowledge, and thereby the interoperability of related knowledge services and advice-giving services, is the focus. Several questions about advice are addressed. What is *advice* and what are *advice documents*? What is its purpose? What types of *advice* exist?

Since at least the 1940s, researchers have studied *advice*[2]. Yet what *advice* is remains unclear. Our vague understanding of *advice* is reflected in the Oxford English Dictionary. In current English usage, the word *advice* connotes variously as "an opinion as to what action to take", "a recommendation", or as, "information imparted"[3]. *Advice* is thought of concurrently in these three different ways, which respectively increase in their generalness, finally encompassing any imparted information! This chapter surfaces and then extends available scientific theory about *advice* to conceptualize it more clearly. There are good reasons to do this.

An argument for the importance of studying advice follows in three succinct parts. It is predicated on the general usefulness of advice, its costs, and on overt unmet needs for advice.

Advice is evidently useful in addressing information overload. Weinberger[4] claims that information overload causes us to fear that we are, "not getting enough of the information we need." Indeed, the needs of decision-makers to reduce information load to manageable levels are well-founded[5]. Sometimes the information people need can be obtained as advice[6,7]. This helps explain the continued growth of professional advice-giving services[8]. Further explanation comes from Bryson[9]. He views advice as a general mechanism, "to put knowledge at the service of power." Power is sometimes exercised on a grand scale. Anthropologists Collier and Ong[10] describe how complex, enduring "global assemblages" of production depend on "fine-grained coordination" arising from flows of expertise and standards, which constitute advice. Reinforcing this point, Jureta[11] claims advice is a tool that begets coordination.

Advice is important for economic reasons. It often costs dearly, demonstrating its perceived utility through pricing. Competent legal advice is currently estimated to cost $200 per hour[12]. Financial planning advice can cost $1000 or more per financial plan[13]. Likewise, the advice of medical experts, available as "second opinions", is sold at similar prices[14]. Evidence shows that irrevocable pre-payment for advice biases purchasers towards its use[15]. Of course, the total cost of advice includes the time and effort it takes to get it[16].

Evidence that there is a serious shortage of advice comes from law and medicine. In law, people frequently represent their own interests without the benefit of legal advice, diminishing their chances of an ideal legal outcome[12]. In medicine, the unmet need for advice is apparent in multiple ways. These include growing knowledge gaps in medical practice[17,18], the high frequency of avoidable medical errors[19,20], and a lack of care coordination[21].

To date, researchers have approached the study of advice from multiple positions. Their work has resulted in several distinct lines of inquiry. Three such lines are explored here with an emphasis on *advice theory*[22]. One line focuses on *human-to-human advice interactions*, another focuses on *machine-to-machine advice interactions*, and a third focuses on *advice interactions involving humans that are intermediated by information and communication technologies* (ICTs). These three lines of inquiry are briefly introduced next and later scrutinized.

By far the broadest line of scientific inquiry about advice examines interactions between human advisors and human advisees. This line runs primarily through the communication, counseling, decision-making, economic, educational, health services, management, organizational, political, and psychological sciences. Bryson's "Notes on a Theory of Advice"[9] is an early example. In it he relates characteristics that differentiate the roles, attitudes, and actions of human advisors from those of human advisees. Bonaccio and Dalal[23] and MacGeorge, Feng, and Guntzviller[22] each provide comprehensive reviews about human-to-human advice.

In contrast, perhaps the most narrowly focused line of inquiry about advice is found in computer science. This line concerns artificial intelligence, computing theory, and distributed computing. It focuses on advice that is exchanged by machines in support of machine reasoning. As an early example, McCarthy published a theory of machine decision-making[24]. In it, advice takes the form of computable facts about situations, goals, and actions. McCarthy's factual advice serves as input that, when combined with existing computable means-ends axioms, i.e., with computable knowledge, enables a machine to reason and make decisions automatically.

A third diffuse and nearly independent line of inquiry investigates advice when it is designed and arranged by human advisors for delivery to human advisees through intermediating ICTs. This line of inquiry focuses on human-in-the-loop automation and decision support systems. It runs mostly through the sciences of automation, cognitive ergonomics, computing, decision-support, design, human-computer interaction, human factors, informatics, information, knowledge engineering, safety, trust, and visualization. An early example of ICT-intermediated advice is the MYCIN interactive computer program. MYCIN generated advice about antibiotic treatments[25]. More contemporary examples of decision support systems abound[26].

Prior research on advice highlights the following open questions. What types of advice exist? What explains why these types exist? How is each type of advice formulated and used? In terms of its content, is all advice normative or is some purely factual? Regarding its purpose, is advice primarily intended to increase the number of alternatives about which decision-makers are aware, or is it to help them decrease the number of alternatives under consideration? Regarding its intended effects, is the force of advice limited to being suggestive or can it also include persuasive communication, like an argument? The answers to these questions about advice are far from self-evident.

The goal of the conceptual analysis presented next is to add clarity about advice, its types, and its purpose. This chapter primarily examines how researchers have approached advice through the lens of the information and documentation sciences.

BACKGROUND

Foundational concepts about advice

The conceptual foundation for this chapter is established next, starting with the concept of advice itself. In general, advice is understood as a category of material information objects having a verbal or written form. A material information object is one that can be sensed or perceived. This perspective aligns with Searle's[27] fundamental notion of a speech (or linguistic) act being a rule-governed "noise or mark" produced by a speaker or writer who has certain intentions in mind. Moreover, this view of *advice-as-information-object* permits advice to be analyzed using the theories and methods of the information and documentation sciences.

Since the 1970s, Buckland has developed a research program relevant to this work. One way Buckland[28] and his predecessors have conceptualized information is as a material thing. For clarity, a material embodiment of information, in the form of verbal utterances or written symbols that can be sensed or perceived in the world, is referred to hereon either as an *information object* or *message*.

When advice is conceptualized as a proper subset of information objects, it means that some information objects are advice while other information objects are not advice. The challenge taken up here is to articulate the property or properties that distinguish advice objects from all other information objects. Properties exclusive to advice could be intrinsic ones, such as an advice-specific grammar, or extrinsic ones, such as its unique purpose [29].

Another foundational concept for this work is the concept of an *advice interaction*. An *advice interaction* is an instance of advisors and advisees respectively giving and receiving advice. *Advice interactions* are frequently, but not always, social phenomena. As Foucault noted[30], there are times when the self becomes its own subject. At such times, a person may self-advise via an internal mental dialog or by self-verbalization.

Also, as noted above, one machine can be made to advise another machine. Yet machine-to-machine advice interactions, while they are obviously "artificial" and not social phenomena, nevertheless ultimately arise out of human needs[5]. This chapter covers advice interactions that involve people, machines, or people using intermediating ICTs. Given these three different *advice interaction modes*, and considering the different roles of advisors and advisees, it is clear that advice must be studied from a variety of perspectives to understand it fully.

Placing advice in the context of decision situations

All of the previously cited advice scholars, and many others, explicitly place advice interactions in the larger context of *decision situations*[9,11,22–24,31]. Some of these scholars have outlined decision-making processes involving advice in detail[9,11].

As the term is understood here, a *decision situation* is any situation when two or more alternatives are under consideration in advance of a decision being made[32]. A *decision*, which is also called a choice, is said to be made (or taken or finalized or reached) when one alternative is chosen while other alternatives are not[33]. Given this definition, decisions may incur opportunity costs resulting from unchosen alternatives. These opportunity costs can be considerable[34]. Hence, *decision situations* encapsulate decision-making processes involving the weighing of options when tradeoffs are inevitable.

Within a decision situation, decision-making processes may begin with problem representation and goal-setting[5,9,22]. These preliminary steps support deliberation over what decision to take and why to take it. Simon's[5] strong argument that human rationality is bounded is important to mention here. Simon points out real limits, in time and expertise, that govern human decision-making processes. Bolstering Simon's argument for bounded rationality is work that has uncovered consistent cognitive biases in human decision-making[35].

In part as a response to these discoveries about human cognition, starting in the 1980s, a body of scientific work began to grow in the area of naturalistic decision making[33,36]. It has since been shown that human decision-making unfolds imperfectly via various formal and informal methods of intuition and analysis. It involves prior experience, mental simulation, and decision situation categorization[33]. During such decision-making processes, advice interactions take place whenever advice is sought and shared.

Not all decision situations ultimately result in a decision. Sometimes decisions are set-aside and forever left unmade. As Lewis[37] points out, not all decision situations need to be resolved because not all decisions finally matter. Things change, sometimes quickly, and when they do a decision situation may be exited without reaching any decision.

Decision situations, decision accuracy, uncertainty and advice

Another key concept related to advice and decision-making is the concept of *decision accuracy*. Many studies have explored the effects of advice on decision accuracy[23]. Taylor[38] operationalized decision accuracy as a measure of the correctness of a series of decisions as judged by an expert panel. Many studies since have operationalized decision accuracy similarly by comparing decisions made to known facts that serve as a "gold standard." LaFond[39] provides a recent example by defining decision accuracy as agreement between the actual choices made by a decision-maker and what is known to be the truth. In a contrasting move, Friedman and colleagues innovated a method to measure how close physician subjects came to giving an accurate diagnosis in response to complex medical cases[40]. Their physician subjects created ranked lists of possible diagnoses. This enabled the study team to use a definitive diagnosis for each case to calculate both decision accuracy and each subject's degree of "correctness."

Next, a host of key concepts about uncertainty and decision situations are shared to illuminate the limits of using decision accuracy as a measure in advice studies.

Two key concepts are *subjective uncertainty* and *environmental uncertainty*[41]. *Subjective uncertainty* is the decision maker's own subjective view of the degree of predictability in a decision situation. *Environmental uncertainty* is the degree to which a future state of nature can be accurately predicted with information and knowledge in the world. Environmental uncertainty is thus a combined measure of what is as yet unknown and what is forever unknowable. It has been noted by Simon[5] and others, that, due to environmental uncertainty, decision accuracy cannot be determined for many, if not most, real-world decisions[42].

Two subcategories of environmental uncertainty are *statistical uncertainty*, which refers to inherent randomness, and *epistemic uncertainty*, which reflects a lack of knowledge in the world[43]. Many decisions are impacted by irresolvable *statistical uncertainty* or potentially resolvable, but as yet unresolved, *epistemic uncertainty*[43]. For decision situations associated with these uncertainties, the best possible advice may be limited to helping decision-makers recognize the types and degrees of uncertainty that pertain to an unmade decision[44].

Another key concept from computability theory is called *decidability*. Here it is sufficient to point out that some decision problems are known to be undecidable[45], while others are thought to be so[46]. For undecidable decision problems, it is impossible to construct a method that always leads to an answer. These findings establish that irresolvable and seemingly irresolvable environmental uncertainties exist. As a consequence, following Simon[5], the additional concept of the *defensibility* of a decision becomes important to the science of advice studies.

Compared to decision accuracy, *decision defensibility* is a weaker standard by which to judge decisions. Many decision problems do not have a single "correct answer." In such cases decisions may be fairly judged defensible or indefensible[5]. Standards akin to decision defensibility appear in the law, such as the *reasonable person* standard for judging liability[47]. Unlike decision accuracy, which is a binary, "yes/no" measure, *decision defensibility* is measured in degrees. As the environmental uncertainty pertaining to a decision situation goes up, the evidence available to defend the decision goes down.

Risk is another dimension influencing evaluations of decision defensibility[48]. Some decisions involve much greater risks than others, potentially making them more difficult to defend in the case of bad outcomes[49].

Taken together, the concepts of decision defensibility and decidability raise the following important question: *When is it reasonable to judge the accuracy of a decision and when is it not?* Many decisions are judgment calls made in the context of decision situations associated with risks, opportunity costs, potential gains, and environmental uncertainties[42,46,47]. These decisions may be judged on a scale of defensibility, but their accuracy is necessarily indeterminable. This chapter frames the concept of *advice* in this light.

What makes advice acceptable to human advisees?

A sizeable body of research has been conducted to surface features that tend to make advice messages acceptable to human advisees. While *advice acceptability* is not the main focus of this work, the results of these studies provide useful background for conceptualizing advice.

Linguistic analyses of actual advice communicated during advice interactions indicate that it is problematic to give advice as a command[50,51]. Instead, advisors tend to employ modality markers (e.g., *you could do this*), pseudo-cleft constructs (e.g., *what I would do is this*), agent de-emphasis (e.g., *that thing needs this*), and conditionals (e.g., *if this is the case then you can do that*). These linguistic devices soften both imperative commands (e.g., *do this!*) and outwardly normative statements (e.g., *you should do this*), thereby improving advice acceptability[52].

Several scholars have noted the need for advisors to analyze the decision problem with advisees prior to offering advice. Leppanen[50] concluded that “advice is easier to communicate when problems are established prior to the delivery of advice.” Similarly, Jefferson and Lee[53] distinguished between “troubles telling” conversations, which do not focus on a particular problem, and “service encounter” conversations, where a problem is the main focus of discussion. They noted that advice is more likely to be accepted during “service encounter” conversations. These results are further bolstered by indications that unsolicited advice is more likely to be rejected[50,53–55].

Advice acceptability is related to the source of advice. It has been shown that relational closeness between advisor and advisee[56], the offering of emotional support prior to advice-giving[57], and advisor expertise, are strongly associated with receptiveness to advice[58]. To wit, it is known that trust between advisors and advisees plays a significant role in advice interactions[59,60].

Finally, advice acceptability is related to the benefits and costs implied by the advice. Advice is perceived to be of higher quality if it is "feasible" to implement, "absent of limitations" that result in losses for the human advisee, and not too radical[57,61]. However, it has also been shown that human advisees generally find advice with unexpected content to be of greater value than advice they anticipate[62,63].

Advice and the concept of "documental becoming"

We now examine ways in which document science may apply to advice. Following Meyriat[64], Lund[65], and Couzinet[66], documents are understood to have both a "dormant" informational-material dimension and an "active" communicational dimension. The former "dormant" dimension is subject to curation. The latter "active" dimension is perceptual [29].

By drawing on these antecedents from document theory, Gorichanaz and Latham[29] used document phenomenology to create a framework of "*documental becoming*" that distinguishes between *information as a material thing* and *information as meaning*. Their framework builds on the idea that information can be understood differently at multiple levels. As Weaver points out, information can be understood at one level in terms of transference of material bits of information but also at another level in terms of conveyance of meaning [67]. In this regard, Weaver's levels of information resemble Austin's similar concepts of locutionary, illocutionary, and perlocutionary impacts of speech[68].

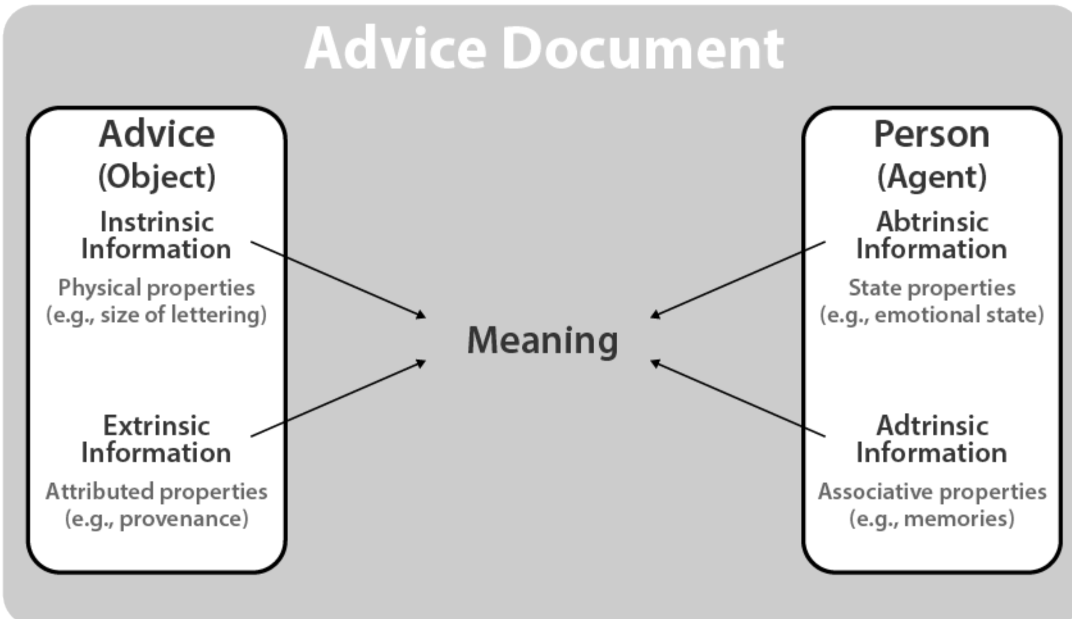
A related but more general perspective is found in the Basic Formal Ontology where *continuants*, which are material things that persist in the world, are contrasted with *occurrent*s, which are fleeting processes that take place in the world[69]. In this regard, it may be more precise to say that *documental becoming* happens when a continuant information object is given meaning through an occurrent process, i.e., a transaction. If what results from an occurrent process of documental becoming is recorded, then it becomes yet another continuant information object. Thus, the potential for a repeating cycle of sequential object-making and meaning-making is implied via transactions of *documental becoming*.

As portrayed in more detail in Figure 1.1, the constructivist documental becoming framework of Gorichanaz and Latham[29] brings together a material information object and a meaning-generating person into a process resulting in a *document*. In their conceptualization, a *document* results from a transaction whereby four types of information are processed and "made into meaning – and action."[29]. The four types of information processed during a meaning-making transaction resulting in a *document* are intrinsic and extrinsic information associated with a material information object and "abtrinsic" and "adtrinsic" information associated with a person who encounters and ascribes meaning to the information object. Abtrinsic information is information about that person's mental state, e.g., an attitude. Adtrinsic information is information about that person's past life experience, e.g., a memory.

Gorichanaz and Latham[29] focus only on "object-person" documents but they do not rule out the existence of "object-machine" documents, where a machine encounters an information object and executes a transaction that also gives rise to situated and momentary meaning – and action, and thus a constructed *document*. Comparing a machine to a person, abtrinsic information may be analogous to the current *machine state* whereas adtrinsic information may be analogous to a *machine-accessible log of past computational events*.

Using the framework of documental becoming[29], *advice* can be distinguished from *advice documents* (Figure 1.1). Advice is an information object or message. *Advice documents* are what result from transactions whereby active processes of reasoning ascribe meaning to *advice* at a specific moment in time. Following Gorichanaz[70], the constructivist perspective about how

Figure 1.1 Phenomenological Structure of 'Advice document' Becoming



documents exist views all *advice documents* as idiosyncratic, context-bound, and unique to the individual *advice beholder* at the moment of advice beholding. An advice beholder is an agent that reasons. In sum, *advice documents* constitute situated meaning that arises when an agent encounters *advice*. As an example, consider a college student acting as an agent who seeks advice about which course to take from his or her academic advisor. Upon receiving a verbal message of *advice* suggesting a particular course, the student ascribes meaning to the *advice* in his or her own mind, thereby giving rise to a situated and momentary *advice document*.

In this chapter, it is necessary to account for object-machine *documents* because advice is known to be shared among machines[24,71,72]. Yet it is certainly controversial to view information processing by a machine as a meaning-making transaction that results in a newly constructed *document*[73]. Here this controversy is mostly sidestepped by appealing, on the one hand, to known capabilities of machine reasoning, which have given rise to "systems of interactive agency" [74]; and, on the other hand, by stipulating that some experiences remain exclusively human (e.g., feelings and emotions)[75]. In keeping with Actor Network Theory, which addresses powers and their effects regardless of their source [76], the term *agent* is used here to signify a participant with influence in an actor network. Any *agent* with reasoning capabilities, be it a person or machine, can act as an advisor, by formulating and sending advice, and also as an advisee, by receiving and giving meaning to advice (Figure 1.1). Furthermore, like human agents, machine agents can also learn[77].

RESEARCH QUESTIONS

The above review provided background and a broad conceptual foundation for this study. It asserted and justified the following initial conjectures. *Advice* is some type of material, transmittable information object. It is given and received during *advice interactions*, which happen within the context of *decision situations*. Decision situations are associated with decisions that involve varying degrees of *environmental uncertainty*. Once made, some decisions can be judged for accuracy. Many other decisions can only be judged on their defensibility. When advice is given by an advisor to a human advisee, what makes it acceptable are the advisor's discourse, advisor and advisee understandings of the decision problem, the nature of the advisor-advisee relationship, the advisor's expertise, the cost paid for the advice, and the perceived feasibility, benefit, and costs of adopting it. Moreover, during an advice interaction, *documental becoming* occurs in the following way. Advice is encountered by an agent and processed via a meaning-making transaction resulting in a momentary, situated *advice document*. To examine the issues raised above in more detail, this study addresses three research questions:

RQ1. In which ways have researchers approached the analysis of advice as a concept?

RQ2. What types of advice emerge by analyzing advice-as-an-information-object using a framework that spans decision accuracy, decision defensibility, and decision decidability to encompass a content-based typology of advice?

RQ3. How may advice be conceptualized in light of existing advice theory while also accounting for a content-based typology of advice?

Answers to *RQ1* arise from a systematic literature review focused on *advice theory*. This literature review covers three lines of inquiry about human-to-human, machine-to-machine, and ICT-intermediated advice interactions. An answer to *RQ2*, in the form of a new framework and related *content-based typology of advice*, comes about in light of the answers to *RQ1*. Similarly, an answer to *RQ3*, in the form of a refined *theory of advice*, comes about by applying insights from addressing *RQ1 and RQ2* to reconcile perspectives and clarify the nature of advice.

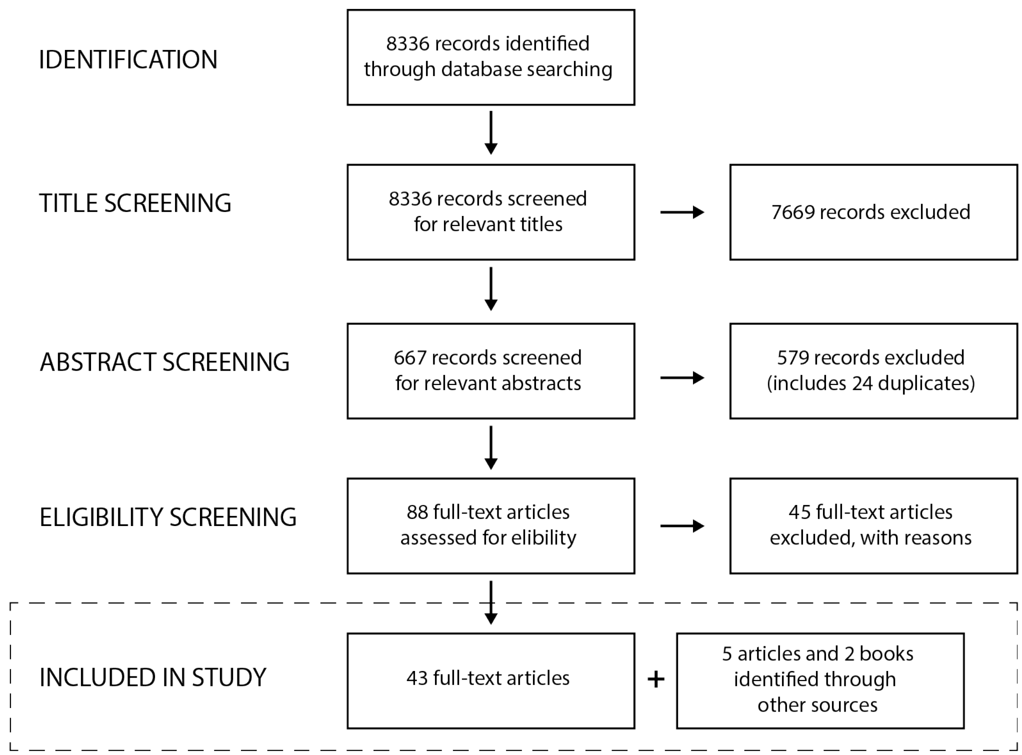
METHODS, RESEARCH MATERIALS, AND ANALYSIS

To answer the research questions above, a systematic literature review of scientific publications about *advice theory* was undertaken. The review identified prior works that explicitly conceptualized *advice* or *advice interactions* in a theoretical manner, or which described models (or types) of advice or advice interactions as a proxy for theory about advice.

By design, the scope of the review was further expanded in an attempt to identify how the term *advice* is generally used within scientific publications found in four domain-oriented databases. This widening of scope was effected in part by the deliberate inclusion of relevant systematic reviews found in the *advice studies* literature. Two scholarly books about advice were identified and included as well[11,54]. The final sample of research materials was further augmented to ensure coverage of several key concepts related to advice, e.g., *construal level*, *coordination*, *decision accuracy*, and *decision defensibility*.

The publication databases searched are listed next with their domains of scientific coverage in parentheses. They are ProQuest's ABI/INFORM Collection (covering business, organization, and management sciences), the Association for Computing Machinery's Digital Library (covering computer and information sciences), the U.S. National Library of Medicine's Pubmed (covering biomedical sciences), and EBSCO's PsychINFO (covering behavioral and social sciences). As a backstop, the general-purpose database Google Scholar was also searched specifically for works about *advice theory* to identify any obviously relevant works that fell outside the scope of the other four domain-oriented databases.

Figure 1.2 Information Flow for Systematic Literature Review



To give structure to the systematic literature review, the PRISMA 2009 Checklist, pertaining to the documentation of a systematic review, was used[78].

A flow diagram for the literature search is shown in Figure 1.2. The keywords used to search publication titles were term expansions of the word root advi* (e.g., advice, advisor, advisee, advising), one of which was mandatory for inclusion in the initial search results, along with 24 additional terms generally related to scientific works of a theoretical nature (e.g., framework, model, level, theory, typology, etc.). No additional searching was done for terms that are near synonyms of advice, e.g., recommendation or guidance. Searches were limited to articles published in English between January 1940 and November 2017. Searches were further limited to finding keywords in article titles only. Nevertheless, this search strategy resulted in a large number of false positive results which had to be set aside. This was accomplished by screening the content of titles and abstracts. Included in the final sample were 48 scientific papers, 5 of which were added from other sources. Two scholarly books were also included.

The conceptual analysis unfolded by reading and marking relevant portions of the works included in the study sample. These relevant portions speak directly to one or more aspects of advice theory. Next, concepts about advice were documented and related using a visual mapping technique and specialized concept mapping software (MindNode Pro, Version 1.11). As the work to analyze concepts during the present study progressed, an opportunity to organize its results emerged when it became clear that advice has been conceptualized in potentially different ways by scientists studying human-to-human advice interactions versus machine-to-machine advice interactions versus ICT-intermediated advice interactions.

While this conceptual analysis only enables partial refinement of reported understandings of the concept of advice, nevertheless it does advance a refined theory of advice that spans all three previously mentioned advice interaction modes. The results reported represent a first step toward a theory of advice spanning all advice interactions. It is important to note that the literature sample used for this conceptual analysis is diverse yet imperfect and incomplete.

RESULTS

The results unfold sequentially in the order of the research questions posed for this study. First, results that address *RQ1* are presented by summarizing the three nearly distinct lines of inquiry found in the scientific literature. For results that address *RQ2*, an advice content framework is developed. This new framework is organized along a continuum of environmental uncertainty encompassing decision accuracy, defensibility, and decidability. A content-based typology of advice is then derived from this framework and described. For results that address *RQ3*, a model of an advice interaction, informed by theories of decision-making, information, and documents, extends previous theorizing to arrive at a refined theory of advice.

A note about the context of all advice and advice interactions. As a lead-in to the results, the critical role that context plays in all advice interactions is reiterated here. Unless context is accounted for, advice is likely to be of little value[79]. Here, what is meant by context is the set of background phenomena, or the field of action, within which a focal event is embedded, such that the field of action and the focal event inform each other[80]. Advice interactions are focal events. They are embedded in the larger context of a decision situation, which in turn is embedded in the even larger context of the informational life-worlds of advisor(s) and advisee(s)[81]. Accordingly, the potential exists for advice to be informed by both the context of the decision situation and the informational life-worlds of those participating in an advice interaction.

***RQ1*. In which ways have researchers approached the analysis of advice as a concept?**

***Advice for human-to-human advice interactions*.**

This section reporting results about advice for human-to-human advice interactions unfolds in two parts. For Part One, a series of domain-specific ideas and theories about human-to-human advice interactions are described to give a sense of their scope, variety, and complexity. Second, to expand on key ideas surfaced in Part One, several domain-agnostic advice theories are examined in Part Two.

Part One of advice for human-to-human interactions:

Domain-specific ideas about advice for human-to-human advice interactions. Business, economic, and financial advisors play important roles in the modern economy. Policy advisors provide advice to people in powerful positions. Counselors provide interpersonal advice to individuals. Some key ideas and theories about advice were identified from all three of these domains.

Downs[82] pointed out the lack of theory about giving people economic advice. He asserts that the purpose of economic advice is to reduce the likelihood of expensive mistakes. He goes on to point out the need to organize the activities of decision analysis properly, especially their timing and scope, so that advice is valuable.

Blankart[83] contrasts those economic advisors wanting to bring about policy consensus, regardless of their own views, with other economic advisors wanting adoption of their own proposals, regardless of the views of policy makers. He then hypothesizes that *advice effectiveness* depends on the advisor finding some middle-ground between these two extreme positions. Harvey [84] takes this idea a step further and posits that economic advisors must settle on two things when giving advice: (1) the content of the economic advice they will give, and (2) the political tactics they will use to "package" their advice sufficiently so that it might be acted on. The works of these and other authors indicate that it is naïve to believe that human-to-human advice interactions involve a straightforward sharing of expertise without rhetoric[50,85].

From the perspective of political science, Bryson[9] theorized about how expert advisors use advice to inform policy decisions made by powerful actors. His work, "Notes on a Theory of Advice", describes using advice to give new dimensions of freedom to the thinking of advisees in power, thereby expanding the "range of choices" they have to consider[9]. Davies[86] takes this notion one step further and argues that advice must, "offer new dimensions of freedom of action." He claims that if advice is given as a recommendation to select one alternative over all others then it is actually not advice at all but is instead a plea[86].

In their "Theory of Policy Advice", Swank, Letterie, and van Dalen[87] model four things: the limited goals of policy-makers, the multiplicity of uncertain consequences arising from any policy choice, the differing opinions of experts weighing in on the same topic in ways that are implicitly value-laden, and the power to select experts, which is wielded by the policy-makers themselves. Using a game theoretic analysis, they demonstrate that policy advice is effective when it diminishes the bias ascribed to one policy maker by another. This result lends support to Blankart's[83] middle-ground hypothesis of advice effectiveness.

Recently, policy advice researchers have explored the concept of a policy advisory system (PAS)[88]. In this work, policy advice is thought of both as a general mechanism for government knowledge utilization and as a way to inform specific policy decision situations. A PAS is a complex set of policy actors and organizations, each with their own goals and constraints, that interact to generate advice that informs policy decisions. Examples of interacting "advisory units" comprising a PAS include government agencies; commissions of inquiry, expert panels, external consultancies, and citizen advisory boards[89]. Some work investigates how PAS dynamics vary amongst polities. Here, what is of greatest interest are the types of advice that have been identified through PAS studies. These include political advice targeted at maintaining power for those in office, public service advice targeted at bringing social gains, and technical advice targeted at issues of the feasibility and the potential consequences of alternatives.

Switching from the macro-perspective of policy advice to the micro-perspective of counseling individuals about choices in their personal lives, relational-cultural theory has been applied towards building growth-fostering relationships between human advisors and advisees [90]. Relational-cultural theory invokes three concepts: relational awareness, mutual empathy, and authenticity. By advising with these concepts in mind, advice is put in service of the larger goal of creating a "mutual, relevant, nurturing, and valuable" advising relationship [90]. Advising in this manner expands the potential scope of personal advice interactions to include discussions of the power dynamics within advisor-advisee relationships (relational awareness), intentional role reversal so that advice giving and receiving become bidirectional (mutual empathy), and reciprocal demonstrations of trust (authenticity).

Part Two of advice for human-to-human interactions:

Domain-agnostic theories about advice for human-to-human advice interactions. Beginning the 1980s, coincident with other initial approaches to the study of naturalistic decision making [36], the Judge Advisor Systems Theory (JAS) was developed by Sniezek [91]. JAS enables studies of the psychology of human judgment and decision-making. It provides a framework for studying how human decision makers conduct social information search involving other people. In the JAS model, the human "judge" has the authority to take a decision after one or more pieces of advice are contributed by one or more human "advisors."

Numerous studies since have used the JAS model to explore advice and advising, mostly in for decision-making tasks involving uncertainty [23]. However, most JAS studies have operationalized advice extremely narrowly as a recommendation in favor of one alternative [92].

Some JAS studies have added advisors' expressions of confidence to their recommended alternative [93], expanding the scope of the advice content used in JAS studies somewhat.

Bonaccio and Dalal note that advice theory, in particular, has received inadequate attention by those who do JAS studies [23]. These authors argue for a construct of advice that includes (a) recommendations for a *single alternative*, (b) recommendations *against* one or more alternatives, (c) factual, non-normative information about alternatives, and (d) advice targeted not at the original target decision but instead at the "meta-decision" of *how to decide it* [92].

As a collection, JAS studies generally confirm that judges discount advice from advisors too much, perhaps by intuitively evoking their own efficient but faulty cost-benefit estimates [23]. This bias that advisors have towards their own beliefs is called egocentric discounting of advice [94]. Most relevant here, however, are the results from JAS studies related to decision accuracy. How decision accuracy has been operationalized and measured is of special interest because it reflects the nature of the advice examined in JAS studies.

To measure decision accuracy, a correct answer for each decision problem has to be known to the study team in advance. For this reason, JAS studies have compared the decisions made by judges who received advice from advisors about the following things: prices for consumer goods (e.g., prices of backpacks) [95], the dates when historical world events occurred (e.g., the date when the Suez Canal first opened) [61], the answers to multiple-choice questions of computer literacy [93], and the fictitious perpetrators in "murder mystery" scenarios [96].

Regarding decision accuracy, JAS studies have shown that, in general, using advice increases decision accuracy [22,23]. In these studies, decision accuracy has been correlated with information availability. Specifically, decision accuracy increases as either the amount of decision relevant information available to advisors, or the amount of performance feedback information available to judges, increase [23].

Studies in communication science investigate advice in the larger, general context of "supportive interactions." A supportive interaction begins when support-seeker discloses a problem, and may prompt others to offer emotional support, comfort, assistance with coping, and advice [57]. If choices have to be made to overcome the disclosed problem, support-seekers may become decision-makers. When they do, they may benefit from advice.

In this context of supportive interactions, two factors from debate theory, “cure” and “cost-benefit”, have been used as starting points to model how to evaluate advice message content [57]. The resulting advice content evaluation model posits that advice message content quality, as perceived by advisees, depends on the advisees’ receptiveness to advice and also on advice message clarity, relevance to the decision problem, the degree to which the advice mitigates threats to face, the feasibility of its advised solution(s), and the absence of limitations to its advised solution(s). An initial study indicated that these factors were all significant and accounted for 51% of variance in perceived advice message quality.

Additional research on advice in the context of supportive interactions has demonstrated the effectiveness of an integrated model of advice giving that consists of three sequential moves for “supportive advisors” to make. These sequential moves that human advisors make are first to offer emotional support, second to perform problem analysis work in conjunction with the advisee, and third to formulate and communicate advice [97].

Further work in communication science has led to the development of Advice Response Theory (ART) [98]. ART is an attempt to move beyond feature-level analysis of advice messages to examine how advice message, advisor, and decision situation combine to influence decision outcomes. Several studies using ART give an early indication that the influence of advice message features is greater than the influence of advisor characteristics in the minds of advisees [99]. This work suggests that, during a supportive interaction, what is said in an advice message, and how it is said, may have more influence on advisees than who says it.

Advice has more recently been studied from a social psychology perspective using Construal Level Theory (CLT). CLT proposes that people perceive objects and events differently according to the temporal, spatial, social, and hypothetical dimensions of distance they perceive between themselves and an object or event of interest. Overall, a higher construal level (i.e., greater "psychological distance") from an object or event of interest elicits more abstract thinking about it whereas a lower construal level (i.e., less "psychological distance") elicits more concrete thinking about it. CLT has been used in an attempt to explain why advisors may approach a decision situation differently from advisees. Studies using CLT suggest that advisors think differently than advisees about decision problems when the advisors are not directly impacted by consequences of the decision [100]. Higher construal levels have also been associated with a greater degree of utilitarianism in decision-making[101].

Juerta[11] published a book on advice analysis within the field of management studies. Here it is only possible to give a superficial indication of the depth of this work. In it he carefully elaborates his argument for the following definition of advice (p. 106):

> "Advice: Any instance *x* of the concept advice must satisfy the following identity criteria:
>
> 1. *x* is some potentially complex speech act that has been performed by an individual called an advisor
> 2. *x* has been experienced by an individual called a recipient
> 3. the recipient can distinguish the dicta (what is said) and the modi (how it is said) in *x*, and from the modi establish which of the dicta he could adopt as beliefs, desires, intentions, or evaluations
> 4. the recipient can form reference relations between at least some of the dicta in *x* and objects in his context of reference"[11]

Jureta views advice as a tool of social communication whose purpose is to coordinate people and things. For him, advice is information given by an advisor to a recipient (i.e., advisee) with the over-arching intent of better orienting the advisee to the world around them. To arrive at the definition above, Jureta uses a new "model of choice" based on intentional states of mind. In this model of choice, a person can only communicate four things: their beliefs, desires, intentions, and attitudes. (To improve clarity, the term "attitude" is substituted here for Jureta's term "evaluation." Here, both terms generally refer to a like or dislike, or to a feeling or disposition.)

Per Jureta[11], advice is generated in the following way. Advisors begin the process of formulating advice by ascribing intentional states (i.e., beliefs, desires, intentions, or attitudes) to advisees, either accurately or inaccurately. Advisors also come to understand something about the decision situation. Next, advisors generate fleeting, speculative, also called naïve [102], mental picotheories about the advisee(s) and the decision situation. Then, advisors apply their picotheories to arrive at beliefs, desires, intentions, or attitudes of their own. Finally, once they have settled on relevant intentional states of their own, advisors communicate their relevant intentional states as advice. Advice generated in this manner either *supports* or *repudiates* the beliefs, desires, intentions, or attitudes (i.e., intentional states) of the advisee. Jureta's theory of advice as a message formed by advisor via ascription and prediction results in a typology of eight functional advice types. In sum, Jureta's theory could be called a theory of advice as Messages of Ascription and Prediction (MAP).

Two more related topics in the literature about human-to-human advice interactions are the topic of *advice-seeking behavior* and the topic of *how human advisees combine* (or *integrate*) *advice from multiple human advisors* when making decisions. It has been noted that most models from social psychology of how humans use information from multiple sources assume either that all information sources have equal influence or that all sources are associated with some unequal but fixed weight of influence[103].

The dynamics of weighting advice from multiple advisors is starting to get some attention from scientists. Through a series of experiments, Hütter and Ache [94], who view advice-taking as an adaptive process, recently demonstrated that participants increase their sampling of advice when advice is more informative and divergent. When integrating this advice, participants gave greater weight to advice that differed most from their initial estimates and advice that was consistent with other advice. This work reinforces earlier results from Calvert [62], who undertook a thought experiment to demonstrate why advisees stand to gain when they seek the opinions of like-minded, and thus predictably biased, advisors. The reason, explains Calvert, is that advice from an advisor who is known to think like the advisee, but which runs counter to the advisee's expectations, is more surprising and thereby more informative.

This section ends with a brief mention of the growing science of human advice networks. An advice network is a pattern of relationships in which one person requests and receives advice from other people [22]. This research applies social network theory to study advice network (and subnetwork) effects. As examples, Nebus [104] explored how advice networks may differ from social networks in ways that reflect that knowledge asymmetry between expert advisors and advisees. In turn, in the context of a single government organization with multiple internal business units where an advice network already existed, Zappa [105] was able to distinguish between within-unit advice relations on the network, which were shaped by status recognition, and cross-unit advice relations on the network, which were instead shaped by norms of reciprocity.

Presupposing an advice network, Perry has articulated a framework for providing scientific advice that suggests routine processes of learning will result in incremental updates to advice provided to a community of interest [106]. Friedman has previously articulated a similar idea for the domain of human health called the Learning Health System[107].

Advice for machine-to-machine advice interactions

This section reports results from the fields of artificial intelligence, theoretical computer science, and distributed network computing. It is relevant that, for distributed computing, both data and software code are automatically shared and processed among multiple machines. Hence, the need for machine-to-machine advice often arises in the context of distributed computing.

In an early approach to the development of artificial intelligence capabilities, McCarthy[24] investigated the antecedents to everyday predictions human beings make about the world. He noted humans represent in their minds facts about situations, goals, and the effects of actions so as to “draw conclusions about which sequences of actions are likely to achieve our goals.” Building on this observation, McCarthy defined computable advice as a series of factual statements represented in a manner that allows the statements to be input into a computing machine. Using such inputs, a machine that has previously been programmed to reason over given factual statements was shown to generate new inferences. McCarthy thus demonstrated that machines can do basic logical reasoning for simple decision situations.

In turn, Michie[71] conceptualized advice from the perspective of machine intelligence. In his paper entitled, "Theory of Advice", Michie conceived of advice as workable approaches to solve semi-hard problems. Semi-hard problems are those computational problems that have solutions which cannot be calculated due to practical limits on computer memory or time. With semi-hard computational problems in mind, Michie described two types of advice. First there is the "great leap forward" type of advice. This type of advice provides a fundamentally new approach, in the form of a novel computer algorithm, to enable a semi-hard computational problem to be solved with available computer memory or time. Second there is the "incremental" type of advice. Michie's incremental advice takes the form of a new rule that saves time or computer memory when it is added as a refinement to an existing computer algorithm. In both cases, Michie's advice is formulated as executable computer instructions, i.e., as "code." Although Michie does not mention machines advising other machines, given the transferability of software code, when one machine shares Michie's *advice as computer algorithm* with another machine, a machine-to-machine advice interaction occurs. Machine-to-machine advising like this happens, for example, when software updates are automatically applied via a computer network.

A number of researchers in the domain of machine learning have conceptualized advice as a predicted or estimated probability, e.g., a number p between 0 and 1 that can be interpreted as the probability of some future event of interest [108]. Others have similarly conceptualized advice as a probability distribution [72,109]. Using advice from multiple advisors in the form of probability distributions, and taking a game theoretic approach, Chung [72] showed how to effectively combine this advice mathematically to compute both optimal and high-performing approximate results for a class of sequential prediction problems including choosing a financial stock portfolio from among m financial stocks. In a similar way, using advice from multiple advisors - in the form of estimated probabilities - a method has been developed to approximate what is otherwise a computationally intractable min/max prediction strategy that minimizes the maximum of the difference between the advisee and the best performing advisor over time. Machine learning applies to situations like these where multiple data sources are used to generate advice (i.e., point estimates or probability distributions). Machine learning allows advice to effectively be combined computationally to make increasingly accurate predictions over time.

The aspect-oriented programming paradigm explicitly uses a concept of advice pioneered by Selfridge[110], Teitelman[111], and Minsky[112]. In this case, advice takes the form of a special class of computer functions that modify or extend the behavior of other computer functions. Advice functions are invoked before, during, or after other functions. They are designed to add common, cross-cutting capabilities to many different software applications, e.g., security capabilities. Quite recently, coordination languages that were originally created to support inter-process communications amongst machines have been extended to support fully automated aspect-oriented advice functions too[113,114].

Finally, the profound effects of large-scale computer networking also appear in this literature about machine-to-machine advice interactions. Studies have been published describing online computation benefitted by machine-to-machine advice. Online computation refers to networked systems that process requests in the order they are received. Advice for online computation is conceptualized as some function of an entire request sequence or request queue, i.e., advice arising as part of a computational oversight function intended to help coordinate distributed request processing overall. Computer scientists are interested in incorporating this type of machine-to-machine advice into transactional requests for the purpose of improving request processing performance of networked systems[109,115].

Advice for ICT-intermediated advice interactions

This section, reporting results about advice for ICT-intermediated advice interactions, principally addresses advice coming from decision support systems (DSSs). Here, a DSS is simply a computer system that generates advice. Therefore, DSSs include expert systems, computer-based advisors, and all other advice-giving systems.

Advice interactions involving DSSs are human-to-human interactions only in a very limited and distant sense. This is because humans design the advice-giving systems that generate the advice DSS users receive. Of course, users may attribute the advice they get from DSS to the systems themselves instead of to the human designers of that advice[116,117].

This section, reporting results about advice for ICT-intermediated human-to-human advice interactions, like the previous one, unfolds in two parts. First, a series of studies on DSSs are reviewed to develop a picture of ICT-intermediated advice interactions. Second, studies addressing human learning as a consequence of using DSSs are specifically treated in brief.

Part One of advice for ICT-intermediated advice interactions:

Developing a picture of ICT-intermediated advice interactions. What is specifically meant in the literature by the term decision support in not entirely clear. Here, decision support is thought of generally as a capability of information and communication technology to play the role of automated advisor within advice interactions. Some partial theories of decision support can be inferred from the approaches to DSSs taken in the following studies.

In early work on a DSS, within the problem domain of the card game Hearts, Mostow studied how a computer system can operationalize general advice that it receives in the form of computable heuristics [118]. Using automated means-ends analysis and previously encoded rules to solve problems, it was shown that a computer system can generate abstract plans based on computable heuristics. These abstract plans can then either be automatically applied by the computer, or shared with human game players, during Hearts game play.

Hadden [119] describes two impacts of Intelligent Advisory Systems (IASs), which are defined as "computer programs that use the techniques of artificial intelligence to mimic human expertise." (As such, IASs are DSSs.) She first describes a momentary function where DSSs apply computable knowledge to enable data retrieval and subsequent interpretation for their human users. She also describes a more general effect of DSSs, which is to make human expertise computable and thus more widely available to other people.

Gilbert and colleagues[120] describe a DSS intended to help users explore a solution space, weigh alternatives, and understand the reasons for the advice offered by the system. The DSS they describe helps people understand social security benefits in the UK. Twenty different types of answers are supported by this DSS, which relies on 4 types of computable knowledge applied at 3 different decision problem levels. The four types of computable knowledge are (1) *taxonomic entity-relationship knowledge*, (2) *formal analytic knowledge about relationships between entities' attributes*, (3) *contingent "how to" procedural knowledge about the social security system*, and (4) *control knowledge about the DSS itself*. The three decision problem levels are the individual case level, the general benefit domain level, and the theory level of meta-domain principles about social security. This DSS remembers prior moves made by its users and uses those records to support an iterative advice interaction.

Licker[102] describes DSSs as "systems that assist in the coming to conclusions." He then defines five types of DSSs, each one offering a different type of advice to its users. Moving from the simplest to the most complex systems, Licker's five types of DSSs are: Display Systems giving text and images using data models, Management Information Systems giving reports using information models, Consulting Systems giving analyses using similarity models, Expert Systems giving explanations using heuristic models, and "Simulation Systems" giving predictive outputs using theoretical models.

Hoch and Schkade[121] studied DSSs and the psychology of human decision-makers. They asked, "Is it better for a decision support system to capitalize on human strengths or compensate for human weaknesses?" This research question goes directly to the allocation of functions between human and machine[122]. In their study, human strengths were operationalized via pattern matching using historic data. Meanwhile, human weaknesses were operationalized by using a prediction model involving a statistical combination of random variables. Hoch and Schkade found that, for a business forecasting task, the degree of environmental uncertainty mattered. Under conditions of high uncertainty, when a linear predictive model was used to consistently combine random variables, having the result computed with the model improved human decision accuracy. This effect was greatly muted under conditions of low uncertainty. Their findings align with decades of research on the usefulness of statistical prediction[123].

Aspinall [124] has since commented on the importance of quantifying uncertainty for human decision-makers in order to improve decision-making. He notes that computational methods, such as the Cooke method [44], exist to combine previously elicited "subjective probabilities" provided by experts into advice with explicit representations of uncertainty for people. Of course, other methods exist to quantify statistical uncertainty associated with random variables [125].

Jones and Brown[126] examined the decision of labor between people and DSSs directly. They investigated whether or not decision-makers use a systematic "divide and conquer" approach when receiving "modeled" and "non-modeled" advice from a DSS. Modeled advice is generated using a statistical prediction model to combine multiple facts into a single result. Non-modeled advice (which Jones and Brown call "information") takes the form of multiple independent facts. They found that humans did not exhibit a theoretically optimal "divide and conquer" strategy when using the studied DSS.

Part Two of advice for ICT-intermediated advice interactions:

Human learning as a consequence of ICT-intermediated advice interactions. What Hadden[119] claims is that DSSs are a mechanism for transferring knowledge among people. This suggests that people can learn lessons from DSSs which they can apply immediately to a decision problem and again later to future decision problems. Indeed, Anstey [127] wrote that, "in many cases the ideal advisory encounter will become a teaching activity, so that the user is better informed and will hopefully be able to avoid not only identical problems but similar ones."

A large body of literature on trust in automation suggests that indeed people do learn from repeated advice interactions with DSSs. In studies of task allocation between humans and machines, Muir[60] noted that trust between humans and machines must be "calibrated." Trust is an evolving, uncertain prediction made by people reflecting their willingness to be vulnerable[59]. Calibration of trust involves a person updating their prediction model about what a DSS will do as new experience becomes available. Existence of such experiential learning resulting from

DSS use has been confirmed. Studies show that DSSs have to give advice users expect for easy decision problems otherwise users will assume the DSSs are incapable of providing reliable advice for difficult ones[128–130].

Gardner and Berry[131] demonstrated a direct learning effect from DSS use. In a study of drug dosing advice, a DSS was shown to have a significant impact on human learning in two ways. First, access to DSS advice in the past was associated with improved dosing in the present in the absence of advice. Second, a group receiving explanations along with dosing advice outperformed other groups on a questionnaire about dosing methods.

Finally, there is currently significant interest in understanding how learning may result from network-enabled DSSs. The Science of Learning program at the National Science Foundation [132] and a growing emphasis on developing Learning Health Systems [107] presage a body of scientific work on network-enabled DSSs capable of unicasting and multicasting computer-generated advice.

***RQ2*. What types of advice emerge by analyzing advice-as-an-information-object using a framework that spans decision accuracy, decision defensibility, and decision decidability to encompass a content-based typology of advice?**

A content-based advice typology

The answer to *RQ2* is given in the form of an advice content framework (Figure 1.3). This framework gives rise to a new advice typology or categorical scheme for advice. Unlike the previously reviewed "functional" advice typology of Jureta or the "format-based" advice typology of Liker, what follows is a content-based advice typology. It categorizes advice messages into four specific types of content, depicted by four numbered "buckets" in Figure 1.3.

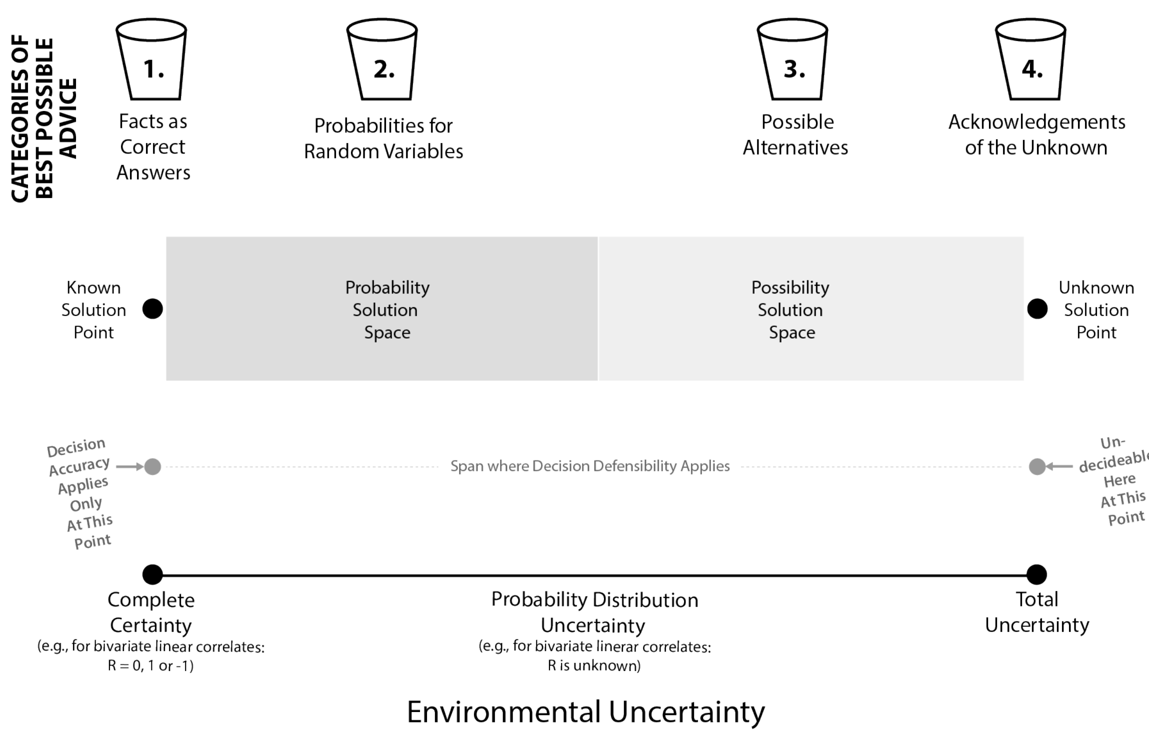
The concept of *environmental uncertainty* is foundational in this result. Recall that environmental uncertainty is the degree to which a future state of nature can be accurately predicted with information and knowledge in the world[41]. The advice content framework in

Figure 1.3 places environmental uncertainty on a horizontal axis and relates it to decision accuracy, defensibility, and decidability on one level, to related solution points and spaces on another level, and to the four types of advice content – 1. facts, 2. probabilities, 3. possibilities, and 4. acknowledgments of the unknown –on yet another level.

When decision problems arise where there is no environmental uncertainty, it means correct solutions to the problems are known in the world. (Of course, many advisors and advisees may remain ignorant of these correct solutions.) In such cases, the solution is known to be completely certain and decisions can be judged for accuracy. The content of the best possible advice message for these decision situations is the correct answer to the decision problem. An example of this is advice that gives an accurate, definitive diagnosis of a medical disease.

Some decision problems arise where there is statistical uncertainty due to random variables, and so their solutions involve probabilities. In these cases, the quality of the decisions made must be judged in terms of decision defensibility and not decision accuracy. Setting aside

Figure 1.3 Advice Content Framework



the magnitude of risk involved, a decision will often, but not always, be more defensible if it is keeping goes with, and not against, the odds or likelihood of a result. Here, success means maximizing benefits while minimizing costs for the decision-maker. In such situations, the content of the best possible advice takes the form of *probabilities (or probability distributions)* for random variables. An example of this is advice, in the form of odds, that suggest to a Blackjack player when to hit or when to stand.

Other decision problems arise where there is even greater environmental uncertainty. These problems are associated with a possibility space where no random variables pertain. Instead, what pertains are only sets of possible alternatives. The quality of the decisions made under these conditions must also be judged in terms of decision defensibility, which will depend in part on awareness of possible alternatives. A decision made without awareness of all possible alternatives may be less defensible, depending on its outcomes. In these situations, the best possible advice takes the form of *possible alternatives*. An example of this is advice that tells a student what elective courses they may benefit from taking.

Due to total environmental uncertainty, some decision problems cannot be solved. These are decision problems where the solution is either unknown or unknowable, making these problems undecidable. Undecidable problems call for a guess and not a decision. In these cases, the best possible advice *acknowledges the total environmental uncertainty* that pertains. A technical example of this is the "halting problem." For this problem, it has been proven that it is impossible to decide, given an arbitrary computer program and an input, whether the program will halt or continue to run forever [133]. A biological example is the impossibility of deciding precisely which individual cell first gave rise to a tumor. In both cases, the best possible advice can only acknowledge the impossibility of finding a solution to the problem.

Summarizing the typology that emerges from the framework in Figure 1.3, a content-based typology of advice, spanning the full range of environmental uncertainty, includes four types of advice: facts (i.e., correct answers), probabilities for random variables (e.g., bivariate linear correlations), possible alternatives, and plain acknowledgements of the unknown. Because the full range of environmental uncertainty has been spanned, this result asserts that no other content-based types of advice exist. It is hypothesized that these four categories are indepdendent. Thus, future work is needed to determine whether or not these four advice content categories overlap to any degree.

***RQ3*. How may advice be conceptualized in light of existing advice theory while accounting for a content-based typology of advice?**

A refined theory of advice

The answer to *RQ3* unfolds as prior advice theory is analyzed in light of the above content-based advice typology. What results is a refined theory of advice.

It is asserted, with support from document theory[29], that advice is an information object. In addition, because essentially all advice theories agree on this point, it is further asserted that advice is generated and transmitted during advice-interactions within the larger context of a decision situation. To address RQ3, the following question thus becomes relevant: Can the concept of advice be further qualified using existing advice theory and the content-based typology of advice above?

Bryson[9] contends the purpose of advice is to give new dimensions of freedom to the thinking of advisees, thereby expanding the number of available alternatives advisees' have when they are facing decisions. Likewise, Davies [86] argues that advice must "offer new dimensions of freedom of action" to advisees. However, Jureta [11] directly counters Bryson and Davies by arguing to the contrary that, "Advice should have the capacity of constraining choice. It should eliminate alternatives, occasionally to the extent that only one remains."

Applying the above content-based typology of advice, it is evident that advice can either expand, contract, or leave the number of alternatives about which the decision-maker is aware unchanged. Advice may expand an advisee's range of choices for decision problems associated with a possibility space. It may contract an advisee's range of choices for decision problems associated with a probability space. It may also contract an advisee's range of choices down to a single correct choice for decision problems associated with a correct answer.

Some advice researchers have asserted that advice is always normative. However, there are examples of advice described in the literature suggest otherwise. Leppanen [50] claims advice is "overtly normative" because it proposes a preferred course of future action. Pilnick [55] claims advice has a normative character while factual information, which she claims is different from advice, is non-normative. This distinction between advice, which is thought by some to always be normative, and "information", also appeared in the work of Jones and Brown on DSS [126]. Yet in studies that use the Judge-Advisor System, advice is typically operationalized as advisors' estimates of known facts [23]. Interestingly, in computer science advice has been thought of in strictly numerical terms, as probabilities or probability distributions[72,108,109,115] , or as computable rules expressed in software code[71,113,114].

The above content-based typology of advice clearly encompasses normative and non-normative advice. However, the confusion over whether advice is normative or non-normative cannot be entirely resolved by understanding its content alone. As the concept has been further developed here, advice is an information object that, upon being received by an agent, gives rise to an advice document through a meaning-making transaction. The meaning ascribed to an advice document is therefore also relevant to this question of normativity.

The perspective of documental becoming makes a distinction between advice, as an information object, and an advice document. Using that distinction, cases are easily outlined where advice messages carry non-normative content while the corresponding advice documents they engender are given normative meaning. As an example, if an advisor with reliable information gives factual and accurate advice that, "the price of gasoline is cheaper today than it will be tomorrow" to an advisee who needs to purchase gasoline soon, the advisee can draw from this factual advice a normative meaning, e.g., "you should purchase gasoline today and not tomorrow." In this case the advice is factual while the normative meaning drawn from it, corresponding to an advice document, is not. Normative meaning may be ascribed by an advisee to an instance of advice even though it may not have been intended by the advisor.

A similar issue mentioned in the literature is whether or not advice can be suggestive or persuasive[103,126]. Again, this determination may in part depend on the meaning-making transaction that gives rise to an advice document. Yet it is clear from the above advice-content typology that advice may impart an argument, e.g., an argument that a particular decision problem is undecidable. Furthermore, in terms of its function, Jureta [11] asserts that advice supports or counters beliefs, presumably also through argumentation. The result is that advice may or may not be suggestive or persuasive.

Figure 1.4 illustrates the resulting refined theory of advice that is the answer to the *RQ3*. In this figure, advice is depicted as the result of documental becoming within the context of an advice interaction taking place within some broader decision situation.

After receiving a cue of some sort, an agent, acting in the role of advisor, formulates advice, which is an information object or message. At that moment, the advisor draws meaning from the advice through a transaction leading to an original, situated advice document, signified *advice document*°. Next, the advisor may transmit, either verbally or in writing, the advice message. At that point, the action in Figure 1.4 switches to an agent acting in the role of advisee.

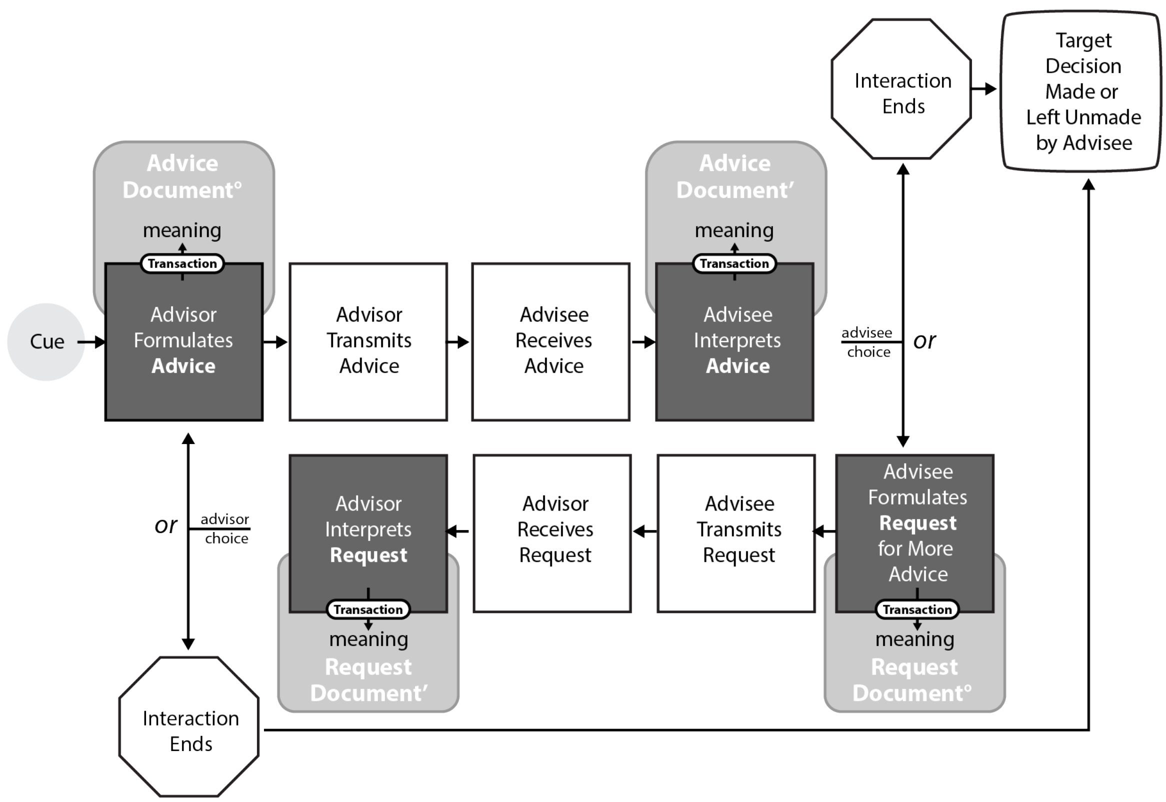
When an advisee receives a message of advice that message may be interpreted by the advisee. Through this transactional interpretation process the advisor ascribes meaning to the advice, leading to a new, situated advice document, signified *advice document'*. The advisee can then choose whether or not to end the advice interaction and also whether or not to exit the decision situation.

During an interactive advice interaction, the advisee may elect to formulate and transmit a request message asking for more advice. Such a request proceeds in an identical fashion through transmission, receipt, and interpretation by an advisor (Figure 1.4).

A careful analysis of this advice interaction portrayed in Figure 1.4 indicates it is complex. It has within it two embedded decisions, making it potentially recursive. This recursion happens in practice when advice is given about the decisions to ask for or provide more advice. To that point, upon close inspection, the identical nature of the "request" pathway between advisee and advisor reveals that, although it seems quite strange, a *request for advice* counts as advice too! To its recipient, a request for advice communicates a desire [11] and it communicates a frame for a choice with two possibilities, which are to meet or deny the request. As such it imparts content about possible alternatives. Seeing that a *request for advice* is itself also a form of *advice* highlights the coordinating effect of advice. Advisees and advisors may coordinate advice interactions in part through cycles of advice requests and advice responses (Figure 1.4).

Finally, although it is known that advice is more acceptable when it takes account of the particulars of the decision situation within which it occurs, there is no requirement that advice be acceptable. Does this mean that all information can be construed as advice? No, it does not. Recall that the result here places advice in the context of an advice interaction taking place in the context of a decision situation. For advice to be realized, there must be a preexisting decision situation into which that advice comes, a situation associated with a pending, unmade decision.

Figure 1.4 Refined Model of an Advice Interaction



For this reason, while it is possible to conceive of advice that is almost wholly uninformed by the details of the decision situation within which it arises, the fundamental fact remains that advice must always *pertain* to some decision situation, otherwise it is not advice. Evidently, this decision-pertaining relationship is central to the concept of advice. To indicate this relationship without suggesting that advice is otherwise informed in any way by the decision to which it pertains, it might be sufficient to say that advice must always *target* some decision. The verb target is used here in the sense of "fix on", signifying that advice is fixed on, or attendant to, an unmade decision.

Summarizing this final result, a number of potential further qualifications of the concept of advice have fallen away through this analysis. Thus, it is theorized that every instance of advice is an information object targeted at an unmade decision.

DISCUSSION

The present study examined advice and advice theory in a comprehensive way to better conceptualize advice. Using insights from document theory and information theory, advice was analyzed via a systematic literature review. An organizing principle for advice studies was identified when it became clear that studies either focused on advice in the context of human-human, machine-machine, and ICT-mediated advice interactions. Synthesizing many previous works across these three modes of advice interaction, what resulted is a conceptualization of advice as an information object targeted at an unmade decision.

Within the literature about advice, a shift from domain-specific to general notions of advice is noticeable[9,22,92]. The domain of computer science is an exception to this. There, advice was conceived of in very general terms early on[24,111,113].

Some more recent studies of advice interactions that are human-human, machine-machine, or ICT-mediated, put an emphasis on advice networks. For human-human advice, advisee-advisor networks have been contrasted to other types of social networks[22]. For machine-machine advice, Internet-scale distributed computing has made computable advice more wide-reaching and impactful than in the past[109,113,114]. For ICT-mediated advice, an emerging science of learning, focused in part on ongoing, global scale human learning enabled by networks, is coming to the fore[77,107,134,135]. These works presage a future when the preponderance of advice in the world may be generated by machines and distributed over networks to people and other machines.

An ongoing challenge with machine-generated advice is that it generally arises from computer procedures that are strictly parameterized, inevitably biased [136], and in which people often place too much confidence [137]. The resulting advice provides sufficient assistance in many cases, but it is also greatly constrained by preconceived notions of decision situation context [16]. However, newer ways of representing computable knowledge so that it can be automatically enriched and combined, such as those associated with the semantic web [138], are starting to enable more context-aware and adaptive machine advice-giving. Yet there will always be limits on the amount of information about decision situation context that can be considered by agents acting as advisors.

A variety of typologies or categorization schemes for advice have been surfaced in this review. Juerta's [11] scheme is described here as a "functional" one because it categorizes advice in terms of what it does, which is to support or refute beliefs, desires, intentions or attitudes. Licker's [102] scheme is different. It describes, in an imprecise way, various forms of advice coming from DSSs on a rough scale spanning data, information, analyses, explanation, and simulation. Bonaccio and Dalal [92] also articulated several different types of advice, including arguments for or against a single alternative, general information about an alternative, and advice on *how* to made a decision.

Building on these thoughts with concepts related to environmental uncertainty, a new content-based typology of advice has been developed in this chapter. In this typology, the best possible advice is defined for four different states of environmental uncertainty. When a decision problem has a correct answer, that answer is the best possible advice. When a decision problem involves random variables, probabilities with the greatest possible precision constitute the best possible advice. When a decision problem is uncertain to the degree that only possible alternative moves are known, then the best possible advice surfaces those possible moves. Finally, when a decision problem is undecidable, acknowledging that it is so is the best possible advice.

As a pragmatic example of how the content-based typology of advice applies in practice, consider an unfortunate case involving all four advice types. Jeanette, a 42-year old mother of three, suffers rapid and profound hearing loss three days after taking a widely used medication for the first time. She stops taking the medication immediately. She needs to decide whether or not the medication precipitated her hearing loss. At first, the best possible advice comes from a pharmacist who acknowledges that there is no known evidence of a relationship between the medication and hearing loss. Next, an online search for possible and probable causes of rapid hearing loss provides additional relevant advice for Jeanette. Finally, through a process of medical examination, Jeanette receives a definitive diagnosis of Cogan's syndrome [139]. As a byproduct of this diagnosis, an accurate answer to the original decision problem is apparent. The time of onset for Cogan's syndrome eliminates the possibility that Jeanette's hearing loss stemmed from taking the medication for just a few days.

The case above highlights an important relationship between knowledge and advice. Knowledge in the world has the effect of diminishing overall environmental uncertainty. Advice is an information object targeted at a decision. Therefore, the best possible advice reflects the current state of all available knowledge about the decision problem at which it is targeted.

As mentioned at the outset, advice is a complex concept and not a self-evident one. When conceptualized as a message, advice can give rise to a situated advice document when it is encountered by an agent. Taken together, the results of this review suggest that the purpose of the best possible advice is to help advisees gain an understanding of what is known and relevant about a decision problem of interest.

CONCLUSION

In this chapter, a systematic review of advice theory led to an analysis of research findings organized according to three modes of advice interactions. These three modes of advice interactions include human-human, machine-machine, or ICT-mediated ones. Based on these findings, advice is conceptualized as information objects targeted at unmade decisions. Furthermore, through a transactional process of documental becoming, advice gives rise to momentary, situated advice documents in the minds of human advisors and advisees or in the digital registers of machines capable of generating and receiving advice.

Using the concept of environmental uncertainty, an advice content framework was developed which gives rise to a content-based typology of advice. Depending on the degree of environmental uncertainty that pertains to a decision problem, the best possible advice targeted at that decision problem either provides a factual answer that solves the problem, probabilities or possibilities that help address the problem, or an acknowledgement that the problem has not yet been, or cannot be, resolved.

This study of advice informs work to develop better advice-giving systems and advice-giving services. A major point that it makes is that networks of advice-giving systems are evolving for the purpose of providing advice to improve coordination in complex, distributed systems of interactive agency.

Advice-giving services combine knowledge with data about the context of a decision to produce their advice. For this reason, in the next three chapters we turn our attention away from advice toward computable knowledge. Specifically, we focus on increasing the interoperability of computable knowledge. We do this so that, by using current computable knowledge, the best possible advice can be produced by advice-giving systems to improve human health. We also do this so that networks of advice-giving systems can more easily be formed by building on better computable knowledge infrastructure.

References

1. Brown, G. E. Science + advice = Science policy advice: The role of scientific expertise in policy-making. *Res. Technol. Manag.* **36,** 9 (1993).
2. Williamson, E. G. & Bordin, E. S. A statisical evaluation of clinical counseling. *Educ. Psychol. Meas.* **1,** 115–131 (1941).
3. Oxford English Dictionary. Oxford English Dictionary Online. *Oxford English Dictionary* (2017).
4. 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).
5. Simon, H. A. *The Sciences of the Artificial*. (MIT Press, 1996).
6. Dexter, P. R. *et al.* A computerized reminder system to increase the use of preventive care for hospitalized patients. *N. Engl. J. Med.* **345,** 965–70 (2001).
7. Schrah, G. E., Dalal, R. S. & Sniezek, J. a. No decision-maker is an Island: integrating expert advice with information acquisition. *J. Behav. Decis. Mak.* **19,** 43–60 (2006).
8. The Fredonia Group. *Professional Services : United States*. (2016).
9. Bryson, L. Notes on a Theory of Advice. *Polit. Sci. Q.* **66,** 321–339 (1951).
10. Collier, S. J. & Ong, A. Global Assemblages Anthropological Problems. *Global Assemblages: Technology, Politics, and Ethics as Anthropological Problems* 3–21 (2008). doi:10.1002/9780470696569.ch1
11. Jureta, I. *Analysis and Design of Advice*. *Vasa* (Springer Berlin Heidelberg, 2011). doi:10.1007/978-3-642-18059-0
12. Hadfield, G. K. The cost of law: Promoting access to justice through the (un)corporate practice of law. *Int. Rev. Law Econ.* **38,** 43–63 (2014).
13. SEC. *Mutual Fund Fees and Expenses*. *Investor Bulletin* (2014).
14. UCSF. UCSF Health Second Opinion Service. (2017). Available at: https://www.ucsfhealth.org/secondopinion/. (Accessed: 17th November 2017)
15. Gino, F. Do we listen to advice just because we paid for it? The impact of advice cost on its use. *Organ. Behav. Hum. Decis. Process.* **107,** 234–245 (2008).
16. Bates, D. W. *et al.* Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inf. Assoc* **10,** 523–530 (2003).
17. Covell, D. G., Uman, G. C. & Manning, P. R. Information needs in office practice: are they being met? *Ann. Intern. Med.* **103,** 596–9 (1985).
18. González-González, A. Information needs and information-seeking behavior of primary care physicians. *Ann. Fam. ...* 345–352 (2007). doi:10.1370/afm.681.Jos
19. Rothschild, J. M. *et al.* The Critical Care Safety Study: The incidence and nature of adverse events and serious medical errors in intensive care*. *Crit. Care Med.* **33,** 1694–1700 (2005).
20. Hogan, H. *et al.* Avoidability of hospital deaths and association with hospital-wide mortality ratios: retrospective case record review and regression analysis. *Bmj* h3239 (2015). doi:10.1136/bmj.h3239
21. Smith, M., Saunders, R., Stuckhardt, L. & Mcginnis, J. M. in *National Academy Press* (2012).

22. MacGeorge, E. L., Feng, B. & Guntzviller, L. M. Advice: Expanding the communication paradigm. *Commun. Yearb. 40* **8985,** 213 (2016).
23. Bonaccio, S. & Dalal, R. S. Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences. *Organ. Behav. Hum. Decis. Process.* **101,** 127–151 (2006).
24. McCarthy, J. *Situations, actions, and causal laws*. (1963).
25. Shortliffe, E. H. A Rule-based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection. in *1974 annual ACM conference* 739–739 (1974). doi:10.1145/1408800.1408906
26. Tintarev, N., O'donovan, J. & Felfernig, A. Introduction to the Special Issue on Human Interaction with Artificial Advice Givers. *ACM Trans. Interact. Intell. Syst.* **6,** 1–12 (2016).
27. Searle, J. WHAT IS A SPEECH ACT? 1–16 (1965).
28. Buckland, M. K. Information as thing. *J. Am. Soc. Inf. Sci.* **42,** 351–360 (1991).
29. Gorichanaz, T. & Latham, K. F. Document phenomenology: a framework for holistic analysis. *J. Doc.* **72,** 1114–1133 (2016).
30. Foucault, M. *Technologies of the Self*. (University of Massachusetts Press, 1988).
31. Shortliffe, E. H. Computer programs to support clinical decision making. *JAMA* **258,** 61–6 (1987).
32. Allingham, M. *Choice Theory*. (Oxford University Press, 2002). doi:10.1093/actrade/9780192803030.001.0001
33. Klein, G. Naturalistic Decision Making. *Hum. Factors* **50,** 456–460 (2008).
34. Frederick, S., Novemsky, N., Wang, J., Dhar, R. & Nowlis, S. Opportunity Cost Neglect. *J. Consum. Res.* **36,** 553–561 (2009).
35. Tversky, A. & Kahneman, D. The Framing of Decisions and the Psychology of Choice. *Science (80-.).* **211,** 453–458 (1981).
36. Klein, G. A. *Analogical Decision Making*. (1986).
37. Lewis, D. *Papers in Metaphysics and Epistemology*. (Cambridge University Press, 1999).
38. Taylor, R. N. Age and Experience ad Determinants of Managerial Information Processing and Decision Performance. *Acad. Manag. J.* **18,** 74–81 (1975).
39. LaFond, L. Decision consistency and accuracy indices for the bifactor and testlet response theory models. *Thesis* (University of Iowa, 2014).
40. Friedman, C. P. *et al.* Enhancement of Clinicians' Diagnostic Reasoning by Computer-Based Consultation: A Multisite Study of 2 Systems. *Jama* **282,** 1851–1856 (1999).
41. Sniezek, J. A. & Buckley, T. in *Individual and Group Decision Making Current Issues* (ed. Castellan Jr., N. J.) (Lawrence Erlbaum Associates, Inc., 1993).
42. Longhurst, C. A., Harrington, R. A. & Shah, N. H. A 'green button' for using aggregate patient data at the point of care. *Health Aff.* **33,** 1229–1235 (2014).
43. Senge, R. *et al.* Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Inf. Sci. (Ny).* **255,** 16–29 (2014).
44. Cooke, R. M. Uncertainty modeling: Examples and issues. *Saf. Sci.* **26,** 49–60 (1997).
45. Putnam, H. Mathematics Without Foundations. *J. Philos.* **64,** 5–22 (1967).
46. Foulds, L. R. The Heuristic Problem-Solving Approach. *J. Oper. Res. Soc.* **34,** 927–934 (1983).
47. DiMatteo, L. A. Counterpoise of Contracts: The Reasonable Person Standard and the Subjectivity of Judgment. *S. C. Law Rev.* **48,** 293–356 (2015).

48. Sitkin, S. B. & Pablo, A. L. Reconceptualizing the Determinants of Risk Behavior. *Acad. Manag. Rev.* **17,** 9–38 (1992).
49. Woo, G. in *Risk Analysis and Natural Hazards* 221–240 (Springer International Publishing, 2016).
50. Leppanen, V. The straightforwardness of advice: Advice-giving in interactions between Swedish district nurses and patients. *Res. Lang. Soc. Interact.* 209–239 (1998). doi:10.1207/s15327973rlsi3102
51. Hudson, T. The discourse of advice giving in english: 'I wouldn't feed until spring no matter what you do'. *Lang. Commun.* **10,** 285–297 (1990).
52. DeCapua, A. & Dunham, J. F. Strategies in the discourse of advice. *J. Pragmat.* **20,** 519–531 (1993).
53. Jefferson, G. & Lee, J. R. E. The Rejection of Advice: Managing the Problematic Convergence of a 'Troubles-Telling' and a 'Service Encounter'. *J. Pragmat.* **5,** 399–422 (1981).
54. Salacuse, J. W. *The Wise Advisor: What Every Professional Should Know About Consulting and Counseling*. (Praeger, 2000).
55. Pilnick, A. 'Patient Counseling' by pharmacists: advice, information, or instruction? *Sociol. Q.* **40,** 613–622 (1999).
56. Feng, B. & MacGeorge, E. L. Predicting Receptiveness to Advice: Characteristics of the Problem, the Advice-Giver, and the Recipient. *South. Commun. J.* **71,** 67–85 (2006).
57. MacGeorge, E. L., Feng, B., Butler, G. L. & Budarz, S. K. Understanding Advice in Supportive Interactions. *Hum. Commun. Res.* **30,** 42–70 (2004).
58. Burleson, B. R. & Feng, B. The Effects of Argument Explicitness on Responses to Advice in Supportive Interactions. *Communic. Res.* **35,** 849–874 (2008).
59. Rousseau, D., Sitkin, S., Burt, R. & Camerer, C. Not so different after all: A cross-discipline view of trust. *Acad. Manag. Rev.* **23,** 393–404 (1998).
60. Muir, B. M. Trust between humans and machines, and the design of decision aids. *Int. J. Man. Mach. Stud.* **27,** 527–539 (1987).
61. Yaniv, I. Receiving other people's advice: Influence and benefit. *Organ. Behav. Hum. Decis. Process.* **93,** 1–13 (2004).
62. Calvert, R. L. The value of biased information: A rational choice model of political advice. *J. Polit.* **47,** 530–555 (1985).
63. Myers, M. When biased advice is a good thing: Information and foreign policy decision making. *Int. Interact.* **24,** 379–403 (1998).
64. Meyriat, J. Document, documentation, documentologie. l'écrit et le document. *Schéma et schématisation* **14,** 51–63 (1981).
65. Lund, N. W. Document Theory. *Annu. Rev. Inf. Sci. Technol.* **43,** 399–432 (2009).
66. Couzinet, V. A Documentologic Approach of Herbarium : Documentary Anabiosis and Philogenetic Classification A Documentologic Approach of Herbarium : Documentary Anabiosis and. *Proc. from Annu. Meet. Doc. Acad.* **2,** (2015).
67. Weaver, W. Recent Contributions to The Mathematical Theory of Communication. *Math. theory Commun.* (1949).
68. Austin, J. L. *How to Do Things with Words.* (Harvard University Press, 1962).
69. Arp, R., Smith, B. & Spear, A. D. *Building ontologies with basic formal ontology.* (MIT Press, 2015).

70. Gorichanaz, T. For Every Document, a Person: A Co-Created View of Documents. *Proc. from Doc. Acad.* **2,** Article 9 (2015).
71. Michie, D. in *Machine Intelligence 8* (eds. Michie, D. & Elcock, E. W.) 151–168 (Ellis Horwood Limited, 1977).
72. Chung, T. H. Approximate Methods for Sequential Decision Making Using Expert Advice. (1994).
73. Dennett, D. C. in *Foundations of Cognitive Psychology* 35–54 (MIT Press, 2002).
74. Smith, B. C. in *Age of Significance* 1–45 (Brian Cantwell Smith, 2010).
75. Talbott, S. *Devices of the Soul: Battling for Our Selves in an Age of Machines.* (O'Reilly Media, Inc., 2007).
76. Munro, R. in *The Sage Handbook of Power* 125–139 (Sage Publications, Inc, 2009).
77. Adriaans, P. *Learning and the Cooperative Computational Universe*. *Handbook of the Philosophy of Science* **8,** (Elsevier B.V., 2008).
78. Moher, D. *et al.* Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Med.* **6,** (2009).
79. Will, R. P., Hardaway, D. E., Will, R. P. & Donald, E. H. Advice for Building Expert Systems : Precepts or Proverbs ? *Interfaces (Providence).* **24,** 79–81 (1994).
80. Duranti, A. & Goodwin, C. *Rethinking Context: Language as an interactive phenomenon*. *Studies in Second Language Acquisition* **16,** (1992).
81. Chatman, E. The Impoverished Life-World of Outsiders. *JASIS* **47,** 193–206 (1996).
82. Downs, A. Some Thoughts on Giving People Economic Advice. *Am. Behav. Sci.* **9,** 30–32 (1965).
83. Blankart, C. B. Towards an Economic Theory of Advice and Its Application to the Deregulation Issue. *Kykos* **34,** 95–105 (1981).
84. Harvey, C. On the art of giving economic advice: tactics, access, damage-limitation, packaging, confessed ignorance, and timing. *Publ. Adm. Dev.* **6,** 445–454 (1986).
85. Marcoul, P. *A Theory of Advice Based on Information Search Incentives*. (2003).
86. Davies, I. K. Some aspects of a theory of advice: The management of an instructional developer-client, evaluator-client, relationship. *Instr. Sci.* **3,** 351–373 (1975).
87. Swank, O. H., Letterie, W. & van Dalen, H. P. A theory of policy advice. *J. Law Econ. Organ.* **15,** 602–614 (1999).
88. Craft, J. & Halligan, J. Assessing 30 years of Westminster policy advisory system experience. *Policy Sci.* **50,** 47–62 (2017).
89. Santos, S. L. & Chess, C. Evaluating citizen advisory boards: The importance of theory and participant-based criteria and practical implications. *Risk Anal.* **23,** 269–279 (2003).
90. Purgason, L. L., Avent, J. R., Cashwell, C. S., Jordan, M. E. & Reese, R. F. Culturally Relevant Advising: Applying Relational-Cultural Theory in Counselor Education. *J. Couns. Dev.* **94,** 429–436 (2016).
91. Sniezek, J. A. & Buckley, T. Cueing and cognitive conflict in judge-advisor decision making. *Organizational Behavior and Human Decision Processes* **62,** 159–174 (1995).
92. Dalal, R. S. & Bonaccio, S. What types of advice do decision-makers prefer? *Organ. Behav. Hum. Decis. Process.* **112,** 11–23 (2010).
93. Sniezek, J. a. & Van Swol, L. M. Trust, Confidence, and Expertise in a Judge-Advisor System. *Organ. Behav. Hum. Decis. Process.* **84,** 288–307 (2001).
94. Hütter, M. & Ache, F. Seeking advice: A sampling approach to advice taking. *Judgm. Decis. Mak.* **11,** 401–415 (2016).

95. Sniezek, J. A., Schrah, G. E. & Dalal, R. S. Improving judgment with prepaid expert advice. *J. Behav. Decis. Mak.* **17,** 173–190 (2004).
96. Van Swol, L. M. & Ludutsky, C. L. Tell Me Something I Don't Know: Decision Makers' Preference for Advisors With Unshared Information. *Communic. Res.* **34,** 297–312 (2007).
97. Feng, B. Testing an Integrated Model of Advice Giving in Supportive Interactions. *Hum. Commun. Res.* **35,** 115–129 (2009).
98. MacGeorge, E. L., Guntzviller, L. M., Hanasono, L. K. & Feng, B. Testing Advice Response Theory in Interactions With Friends. *Communic. Res.* **43,** 211–231 (2016).
99. Bo Feng & MacGeorge, E. L. The Influences of Message and Source Factors on Advice Outcomes. *Communic. Res.* **37,** 553–575 (2010).
100. Danziger, S., Montal, R. & Barkan, R. Idealistic advice and pragmatic choice: A psychological distance account. *J. Pers. Soc. Psychol.* **102,** 1105–1117 (2012).
101. Xiao, W. *et al.* Moral hypocrisy on the basis of construal level: To be a utilitarian personal decision maker or to be a moral advisor? *PLoS One* **10,** 1–13 (2015).
102. Licker, P. S. Getting Advice from a Computer : Five Models and a Simulator. *Data Base* (1991).
103. Collins, E. C., Percy, E. J., Smith, E. R. & Kruschke, J. K. Integrating advice and experience: Learning and decision making with social and nonsocial cues. *J. Pers. Soc. Psychol.* **100,** 967–982 (2011).
104. Nebus, J. Building collegial information networks: A theory of advice network generation. *Acad. Manag. Rev.* **31,** 615–637 (2006).
105. Zappa, P. Toward a Multilevel Social Exchange Theory of Advice Relations in Organizations. *Acad. Manag. Proc.* **2016,** (2016).
106. Perry, R. I., Walters, C. J. & Boutillier, J. A. A framework for providing scientific advice for the management of new and developing invertebrate fisheries. *Rev. Fish Biol. Fish.* **9,** 125–150 (1999).
107. Friedman, C. P., Rubin, J. C. & Sullivan, K. J. Toward an Information Infrastructure for Global Health Improvement. *IMIA Yearb. Med. Informatics* 1–8 (2017).
108. Cesa-Bianchi, N. O. *et al.* How to Use Expert Advice. *J. Assoc. Comput. Mach.* **44,** 427–485 (1997).
109. Emek, Y., Fraigniaud, P., Korman, A. & Rosén, A. Online computation with advice. *Theor. Comput. Sci.* **412,** 2642–2656 (2011).
110. Selfridge, O. G. Pandemonium: a paradigm for learning in mechanisation of thought processes. in *Proceedings of a Symposium held at the National Physics Laboratory on 24-27 Nov. 1958, volume I., (NPL Symposium no. 10, HMSO, 1959).* 513–526 (1958).
111. Teitelman, W. PILOT: a step toward man-computer symbiosis. (MIT, 1966).
112. Minsky, M. *A framework for representing knowledge.* (1974).
113. Hankin, C., Neilson, F., Neilson, H. R. & Yang, F. in *Coordination Models and Languages* 153–168 (Springer-Verlag, 2008).
114. Terepeta, M., Nielson, H. R. & Nielson, F. in *Coordination Models and Languages* 137–151 (International Federation for Information Processing, 2012).
115. Dobrev, S., Královič, R. & Pardubská, D. How Much Information about the Future Is Needed? *SOFSEM 2008 Theory Pract. Comput. Sci.* **4910,** 247–258 (2008).
116. Nass, C. & Moon, Y. Machines and Mindlessness: Social Responses to Computers. *J. Soc. Issues* **56,** 81–103 (2000).

117. Nass, C., Fogg, B. & Moon, Y. Can computers be teammates? *Int. J. Hum. Comput. Stud.* 669–678 (1996).
118. Mostow, J. A Problem-Solver for Making Advice Operational. in *AAAI-83 Proceedings* 279–283 (1983).
119. Hadden, S. G. Intelligent Advisory Systems for Managing and Disseminating Information. *Public Adm. Rev.* **46,** 572–578 (1986).
120. Gilbert, N., Buckland, S., Frohlich, D., Jirotka, M. & Luff, P. Providing Advice Throuh Dialogue. in *Proceedings of the European Conference on Artificial Intelligence, Stockholm.* 301–307 (1990).
121. Hoch, S. J. & Schkade, D. A. A Psychological Approach to Decision Support Systems. *Manage. Sci.* **42,** 51–64 (1996).
122. Grote, G., Weik, S. & W, T. Criteria for the complementary allocation of functions in automated work systems and their use in simultaneous engineering projects. **16,** 367–382 (1995).
123. Bishop, M. a. & Trout, J. D. 50 Years of Successful Predictive Modeling Should Be Enough: Lessons for Philosophy of Science. *Philos. Sci.* **69,** S197–S208 (2002).
124. Aspinall, W. A route to more tractable expert advice. *Nature* **463,** 294–295 (2010).
125. Tabachnick, B. G., Fidell, L. S. & Osterlind, S. J. *Using multivariate statistics.* (Allyn and Bacon, 2001).
126. Jones, D. R. & Brown, D. The division of labor between human and computer in the presence of decision support system advice. *Decis. Support Syst.* **33,** 375–388 (2002).
127. Anstey, P. Computing advice at a distance: The 'Remote Advisory' concept. *Softw. Pract. Exp.* **16,** 1045–1052 (1986).
128. Lewandowsky, S., Mundy, M. & Tan, G. P. The dynamics of trust: comparing humans to automation. *J. Exp. Psychol. Appl.* **6,** 104–123 (2000).
129. Madhavan, P., Wiegmann, D. A. & Lacson, F. C. Automation failures on tasks easily performed by operators undermine trust in automated aids. *Hum. Factors* **48,** (2006).
130. Madhavan, P. & Wiegmann, D. A. Effects of Information Source , Pedigree , and Reliability on Operator Interaction With Decision Support Systems. *Hum. Factors* **49,** 773–785 (2007).
131. Gardner, P. H. & Berry, D. C. The Effect of Different Forms of Advice on the Control of a Simulated Complex System. *Appl. Cogn. Psychol.* **9,** (1995).
132. Meltzoff, A. N., Kuhl, P. K., Movellan, J. & Sejnowski, T. J. Foundations for a New Science of Learning. *Science (80-.).* **325,** 284–288 (2009).
133. Burkholder, L. The Halting Problem. *SIGACT News* **18,** 48 (1987).
134. Friedman, C. *et al.* Toward a science of learning systems: a research agenda for the high-functioning Learning Health System. *J. Am. Med. Informatics Assoc.* 1–6 (2014). doi:10.1136/amiajnl-2014-002977
135. Etheredge, L. M. A rapid-learning health system. *Health Aff.* **26,** (2007).
136. Sandvig, C., Hamilton, K., Karahalios, K. & Langbort, C. When the Algorithm Itself is a Racist: Diagnosing Ethical Harm in the Basic Components of Software. *Int. J. Commun.* **10,** 19 (2016).
137. Lynch, C. Stewardship in the 'Age of Algorithms'. *First Monday* **22,** 1–18 (2017).
138. Berners-lee, T., Hendler, J., Lassila, O. & Others. The semantic web. *Sci. Am.* **284,** 34–43 (2001).

139. Gluth, M. B., Baratz, K. H., Matteson, E. L. & Driscoll, C. L. Cogan syndrome: a retrospective review of 60 patients throughout a half century. *Mayo Clin. Proc.* **81,** 483–488 (2006).

CHAPTER II

MedMinify: Study of a Promising but Stand-alone Advice-giving System

INTRODUCTION

The previous chapter explored *advice* as a concept. This chapter builds on the concept of advice while narrowing our focus to information-and-communication-technology (ICT)-intermediated advice-giving. ICT-intermediated advice-giving means producing advice using advice-giving systems, advice-giving services, and decision support systems. Together, these similar terms delineate a broad class of information systems that generate advice and communicate it primarily to people.

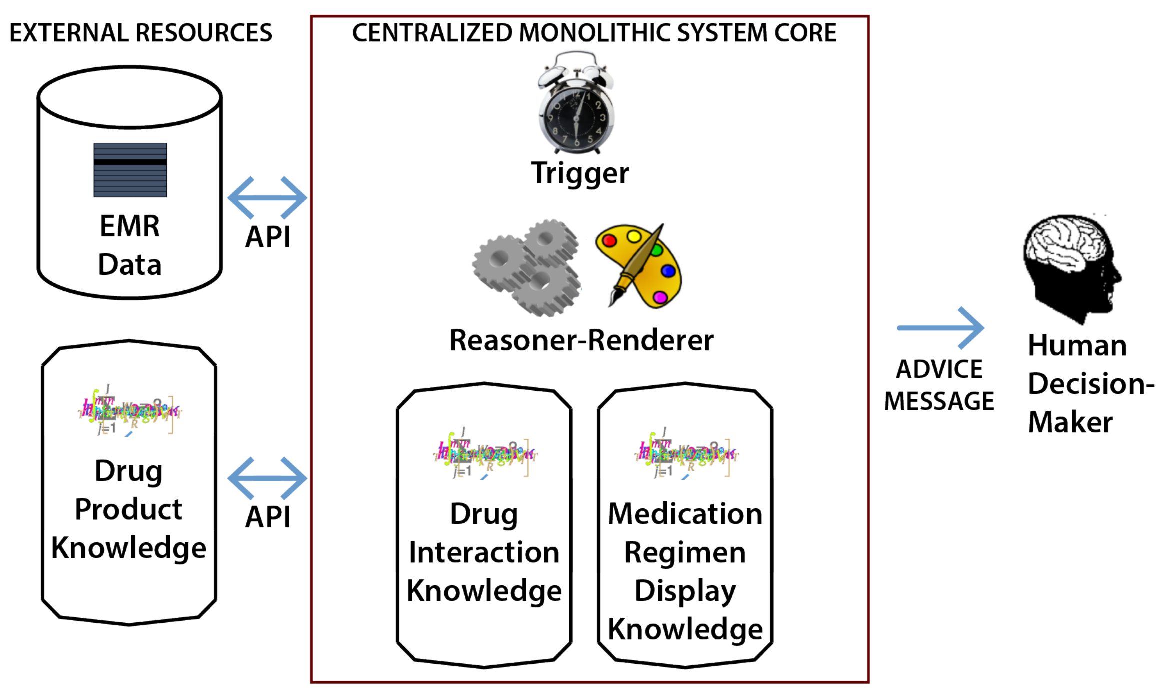
This chapter is the first of a trilogy of chapters which together relate the core scientific argument of this dissertation. In these chapters I argue that it is feasible to devise a domain-agnostic technical knowledge infrastructure capable of increasing the interoperability of computable knowledge and, consequently, of the advice-giving systems and services that use it.

The trilogy comprised by this chapter, along with Chapters III and IV, unfolds in the following way. To begin, this chapter does two things. It describes a potentially useful advice-giving system called MedMinify and it also critiques the technical architecture of that system. Chapter III reports progress on a potentially better architecture for technical knowledge infrastructure to support advice-giving systems that is realized as the Knowledge Grid. Finally, Chapter IV describes a potentially useful technical data-to-advice pipeline called ScriptNumerate that makes use of the Knowledge Grid. This trilogy of chapters reveals some common limitations to the interoperability of computable knowledge. Also covered are the design and development of knowledge infrastructure to overcome those limitations. Finally, this trilogy of chapters relates a demonstration project showing that it is feasible to devise a domain-agnostic technical knowledge infrastructure capable of increasing the interoperability of computable knowledge and, consequently, of the stand-alone advice-giving systems and services that use it.

Stand-alone advice-giving systems are a subclass of advice-giving systems. Stand-alone advice-giving systems are monolithic, vertically integrated systems without independently functioning and separately accessible subcomponents. Put another way, *stand-alone advice-giving* systems have a centralized, and not a distributed, system architecture.[1] *Stand-alone advice-giving systems* are not built using the highly modularized, system-of-systems approach described in the Introduction. Instead, *stand-alone advice-giving systems* are built in custom ways that make them idiosyncratic and inflexible. MedMinify is such a stand-alone system.

MedMinify was not developed using a common knowledge infrastructure for all of its computable biomedical knowledge. Its general architecture is portrayed in Figure 2.1 below

Figure 2.1 General Stand-alone System Architecture of MedMinify



MedMinify's architecture includes two external resources (Figure 2.1). One of these is a source of prescription data and the other is a source of drug product knowledge. Both of these sources are engaged via different Application Programming Interfaces (APIs). The rest of MedMinify is concentrated in a centralized monolithic system core. The four items in its core are inseparable and cannot stand alone. These are its Trigger function, combined Reasoner-Renderer function, and two additional knowledge resources. These four items, comprising MedMinify's core, are implemented and integrated in idiosyncratic ways. In particular, MedMinify's Drug Interaction Knowledge and Medication Regimen Display Knowledge are implemented as "one-offs" that function only within MedMinify. It is the vertically integrated, multi-function centralized monolithic system core that makes MedMinify a stand-alone advice-giving system.

In hindsight, even though the promise of MedMinify as an advice-giving system is demonstrated here, because most of its system components have a relatively low degree of interoperability, it is difficult to use and maintain MedMinify as is.

This chapter presents promising results from evaluating the *advice* from MedMinify specifically in light of the limited interoperability of its *computable knowledge*. It puts into relief the need for, and potential opportunities to attain, more highly interoperable computable knowledge, and consequently more highly interoperable knowledge services, advice-giving systems and advice-giving services. To support an analysis of interoperability, I provide a categorical model of interoperability in this chapter which ranges over five categories from low to high.

As portrayed in Figure 2.1, MedMinify's computable knowledge takes three forms. Some of its computable knowledge is drug-drug interaction knowledge. Some of it is knowledge about which drug products are available to be prescribed. The rest of it is knowledge of how to render or display a medication regimen in a manner that is known to be easy for people to read and understand.[2,3] MedMinify implements these three forms of computable knowledge in three different ways, two of which are tightly bound to its monolithic system core. It turns out that in all three cases, however, methods exist to potentially increase the interoperability of the computable biomedical knowledge that MedMinify uses.

BACKGROUND ABOUT MEDMINIFY

Dual purposes of MedMinify

To minify is to reduce the amount of something[4]. The advice-giving system *MedMinify* is a prototype originally built for the purpose of improving patient adherence to prescribed medication regimens at home. MedMinify was evaluated in a study published in 2014 [5]. The results of that study include an review of MedMinify's design and technical architecture (see Figure 2.2 below).

The 2014 study of MedMinify investigated only its *potential* to generate and communicate advice to help minify the number of times individuals have to take medications each day and the quantity of pills they have to ingest daily[5]. While MedMinify was shown to have some potential, in hindsight, it has some far-reaching limitations too. So, although unintended, another purpose of MedMinify has emerged over time. MedMinify exposes how a stand-alone system architecture limits interoperability in several identifiable ways.

About medication adherence

To describe what MedMinify does more fully, the topic of medication adherence is discussed next. Medication adherence improves when home medication regimens are made as simple as possible for patients to execute[6]. In addition, for safety, simpler home medication-taking schedules may diminish medication mishaps[7]. However, research on medication adherence is primarily justified on the basis that better adherence resulting from simplified medication regimens improves the health and well-being of consumers. It is known that, for individuals with high blood pressure, high cholesterol, or diabetes, adherence to home medication regimens improves health[8]. Conversely, lack of adherence to home medication regimens for these chronic conditions causes considerable harm[9].

The overall cost of non-adherence to medication regimens has been estimated to be $100 billion per year in the United States[10]. Some strategies to increase medication adherence have proven modestly effective[10]. These include strategies to improve communication of what medications to take and why; strategies to support behavior change, such as self-monitoring of adherence; strategies to decrease out-of-pocket costs of medications; and, strategies to cue consumers to take their medications, such as reminders and calendars[10]. The most effective approaches combine these strategies[10]. However, in terms of effect size, the evidence indicates that greater adherence improvements result from decreasing the complexity of home medication regimens than from combining educational, behavioral, and cueing interventions[10].

One strategy to decrease the complexity of medication regimens is to standardize the times when daily medication-taking events occur. Another related strategy is to limit the number of times medications are to be taken each day to a maximum of four[11]. Yet another strategy is for health care providers to conduct medication regimen reviews intended to simplify medication regimens[12]. To facilitate all three of these strategies, the stand-alone advice-giving system MedMinify was developed, in prototype form, and its capabilities were explored.

If and when it is upgraded and deployed for real-world use, MedMinify is intended to assist people to simplify home medication regimens. It advises users as to the availability of sustained-release or fixed-dose combination drug products. It also indicates whether there are pertinent drug interactions related to the *simultaneous ingestion* of two prescribed drug products. The literature points to the potential utility of simplifying medication regimens in these ways[12,13].

Two types of advice MedMinify generates and communicates are advice about the availability of two different classes of oral drug products, sustained-release and fixed-dose combination products. A *sustained-release oral drug product* is a tablet or capsule formulated to release its active drug ingredient into the gastrointestinal tract slowly and consistently over 8 to 24 hours. A *fixed-dose combination oral drug product* is a tablet or capsule containing two or more active drug ingredients in fixed amounts[14]. An example of this is the product with the brand name Lotrel. Lotrel contains two active drug ingredients to treat high blood pressure, amlodipine and benazepril. It comes in different tablets containing these two drugs in various fixed amounts.

In a systematic review of the association between the number of daily medication-taking events and adherence, Claxton, Cramer, and Pierce found mean adherence to a medication regimen declined as *daily medication-taking events* increased. Mean adherence for one, two, three, and four *daily medication-taking events* was 79% (Standard Deviation (SD), 14%), 69% (SD, 15%), 65% (SD, 16%), and 51% (SD, 20%) respectively[13]. The mean adherence differences between one *daily medication-taking event* and either three or four *daily medication-taking events* were statistically significant. The mean adherence difference between two *daily medication-taking events* and four *daily medication-taking events* was also statistically significant[13]. In light of more recent concurring evidence, these findings indicate that decreasing an individual's number of *daily medication-taking events* results in improved adherence[15,16].

Controlled studies of a standardized home medication scheduling scheme, where patients are instructed to take their medications only at breakfast, lunch, dinner or bedtime, indicate that consumers generally do not need to take medications for most chronic conditions more than four times a day[2,3,17]. These studies suggest there is an opportunity to minify the number of daily medication-taking events.

Adherence to home medication regimens is also influenced by the number of pills taken per day, a factor that is called the *daily pill burden*. To minify a person's *daily pill burden* is a potentially helpful adherence intervention[16,18]. In a recent meta-analysis, higher daily pill burden was associated with lower medication adherence to medication regimens that treat human immunodeficiency virus infection[16]. In another study, agreement with the statement, "I am already taking too many medications", was a predictor of very low adherence amongst women being treated for osteoporosis[19].

One difficulty in minifying *daily medication-taking events* and *daily pill burden* is that consumers may actually <u>increase</u> the number of scheduled *daily medication-taking events* on their own for a variety of reasons[20]. A partial explanation for this complexification could be concern over drug interactions[20]. Consumers may assume that taking oral medications at different times of day mitigates adverse events from drug interactions. Actually, relatively few drug interactions result from their *simultaneous oral ingestion*[21]. Much more often drug interactions result from systemic effects at the sites in the body where drug metabolism, pharmacologic action, or drug excretion take place[21]. To counteract their largely false beliefs about drug interactions due to simultaneous ingestion of drugs, consumers need advice to help decide which drugs they should and should not ingest at the same time. MedMinify also provides this advice.

To improve medication adherence, prior research has examined the use of information technology to assist in reducing medication regimen complexity and polypharmacy[22]. Others have reported on the automated generation of printed home medication schedules[2]. MedMinify is designed to use information technology differently to improve adherence. It gives advice on how to minify *daily medication-taking events* and *daily pill burden* while ensuring that the advice given is in concordance with what is known about the relatively few serious drug interactions that result from the simultaneous ingestion of two drugs.

SIGNIFICANCE

The research on MedMinify presented in this chapter is significant for several reasons. While the majority of patients take multiple medications for multiple conditions, many interventional trials have focused on adherence to single medication therapies or to treatments for one condition[20]. MedMinify is intentionally designed to analyze prescriptions for all scheduled tablets and capsules (i.e., for all “oral solid dosage forms” or “pills”) that consumers must ingest orally to adhere to their actual home medication regimens.

In addition, whereas pharmacist-led medication regimen review helps to simplify medication regimens, a lack of pharmacist time to conduct home medication regimen reviews, and a lack of insurance coverage for this service, greatly limit its scope and impact[12]. An information resource that gives advice on how to minify *daily medication-taking events* and *daily pill burden* could decrease the amount of time required for pharmacists to review home medication regimens, making it more feasible to provide this service routinely. Further, the same advice-giving capabilities may also facilitate the reconciliation of a patient’s prescriptions at transitions in care, a task that is thought to be important for ensuring safe use of medications[23,24].

As stated above, in the context of this dissertation, MedMinify is significant because it serves as an example of a stand-alone advice-giving system with limited interoperability.

In part, this chapter reports on the advice that MedMinify offers for simplifying medication regimens. In the next section I share the work of the first study of MedMinify's advice generation and communication capabilities. The results cover the architecture of MedMinify and show why MedMinify is a stand-alone advice-giving system.

In addition, the results of the study are supplemented with information about the varying degrees to which MedMinify's three types of computable knowledge interoperate to help it generate and communicate its advice (Figure 2.1). Then, looking forward, the chapter ends by highlighting several opportunities to increase the interoperability of computable biomedical knowledge and, by extension, the interoperability of related automated knowledge services, advice-giving services, and advice-giving systems. These are opportunities to improve the design of technical knowledge infrastructure by increasing the interoperability of computable knowledge that are discussed further in the following chapters.

RESEARCH QUESTIONS

I addressed the following three research questions to evaluate MedMinify's potential as an advice-giving system[5]:

RQ1. What is a workable architecture for a stand-alone advice-giving system capable of generating advice to help minify *daily medication-taking events* and *daily pill burden* while accounting for drug interactions from simultaneous ingestion and, in particular, what computable knowledge resources and related knowledge services does the advice-giving system's architecture include?

RQ2. How much advice can be generated by analyzing home medication regimens with an advice-giving system capable of identifying candidate regimens for minification of *daily medication-taking events* and *daily pill burden* while accounting for drug interactions from simultaneous ingestion?

RQ3. To what extent would the requirements of home medication regimens change if advice provided by MedMinify on how to minify *daily medication-taking events* and *daily pill burden* were implemented?

METHODS

MedMinify design and development

I designed and developed MedMinify in early 2014 to meet the following requirements, which are expressed as capabilities to do the following ten things:

• Access prescription data organized as individual medication regimens
• Analyze and summarize an individual's medication regimen data
• Identify all of the active ingredients in all current prescriptions in a regimen
• Check for potential substitute drug products containing the same active ingredients
• Check for drug interactions related specifically to simultaneous oral ingestion
• Generate and communicate advice for minifying daily medication taking events
• Generate and communicate advice for minifying daily pill burden
• Generate and communicate advice about drug interactions
• Render and display an evidence-based, easy-to-understand visual of a patient's regimen

Software design and application development of MedMinify involved the Substitutable Medical Applications and Reusable Technologies or "S.M.A.R.T." reference Electronic Medical Record (EMR) platform[25] (Version 1, smarthealthit.org), the Django web framework for Python (Version 1.5, djangoproject.com), and the RxNorm Application Programming Interfaces[26,27] (APIs), which are described in more detail at: rxnav.nlm.nih.gov/APIsOverview.html.

To give MedMinify's the capability to lookup active ingredients of drug products and to find potential substitute prescribable drug products, the RxNorm Application Programming Interface was used via the WWW. The automated knowledge services provided by RxNorm's API are external to MedMinify and were shared with other systems. APIs are one method of separating tightly integrated computer system components, thereby increasing their interoperability and making data and computer logic more accessible.[28]

To support the drug interaction identification capability of MedMinify, a knowledgebase containing partial drug interaction monographs was used (First DataBank, Inc. (FDB), San Francisco, CA). This company, FDB, provided us with a *one-time snapshot* from their proprietary drug knowledgebase. This one-time snapshot came in the form of a table of select fields copied from FDB's complete set of highest severity drug interaction monographs. This knowledge is embedded in MedMinify's centralized monolithic system core (Figure 2.1).

The selected three fields in the table provided by FDB were the drug interaction monograph *title field*, drug interaction *mechanism field*, and the *clinical effects field*. Keyword screening of this FDB knowledge resource was conducted to select only those highest severity drug interactions related specifically to *simultaneous oral ingestion of medications*. The 34 keywords used for this screening included "gut", "intestine", "bioavailability", "absorption", "gastric", and "pH". A list of 59 unique, high-severity *drug interaction mechanisms* associated with simultaneous oral ingestion of drugs from two or more chemical classes was created as part of this process of determining what are relevant drug-drug interactions for MedMinify.

After expanding the drug interaction mechanisms list, which is based on chemical classes of drugs, to include all specific drug-drug pairs associated with each mechanism, and after removing duplicates, a list of 559 interacting, active drug ingredient pairs associated with drug interactions due only to simultaneous ingestion of pills resulted. This list of 559 interacting drug pairs became one of the three sources of computable knowledge resources used by MedMinify. An example row from this drug interaction knowledge resource includes the antacid omeprazole (RxNorm ingredient number = 7646) and the antiviral drug atazanavir (RxNorm ingredient number = 343047). Regarding the mechanism of this drug-drug interaction, it is known that omeprazole lowers the acidity of the gastric juices, thereby decreasing the solubility of atazanavir. In turn, this disrupts atazanavir absorption and diminishes its potential efficacy.

Otherwise, to support the rest of MedMinify's capabilities listed above, a copy of the SMART Reference EMR was deployed on a local server at the University of Michigan. In addition, the code for MedMinify, which is a web application, was developed and tested using the Python computer language, the Django web framework and an Internet Browser (Apple Safari, Version 6.1.1). Finally, the drug interaction knowledgebase table from FDB was loaded into a SQL database using SQLiteStudio (Version 2.1.5 by Pawel Salawa), making this knowledge directly accessible to MedMinify.

Medication regimen data source and sample

To evaluate MedMinify, de-identified prescription data were used from the University of Michigan's electronic medical record (EMR) system (EpicCare Ambulatory EMR, Epic, Verona, WI). The University of Michigan's Institutional Review Board reviewed and approved this study. Up to date medication regimens, provided as records of electronic prescriptions for 41,903 unique individuals, were selected via a query with the following criteria, a-c:

a. Each individual whose medication regimen was selected was alive and of a chronological age greater than or equal to 60 years on the date of the query, January 15, 2014.

b. To ensure regimen currency, each individual whose medication regimen was selected had at least one medication list update documented in the University of Michigan's EMR during calendar year 2013 or the first two weeks of 2014.

c. Each individual's University of Michigan EMR profile had an International Classification of Diseases (ICD version 9) coded diagnosis for hypertension (401.x, 405.x, 416.x, 459.3x) and/or diabetes (249.x, 250.x, 253.5) and/or hypercholesterolemia (272.x).

These three inclusion criteria were chosen to provide actual, real-world medication regimen data for adults likely to be taking multiple medications to treat common chronic diseases. This query of the University of Michigan's EMR resulted in data for 380,932 prescriptions for 41,903 different people. Besides patient identifiers such as age and gender, these data included prescription fields for ordering date, prescription start date, prescription end date, generic drug name, drug product strength, frequency, medication dosage form, route of administration, quantity to dispense, number of refills, medication therapeutic subclass, prescriber instructions, and RxNorm Semantic Clinical Drug (SCD) code. Not all of these data were usable. A data exclusion diagram for the study prescriptions is shown below in Figure 2.2.

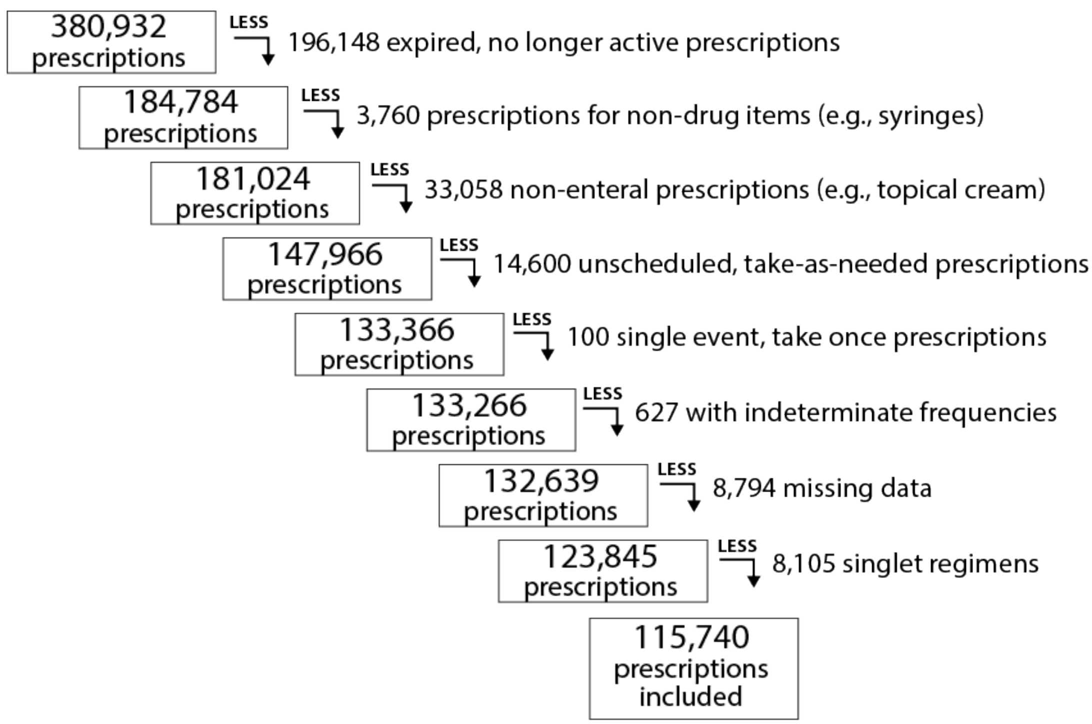


Figure 2.2 Prescription Data Exclusion Diagram for MedMinify Evaluation Study

After exclusions were applied and prescriptions that were missing data were removed, 123,845 active prescriptions remained for 35,042 people. These prescriptions were all prescriptions for scheduled, recurring home medications with known product strengths prescribed to be taken orally or sublingually. Next, because MedMinify's purpose is to simplify multi-prescription regimens, another 8,105 medication regimens consisting of *only one* medication were further excluded. This finally left 115,740 prescriptions, comprising the home medication regimens of 26,937 individuals, each with more than one scheduled and recurring oral medication (Figure 2.2). These included prescription data for this study did have medications prescribed for use on a weekly (e.g., "three times a week") or monthly (e.g., "once a month") schedule.

For this study of MedMinify's advice-giving performance, the 26,937 medication regimens included in the final data set represent the population of qualified multi-drug medication regimens used to treat chronic hypertension, hypercholesterolemia and diabetes. The 26,937 regimens include regimens for 14,403 females (53.5%) and 12,534 males (46.5%). The 26,937 individuals included had an average age of 71 years (SD 8.3, Range 60 to 103).

Several steps were completed to convert this available real-world medication regimen prescription data to the custom formats used by the SMART Reference EMR. These data conversions were made using rules in a spreadsheet. For example, the quantity of pills to be administered was not a discrete field in the available prescription data but the SMART Reference EMR data model included a discrete value for this field. So, the quantity of pills to take at each medication-taking event was extracted from the text instructions field automatically. As a case in point, the actual prescription instruction "Take 2 tablets daily" resulted in an extracted numeric quantity of pills to take value of "2" for this prescription.

Because each medication regimen in the study data set required approximately 10 seconds of time to analyze with the MedMinify prototype advice-giving system, and 30 seconds more to assess whenever advice was generated, instead of analyzing all 26,937 regimens, we used a random sample of 1,500 regimens consisting of 6,241 prescriptions automatically drawn from the population of 26,937 qualified regimens using SPSS (version 21).

For this study, random sampling was preferred over a selective sample of the most complex regimens because random sampling offered the more conservative test of MedMinify's capabilities. Using a sample of only complex home medication regimens could enhance estimates of the impact of MedMinify.

The sample size of 1,500 regimens was chosen to give a desired margin of error of 0.025 for 95% confidence intervals for Chi-squared tests with one degree of freedom[29]. Sampling was done without replacement resulting in a hypergeometric distribution. However, because the population of 26,937 is much larger than the sample size of 1,500, it was assumed that the binomial distribution provided a reasonably good approximation for calculating proportion confidence intervals. Therefore, the Agresti-Coull adjusted Wald interval for large sample sizes was used to calculate the confidence intervals for sample proportions reported in this chapter[30].

Generation of advice to minify daily medication-taking events or daily pill burden

1,500 randomly sampled regimens were analyzed using MedMinify. For each regimen, MedMinify calculated the number of prescriptions in the regimen, the *daily pill burden*, the number of unique pills in the regimen, and the maximum number of *daily medication-taking events* required by the regimen.

Next, for every immediate-release pill taken more than once a day, MedMinify engaged an RxNorm API to see if a sustained-release product was available as a potential substitute. If so, MedMinify recommended that the sustained-release product or products be considered as substitutes. Subsequently, for every pair of active ingredients prescribed as two single ingredient products, MedMinify checked RxNorm to see whether a fixed-dose combination drug product was available that combined two active ingredients in one pill. If so, MedMinify recommended that the fixed-dose combination product or products be considered as substitutes.

Continuing its regimen-specific analysis, MedMinify next screened every pair-wise combination of active drug ingredients in each regimen against the table of 559 known interacting drug pairs where the drug interaction is associated mechanistically with simultaneous ingestion. When a potential drug interaction like this was identified, MedMinify gave advice not to take the two interacting drugs at the same time of day.

Finally, MedMinify also counted all instances when a single active ingredient in a regimen appeared in the drug interaction knowledge table (i.e., one drug of an interacting pair) even where no drug interaction was found.

This study was a lab function study[31]. Its scope included only the processing of real-world medication regimen data using MedMinify to generate and communicate advice. Thus, there were no clinical or patient end users of MedMinify for this study, and none of the advice generated and communicated by MedMinify was ever shared beyond the study team.

Quantification of how much home medication regimen requirements would change if the advice from MedMinify were implemented

All instances of advice generated and communicated by MedMinify about changing the sampled medication regimens were documented, counted, and considered in the following manner that approximated the consideration this advice would receive in pharmacy practice.

Assuming MedMinify's advice was accepted and heeded, the potential change or changes to each medication regimen where advice was generated and communicated were considered in two ways. First, if a sustained-release drug product or products was recommended, I used my background as a pharmacist to check manually to see whether the maximum number of medication-taking events per day could be reduced, and, if so, by how many daily events. (This additional step is required because the opportunity to minify the maximum number of medication-taking events per day depends on whether the recommendation to substitute a sustained-release product applies to the prescription with the most daily medication-taking events, and whether there is only one prescription with the highest number of daily medication-taking events or multiple. This step was not automated although it potentially could be.)

Second, if either a sustained-release or a fixed-dose combination drug product, or both, were pointed out by MedMinify, another manual check was done to see whether the daily pill burden could be reduced, and if so by how much. When overlapping advice arose, for example when the same active ingredient could be taken via a sustained-release drug product or as a component of a fixed-dose combination drug product, preference was given to minifying the number of *daily medication-taking events* over minifying the *daily pill burden*. This preference assumes, based on pharmacy practice experience, that to minify the number of *daily medication-taking events* brings a slightly greater benefit to patients than to minify *daily pill burden*.

RESULTS

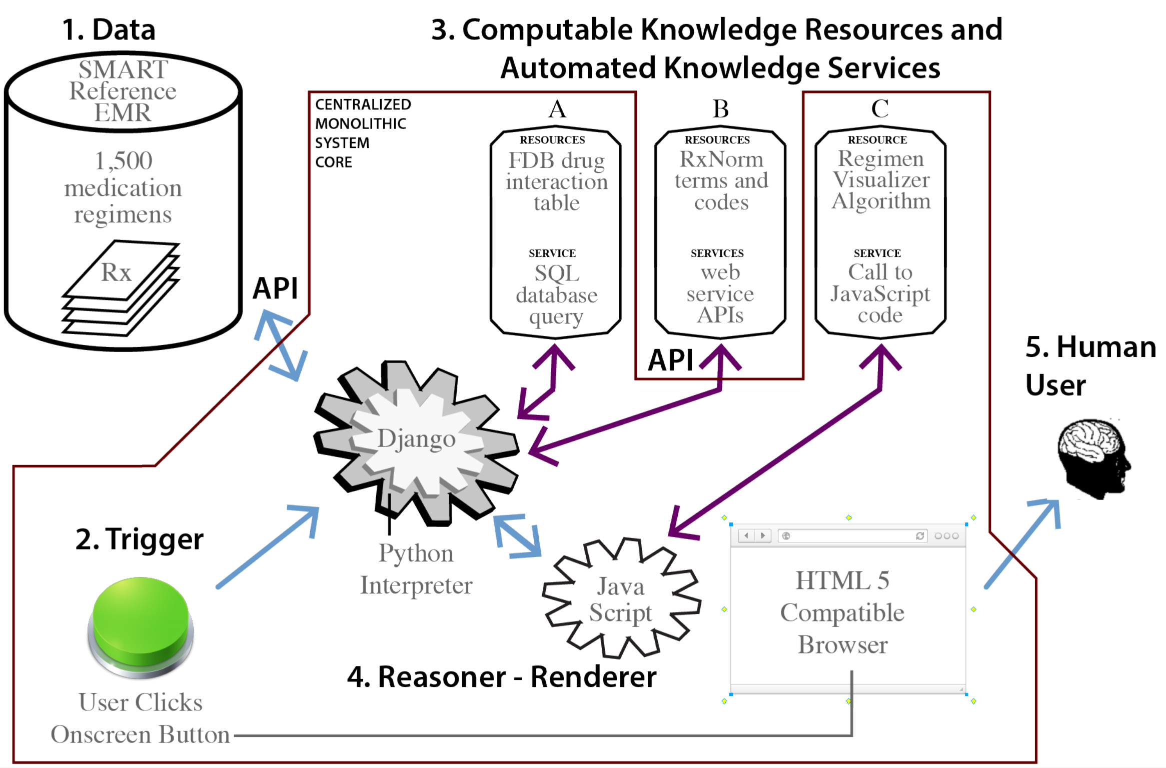
This section reports results in the order of the research questions posed for the study of MedMinify. First, results that address *RQ1* are presented. They describe the system architecture of MedMinify. Adding to these results, a supplemental comparison of MedMinify's three computable knowledge subparts is reviewed. Next, results addressing *RQ2*, stemming from a study of MedMinify using real-world patient home medication regimen data, are described and summarized. To complete this Results section, results are shared that address *RQ3* and provide insight about the potential impact of the advice generated and communicated by MedMinify.

***RQ1*. What is a workable architecture for a stand-alone advice-giving system capable of generating advice to help minify daily medication-taking events and daily pill burden while accounting for drug interactions from simultaneous ingestion and, in particular, what computable knowledge resources and related automated knowledge services does the advice-giving system's architecture include?**

MedMinify's overall advice-giving system architecture

Using Friedman's general model of an advice-giving system, a more detailed system architecture of MedMinify is portrayed in Figure 2.3. In this figure, the five parts are numbered: 1. Data, 2. Trigger, 3. Computable Knowledge Resources and Automated Knowledge Services, 4. combined Reasoner-Renderer, and 5. Human User.

Figure 2.3 Detailed System Architecture of MedMinify



The MedMinify advice-giving system is an example of a SMART Reference EMR application (Figure 2.3)[25]. The SMART Reference EMR provides an API offering needed mechanisms to authenticate systems and execute transactions by reading data from, and writing data to, its SMART Reference EMR database.

With respect to the part marked "1. Data" in Figure 2.3, to meet MedMinify's requirement for access to prescription, the SMART Reference EMR's API permitted MedMinify access to prescription data stored in its database with one minor limitation. In 2014, the SMART Reference EMR data model could not fully represent prescriptions with specific days and times in coded fields (e.g., "Take 1 tablet at 2:00 p.m. on Tuesdays and Thursdays."). For this reason, a few prescriptions had to be represented in less detail with respect to the timing of prescribed doses upon conversion to the SMART Reference EMR data format for prescriptions. (This data model limitation no longer pertains to prescription data from SMART because the SMART Reference EMR has been upgraded to address it.)

The mandatory part of any advice-giving system marked “2. Trigger” in Figure 2.3 was implemented in a straightforward way by using an onscreen button in the MedMinify web application. When the user of MedMinify clicked on this button, the medication regimen for the current patient, already in context on screen, was analyzed and, if applicable, advice was generated to help simplify the medication regimen.

Another required part for any advice-giving system is marked “3. Computable Knowledge Resources and Automated Knowledge Services” in Figure 2.3. For MedMinify, as indicated previously, three different instances of computable knowledge resources are used. These are the subparts or part 3 marked “A”, “B”, and “C” in Figure 2.3. Independent, two-way connections between subparts “A”, “B”, and “C” and the Reasoner-Renderer (Part 4) are indicated by purple arrows. As noted previously, two of these three subparts, “A” and “C” are tightly integrated within MedMinify’s centralized monolithic system core. These three subparts each combine a computable knowledge resource with an automated knowledge service. They are described next.

To meet the requirement for integration of computable drug interaction knowledge for MedMinify, the subpart denoted with an “A” in Figure 2.3, which includes a custom SQL database, was added to MedMinify. The Django web framework made it possible to easily add this integrated SQL database to MedMinify. A table was created in this embedded SQL database to organize and hold the 559 two-drug pairs that have been associated with drug interactions arising from their simultaneous oral ingestion. This drug interaction knowledge is applied by MedMinify to generate and communicate advice about interactions related to the simultaneous oral ingestion of drugs. To apply it as an automated knowledge service, a SQL database query is executed for each medication regimen MedMinify processes as instance data (Figure 2.3).

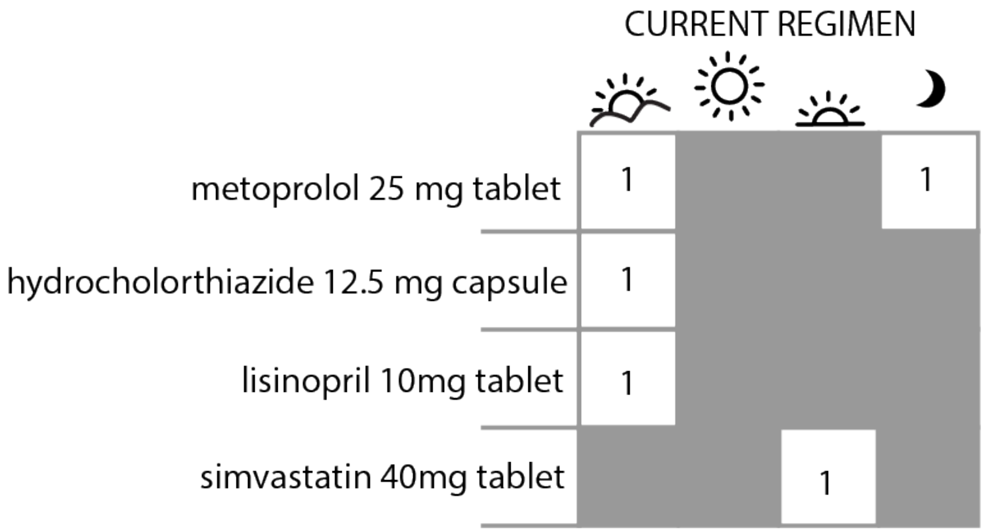
The subpart of MedMinify involving computable knowledge and related knowledge services from RxNorm is denoted with a “B” in Figure 2.3. RxNorm has an API that provides a knowledge service which is used by MedMinify to check for the availability of sustained release and fixed-dose combination drug products. To find and access this drug product availability information, procedural code within MedMinify was written in the Python programming language. This code manages calls and responses to and from RxNorm's knowledge service API. RxNorm drug product knowledge, accessed via its APIs over the WWW, is applied by MedMinify to generate and communicate MedMinify's drug product advice.

Within MedMinify's architecture, a third subpart comprised of a computable knowledge resource and a related service is denoted with a "C" in Figure 2.3. This subpart is an instance of an algorithm encoded to render the patient's current medication regimen to the screen using the evidence-based Universal Medication Schedule (UMS) image design shown below in Figure 2.4[2,3]. For that reason, this computable knowledge resource is called the Regimen Visualizer Algorithm (RVA). An instance of this algorithm is engaged when MedMinify places an internal call to a JavaScript module that is executed by an HTML 5 Compatible Browser and which results in a computer-generated visualization of a medication regimen.

The design of the UMS medication regimen visualization rendered by the RVA has four columns, one each for morning, noontime, afternoon and evening. It also has rows for each prescribed oral solid medication. An example of this UMS visual is portrayed in Figure 2.4 immediately below. Evidence suggests that this design is easier for people to understand than other ways of presenting medication regimen data[2,3].

Returning to the overview of the technical architecture of MedMinify, the fourth required part of any advice-giving system is marked "4. Reasoner-Renderer" in Figure 2.3 further above. In the case of MedMinify, this part integrates the reasoning function and the rendering function so tightly that they cannot be separated from one another. The Reasoner-Renderer includes a Python interpreter capable of combining software code written in Python, Version 2, with data to compute advice messages. The Reasoner-Renderer also includes a Django-based webserver for serving webpage content to a browser. In addition, the Reasoner-Renderer includes a JavaScript component capable of shuttling data to and from the screen of an HTML 5 Compatible Browser. This component is also capable of feeding the browser with logic for rendering images dynamically. It is tightly integrated components like these that make MedMinify a stand-alone advice-giving system.

Figure 2.4 Example of the Universal Medication Schedule Visualization



Finally, as is the case for all ICT-mediated advice interactions supported by advice-giving systems, the human user is included as part of the system. For this reason, in Figure 2.3, the part "5. Human User" is depicted on the far right of the figure.

The combination of the parts in Figure 2.3 afforded a workable advice-giving system architecture that supported a functional prototype version of MedMinify. This architecture resulted in a stand-alone MedMinify application. Its connection to the SMART Reference EMR's API carries with it the potential to link MedMinify to other, commercial and open source EMR platforms. However, MedMinify has yet to be developed beyond its prototype stage, and it has yet to be integrated with any other EMR platform except for the SMART Reference EMR. In addition, since MedMinify was developed in 2014, the SMART Reference EMR has been completely re-engineered and its unique medical data model replaced with models for Health Level Seven (HL7) Fast Health Interoperability Resources (FHIR)[33]. This move to FHIR by the SMART EMR team is a step towards better interoperability for health data.

Comparison of the interoperability of MedMinify's computable knowledge resources and related automated knowledge services

RQ1 focuses attention on MedMinify's system architecture. Additional attention is given here to the interoperability of its computable knowledge components. The columns in Table 2.1 give information about these components, which are subparts "A", "B", and "C" in Figure 2.3. The table shows the varying degrees to which MedMinify's computable knowledge resources and services are capable of interoperating with other advice-giving systems besides MedMinify.

Table 2.1 Comparison of MedMinify's Three Computable Knowledge Subparts

	Subpart A	**Subpart B**	**Subpart C**
CONTENT	Drug-drug Interaction Knowledge	Prescribable Drug Product Knowledge	Regimen Visualization Knowledge
Computable Knowledge Resources			
KNOWLEDGE SOURCE	First Databank (FDB)	RxNorm (NLM)	Scientific Literature[2,3]
ORIGINAL FORMAT	Microsoft Excel Spreadsheet	Rich Release Format (RRF) File	Written Publication
COMPUTABLE FORM	SQL Table	?	JavaScript Algorithm Implementation
NEED FOR ROUTINE UPDATES	High	High	Low
DEGREE TO WHICH RESOURCE IS INTEOPERABLE	Modestly	Highly	Slightly
Automated Knowledge Services			
TECHNICAL FUNCTION	Capability to execute custom SQL Query	Provision of shared API webservice	Capability to execute custom JavaScript code
DEGREE TO WHICH SERVICE IS INTEROPERABLE	Modestly	Highly	Slightly
Potential Advice Enabled			
POTENTIAL ADVICE MESSAGE CONTENT	Avoid Taking Drug Product A and Drug Product B at the Same Time of Day	Consider Taking a Sustained Release Drug Product ~ OR ~ Consider Taking Fixed-Dose Combination Drug Product	*See Figure 2.4*

Computable knowledge resources and related knowledge services can exhibit varying degrees of interoperability. For the purposes of this dissertation, these varying degrees of interoperability are described in very coarse terms using a five-level scale. The five terms used here to express the varying degrees of interoperability are: Not, Slightly, Modestly, Highly, and Perfectly. These terms are defined in the following ways:

Not:	Two things cannot, under any circumstances, be made to jointly and suitably perform a desired set of functions
Slightly:	Given two things, to make them jointly and suitably perform a desired set of functions, it requires *less* work to replace or entirely remake one or both of them than to modify and upgrade one or both of them
Modestly:	Given two things, to make them jointly and suitably perform a desired set of functions, it requires *about the same amount* of work to replace or entirely remake one or both of them than to modify and upgrade one or both of them
Highly:	Given two things, to make them jointly and suitably perform a desired set of functions, it requires *more* work to replace or entirely remake one or both of them than to modify and upgrade one or both of them
Perfectly:	Two things can perform a desired set of functions suitably without any further work either to make it or keep it so

Obviously, the five categorical levels of the interoperability scale presented above can only be used to roughly estimate the degree to which two things jointly perform a desired set of one or more functions in a suitable manner. Users of this categorical scale still need to determine many things for themselves, starting with what are the desired functions to be performed and what does it mean for two things to jointly perform them in a suitable way. Yet, the five-level scale above is sufficient to compare a variety of cases with respect to interoperability.

The five-level interoperability scale above enables us to distinguish, in a coarse way, cases where two things – especially two computer system components – are highly interoperable from other cases where they are not. The underlying logic of this interoperability scale relates to work. When interoperability is slight, the work of replacing or entirely remaking things to increase interoperability is justified. Where interoperability is high, only the work of modifying or upgrading things to increase interoperability further is justified.

Furthermore, the five-level interoperability scale above successfully reflects the general principle, outlined in the Introduction to this dissertation, that interoperability is a matter of degrees. The scale places great emphasis on the work of augmenting interoperability. If two things can be jointly used to perform a set of functions, then the scale clarifies that the degree to which they jointly function can be measured in terms of the work required to make it so.

With this five-level categorical interoperability scale in mind, some comparisons about MedMinify's three computable knowledge subparts, "A", "B", and "C" can be drawn by referencing the information in Table 2.1.

Subpart "A" includes a knowledge resource from FDB, which is a one-time snapshot in the form of an Excel spreadsheet. This knowledge resource was converted into a one-of-kind-table in a SQL database, making it a computable resource. Its related knowledge service executes a custom query against that one-of-a-kind SQL table. This computable knowledge resource and its related knowledge service are both only *modestly* interoperable when used to suitably perform the functions they provide within MedMinify. If another Advice-giving System, referred to here as ANAGS, wanted to take advantage of subpart "A", either it or subpart "A", or both, would have to be significantly changed. These changes would require about as much work as starting over to provide these functions to ANAGS.

Subpart "C" includes a computable resource from published manuscripts in the form of a JavaScript software routine. Its RVA software routine provides the computable instructions needed for browsers and other systems to draw an instance of the UMS when given data describing a patient's medication regimen. The computable knowledge resource and related automated knowledge service comprising subpart "C" are both only *slightly* interoperable when used to suitably perform the functions they provide to MedMinify. Because of how tightly integrated the RVA is with the other parts of MedMinify's Reasoner-Renderer, if ANAGS wanted to take advantage of this same function, it would be easier to remake subpart "C" just for ANAGS than it would be to excise it from MedMinify. As it stands, subpart "C" is so deeply embedded within the structure of MedMinify's software code that it is a very difficult task to isolate it.

In contrast, MedMinify's computable knowledge subpart "B" includes a *highly* interoperable knowledge resource from the NLM, in the form of an RxNorm Rich Release Format (RRF) file. This RRF file has a well-defined and described standardized file structure. The NLM also provides a companion *highly* interoperable RxNorm knowledge service, in the form of a well-described, open API available via the WWW[26,27]. Interestingly, it is inscrutable to RxNorm API users how the knowledge service it provides works technically. All API users need to know to use the RxNorm API is the format of its service requests and service responses. By using the specified format, any advice-giving system like MedMinify can be made to request and receive RxNorm drug product information via the WWW.

It is of interest that an API is partly responsible for making it so that subpart "B" of MedMinify's larger computable knowledge component (Part 3) is a *highly* interoperable subpart. Why is this so? In a recent article, Plantin et al. compare the architectural and other properties of larger-scale information infrastructure, like the Internet, with those of smaller-scale private platforms[28]. In this context, APIs are defined as technical points of access to data and computer logic provided by platforms. When APIs are deployed over the WWW, as is the case for the RxNorm APIs, the architecture of the WWW, which is based on a host of open protocols and standards, helps to make the APIs even more highly interoperable.[28]

One thing this discussion has so far ignored is how easy or difficult it is to keep MedMinify's computable knowledge subparts "A", "B", and "C" updated with the most current knowledge. Because knowledge is always in flux, always being updated and refined[33], the need for routine updates is relevant to the interoperability of MedMinify and other advice-giving systems. In the case of MedMinify, subpart "A" and subpart "B" have a high need for routine updates. Drug-drug interaction and drug product knowledge change frequently. On the other hand, subpart "C" has a low need for updates. The design of the UMS, and the JavaScript code used to render it, are both very stable. The routinely provided updates to drug product knowledge provided by the NLM are another reason why subpart "B" is highly interoperable whereas subpart "A", which is based on a one-time snapshot of knowledge from a proprietary knowledgebase, is only modestly interoperable.

Next let me explain why MedMinify's subpart "B", comprised of RxNorm resources and its API, is *not* thought to be perfectly interoperable for the set of functions it provides to MedMinify. The reason for this relates back to the inscrutability of the internal workings of its API from the perspective of advice-giving system developers. As a computable knowledge subpart for MedMinify, RxNorm is controlled externally by the NLM. For this reason, with or without warning, the NLM is within their rights to unilaterally change or decommission RxNorm. Therefore, from the perspective of MedMinify and similar systems, the interoperability of its computable knowledge could be further improved by enabling an updateable copy of RxNorm to come under the control of others, including makers of advice-giving systems. Indeed, this may even be possible, but it is not the way RxNorm works. Hence, the interoperability of RxNorm is not perfect because the NLM maintains policies and procedures governing the use of its APIs that constrain their use somewhat. This example shows how interoperability is a socio-technical phenomenon. Interoperability involves policy and governance as well as the technical features of things.

Summary of the results about MedMinify's advice-giving system architecture

Summarizing the results in this section, MedMinify has a workable system architecture, which is depicted in Figure 2.3. Its architecture includes three computable knowledge subparts. Each of these subparts combines a computable knowledge resource with an automated knowledge service. The degree to which the resources and services comprising these subparts interoperate within the context of MedMinify's technical architecture varies considerably.

***RQ2*. How much advice can be generated by analyzing home medication regimens with an advice-giving system capable of identifying candidate regimens for minification of *daily medication-taking events* and *daily pill burden* while accounting for drug interactions from simultaneous ingestion?**

Descriptive statistics and results quantifying the number of sampled regimens prompting advice from MedMinify are given in Table 2.2 below. In that table, the 1,500 sampled home medication regimens are categorized according to the current or prescribed number of daily medication-taking events associated with each regimen.

Table 2.2 Results by Number of Daily Events: Sampled medication regimens for individuals of age ≥ to 60 years with diagnoses for hypertension or diabetes or hypercholesterolemia organized by the current number of daily medication-taking events per regimen.

Current Number of Daily Medication-Taking Events	Number of Sampled Regimens	Number of Sampled Regimens prompting Advice from MedMinify	Average Number of Prescriptions per Sampled Regimen	Average Daily Pill Burden per Sampled Regimen
8 EVENTS DAILY	1	0	7.0	26.0
6 EVENTS DAILY	3	2	5.3	15.6
5 EVENTS DAILY	4	2	6.3	15.8
4 EVENTS DAILY	44	22	5.4	11.3
3 EVENTS DAILY	178	126	5.7	10.4
2 EVENTS DAILY	643	388	4.6	6.7
1 EVENT DAILY	626	79	3.2	3.4
0* EVENTS DAILY	1	0	2.0	0.0
TOTALS	1,500	619		

* One regimen included only two prescriptions for medications to be taken weekly and no medications to be taken daily.

Per the results organized and presented in Table 2.2., MedMinify generated advice for 619 (41.3%) of the 1,500 randomly sampled regimens (95% Confidence Interval (CI) of 38.7% to 43.7%). These 619 regimens are regimens where either the number of *daily medication-taking events* or the *daily pill burden* could potentially be reduced.

In Table 2.3 below, the 619 regimens where at least one message of advice, or *advisory*, about a potential change to a medication regimen was generated and communicated by MedMinify are categorized by advice type. Population estimates for the proportion of regimens subject to each specific type of advice are given with 95% confidence intervals.

Table 2.3 Results by Advisory Type: Counts and percentages of sampled regimens (n = 1,500) are provided by specific advisory type with confidence intervals for population estimates of the percentages of regimens for which MedMinify would generate advice for the entire study population of 26,937 home medication regimens.

Advisories Generated per Sampled Regimen	Number of Sampled Regimens	% of Sampled Regimens with Population Estimate as (95% CI's)
ANY ADVISORY	619	41.3% (38.8, 43.8)
SUSTAINED RELEASE PRODUCT ADVISORY ONLY	407	27.1% (24.9, 29.4)
FIXED-DOSE COMBINATION PRODUCT ADVISORY ONLY	114	7.6% (6.3, 8.9)
SUSTAINED RELEASE AND FIXED-DOSE COMBINATION PRODUCT ADVISORIES	94	6.3% (5.0, 7.5)
SUSTAINED RELEASE AND DRUG INTERACTION ADVISORIES	3	0.2% (0.0, 0.4)
DRUG INTERACTION ADVISORY ONLY	1	0.1% (0.0, 0.3)

Specifically regarding the very few advisories about drug-drug interactions possibly resulting from the simultaneous ingestion of two drugs, for the 1,500 sampled home medication regimens, MedMinify identified *at least one* drug on the FDB drug interaction pairs list in 1,209 (81%) of regimens, 95% CI [78.6, 82.6]. However, MedMinify only found relevant *drug pairs*, and thus advised against taking the two drugs comprising a pair at the same time, in 4 out of 1,500 regimens (Table 2.3).

Three of the four drug interaction advice messages from MedMinify indicated a concern about ingesting cyclosporine and simvastatin at the same time. (This concern is due to competition over drug absorption in the small intestine and liver that could change the serum levels and efficacy of either drug[34]. It is not clear whether taking cyclosporine and simvastatin at different times of day mitigates this risk.) The fourth case of a drug-drug interaction related to simultaneous ingestion involved the drugs rosuvastatin and gemfibrozil. (In this case there is a risk of muscle damage when using these drugs simultaneously. The exact mechanism of this drug interaction is unknown. There is a suggestion in the literature that simultaneous ingestion may play a role in the interaction[35]. It is not clear whether taking rosuvastatin and gemfibrozil at different times of day mitigates this risk.)

***RQ3*. To what extent would the requirements of home medication regimens change if advice provided by MedMinify on how to minify *daily medication-taking events* and *daily pill burden* were implemented?**

If MedMinify's advice were accepted and implemented for all of the regimens for which advice messages were generated, 320 out of the 1,500 regimens (21.3%, 95% CI [19.3, 23.4]) would have *at least one* fewer *daily medication-taking event* (Table 2.4). In addition, three regimens would *gain* an additional *daily medication-taking event* in an attempt to mitigate a drug interaction.

Finally, the *daily pill burden* for an additional 295 of the 1,500 regimens (19.7%, 95% CI [17.7, 21.8]) would be minified by an average of 1.4 pills per regimen. Table 2.4 includes more detail about the regimen changes that would come about if MedMinify's advice messages were heeded.

Table 2.4 Impact of Advisories. Sampled medication regimens (n = 1,500) for individuals of age ≥ to 60 years with diagnoses for hypertension or diabetes or hypercholesterolemia organized by the current number of daily medication-taking events per regimen.

Current Number of Daily Medication-Taking Events	Number of Sampled Regimens	Number of Sampled Regimens with Fewer Daily Medication-Taking Events	Number of Sampled Regimens with Decreased Daily Pill Burden Only	New Average Daily Pill Burden per Sampled Regimen if Advice were Implemented (Difference)
8 EVENTS DAILY	1	0	0	26.0 (0)
6 EVENTS DAILY	3	1	1	13.3 (-2.3)
5 EVENTS DAILY	4	1	1	14.8 (-1.0)
4 EVENTS DAILY	44	10	12	10.2 (-1.1)
3 EVENTS DAILY	178	80	45	9.3 (-1.1)
2 EVENTS DAILY	643	228	157	5.8 (-0.9)
1 EVENT DAILY	626	N/A†	79	3.3 (-0.1)
0* EVENTS DAILY	1	N/A	N/A	0.0 (0)
TOTALS	1,500	320	295	

* One regimen included only two prescriptions for medications to be taken weekly and no medications to be taken daily.
† One is the least possible daily medication-taking event number. MedMinify never advises to discontinue medications.

DISCUSSION

The overall potential of the advice MedMinify generates and communicates

Taking medications every day at home to treat chronic diseases is difficult for most people. The degree to which a person adheres to a home medication schedule is a result of a complex set of influences including economic factors, social support, individual dedication to the task, and beliefs about the utility of prescribed medication treatments[6,10,36]. For individuals with means who have supportive relationships and believe that taking medications is beneficial, adherence to complex medication regimens is *still* difficult[13]. More attention on simplifying regimens is needed to make every home medication regimen as easy to execute as possible[11].

Pharmacists have long been champions of simplifying medication regimens and reducing polypharmacy[37]. However, many consumers are unaware that their pharmacist can assist them to simplify their home medication regimens. Evidence shows that relatively few consumers are assured to have the simplest possible home medication regimens[20]. Advice from MedMinify could potentially improve the efficiency of time-consuming home medication regimen reviews offered by pharmacists. These reviews by pharmacists are provided in a mostly haphazard manner to consumers today.

In this initial laboratory function study of the MedMinify advice-giving system, advice to minify daily medication-taking event and daily pill burden was generated for 41.3% of actual home medication regimens for adults with one or more of three common chronic diseases. This result is bolstered by a previous analysis we did where we found that similar drug product substitution advice could be generated for 38.4% of 2,944 home medication regimens of retirees surveyed in 2007[38], and by results from Elliott that potential changes to reduce complexity were identified in 45.7% of reviewed medication regimens at hospital discharge[12].

Based on a careful analysis of the 1,500 sampled regimens, we believe the 41.3% figure may overestimate the actual percentage of candidate home patient medication regimens subject to useful simplification advice. There are countervailing issues that would prevent changes to a home medication regimen in some cases when MedMinify provided advice. These include consumer preference, clinical need, drug product cost, and perceived low marginal utility from making a change. Yet, if only 1 out of 5 home medication regimens for which advice was generated and communicated by MedMinify were actually simplified, the percentage of home medication regimens changed would still be approximately 8% in this population.

If 8% of home medication regimens could be simplified then routine screening of medication regimens with an advice-giving system like MedMinify may be justified. This is a particularly important finding when one considers that 15.3% of home medication regimens in this population require individuals to take medications more than two times a day (Table 2.2). The results of this study indicate that 151 of 230 complex regimens (65.6%) were candidates for simplification, and that the greater fraction of the advice given to simplify these regimens would have resulted in diminishing the number of *daily medication-taking events* (Table 2.4). Because the regimens studied were real ones, and because there is other supporting evidence that medication regimens are unnecessarily complex[20], it seems reasonable to generalize these findings to similar older adult populations with high blood pressure, high cholesterol, and/or diabetes. Therefore, this study shows that advice from MedMinify has the potential to be helpful.

Reviewing the types of advice MedMinify provides

Based on the content-based typology of advice from Chapter I, MedMinify provides two different types of advice content. MedMinify's advice is designed to communicate *definitive, accurate facts* about the availability of drug products and the existence of certain drug-drug interactions or to communicate *possible* drug product alternatives from which to choose. MedMinify does not communicate medication regimen simplification advice in terms of probabilities nor does it generate advice about anything that is either unknown or unknowable.

Specific messages of advice to substitute sustained-release products for immediate-release ones were the most common recommendations made by MedMinify (Table 2.3). An example would be to use the antihypertensive beta-blocker carvedilol as a sustained-release product once a day instead of taking the drug twice a day as an immediate-release product.

MedMinify generated fixed-dose combination recommendations for a total of 208 out of the 1,500 regimens (Table 2.3). An example of a fixed-dose combination advice message would be a message recommending substitution of a pill that contains both the antihypertensive drugs hydrochlorothiazide and lisinopril, instead of taking these two drugs as two separate pills.

It was expected that very few drug interactions would be identified in real regimens because of prior drug interaction screening at the clinic or in the pharmacy. In this study, while most regimens included one drug in the drug interaction knowledge table, rarely were two interacting drugs found in the same medication regimen. For a drug interaction to be detected by MedMinify, commonly used medications, such as the “cholesterol-lowering statins”, had to have been paired with other drugs that are rarely used. By checking for the rare drug interactions that can potentially be managed by taking medications at different times of day, MedMinify can conversely confirm that most oral medications prescribed to treat high blood pressure, diabetes, and high cholesterol can safely be taken together at the same time of day.

Limitations regarding the study of MedMinify’s advice

This study of MedMinify’s advice has several limitations. With respect to the methods used in this study, the clinical appropriateness of the advice offered by MedMinify was not formally assessed. Also, prescriptions for non-oral drug products were excluded, as were prescriptions where the instructions tell the patient to “take as needed”, making the regimens studied less complex than they actually are in reality.

In addition, MedMinify’s advice-giving is limited in its scope. It does not reconcile medication-taking events with individual preferences for meal times, work hours, or sleep schedules. Also, MedMinify does not account for the differing costs of various medication products, nor does it account for various prescription insurance plans. Moreover, MedMinify does not recognize the difference between temporary or trial prescriptions and other prescriptions for medications intended to be in use for a long time. Also, by limiting the scope of drug interaction assessment to only the highest severity interactions, as defined by FDB, some less severe but important drug interactions were surely overlooked.

Interoperability challenges to be overcome related to MedMinify’s computable knowledge resources and automated knowledge services

Significant challenges to building and deploying advice-giving systems that actually improve health fall into a number of categories[39,40]. There are challenges sourcing the data needed to compute advice with these systems, challenges implementing advice-giving systems within established workflows, and challenges related to the design and upkeep of the computable knowledge resources and related automated knowledge services used by these systems. This chapter finishes by framing a series of limitations that are specific to MedMinify. These limitations relate to the interoperability of its computable knowledge.

While advice provided by MedMinify evidently has the potential to be helpful, its system architecture has some critical limitations. These limitations have to be overcome to make the computable knowledge components used by MedMinify more highly interoperable. While they are real limitations in MedMinify's case, they are also opportunities to improve the design and technical architecture of the computable knowledge components of advice-giving systems.

MedMinify stands alone as an advice-giving system primarily because it relies on two unique computable knowledge subparts. These are MedMinify's Subpart A with drug interaction knowledge and its Subpart C with regimen visualization knowledge (Table 2.1). These two computable knowledge subparts function only within the context of MedMinify's particular technical architecture.

In the case of MedMinify's drug interaction knowledge subpart, which is comprised of static computable knowledge stored in a database table and accessed by a query, two limitations need to be overcome to improve its interoperability. The first limitation is that, in its current form, this computable knowledge cannot be accessed by other systems. The second limitation is that this computable knowledge is a static isolate without a globally unique identifier. As such, it stands apart from any mechanism that could potentially keep it up to date. If these limitations are overcome, then there is the potential to for widespread use and routine updating which is associated with the virtuous learning cycle of Learning Health Systems.[41]

To overcome the first limitation, which is about access, poses a technical challenge. It is the challenge of modularizing, encapsulating, and then making MedMinify's drug interaction knowledge both accessible and serviceable to other information systems. To make computable knowledge accessible means to share it with other systems as a resource. To make computable knowledge serviceable is harder. It involves providing an automated knowledge service, such as an API, that multiple systems, and not just one, can easily draw on.

MedMinify's regimen visualization algorithm subpart is only slightly or modestly interoperable. It could also be made more accessible and interoperable by using an API.

Many other medication-use systems could potentially benefit from a capability to draw better medication regimen visuals and from a capability to identify drug interactions specifically related to simultaneous ingestion of two drugs. As one example, information systems used on nursing wards and in nursing homes could benefit from both of these capabilities. The first capability would help nurses and others communicate medication regimens to patients. The second capability could be used to offset the timing of medication administration to mitigate the effects of some drug interactions.

To overcome the second limitation of MedMinify's computable drug interaction knowledge poses two procedural challenges. These are the challenges of uniquely identifying this and other instances of computable biomedical knowledge and of developing reliable mechanisms to keep instances of computable biomedical knowledge current.

MedMinify's Regimen Visualization Algorithm (RVA) subpart is comprised of static JavaScript code. To augment its interoperability sufficiently so that it can be shared with other systems, that code needs to be modularized or encapsulated, uniquely identified, and made serviceable, perhaps by using an API. It would also help to have mechanisms to version and update it.

MedMinify exposes a series of three limitations constraining its computable knowledge resources and related automated knowledge services. The limitations in this series are (1) a lack of modularization enabling instances of computable knowledge and related services to be externalized from any system, (2) a lack of unique identifiers for instances of computable knowledge and related services, and (3) a lack of mechanisms to keep computable knowledge and related services up to date.

Summarizing, as a stand-alone advice-giving system, MedMinify puts into relief the need for more easily shareable and updatable computable knowledge resources and related automated knowledge services. It also shows that potentially useful advice to simplify medication regimens can be generated automatically. Work to overcome the series of limitations above and increase the interoperability of computable knowledge resources and related automated knowledge services so that they can be used by multiple systems follows in Chapters III and IV of this dissertation. These two upcoming chapters focus on the development and early testing of the Knowledge Grid, respectively. The Knowledge Grid includes a set of technical components intended to help overcome the technical and procedural challenges that pertain to much of the computable knowledge used by MedMinify.

CONCLUSION

MedMinify generates potentially useful advice for simplifying medication regimens. Because of its system architecture MedMinify is a stand-alone advice-giving system. It has a number of tightly integrated core components, and these components have a low degree of interoperability with any system other than MedMinify. Therefore, among other requirements not yet met by MedMinify, a better advice-giving system for simplifying home medication regimens would provide for more highly interoperable, shareable, and easily updateable computable knowledge resources and related automated knowledge services.

Improvements in the interoperability of computable knowledge are required to accelerate the flow of health knowledge, and advice based on health knowledge, throughout the health system[41,42]. Without improvements in the interoperability of the computable knowledge components of advice-giving systems like MedMinify, stand-alone advice-giving systems will continue to have only a modest impact on health practice. We know this because of the increasing number of promising but stand-alone advice-giving systems that have been developed, tested, and described in the scientific literature, but yet used only sparingly in practice.[43–45]

This dissertation now turns to a body of scientific work on increasing the interoperability of computable knowledge and related automated knowledge services and advice-giving systems and services. That body of work is intended to make advice-giving systems like MedMinify easier to develop, deploy, and use in a wide variety of situations.

References

1. Mizuno, T. & Nishizaki, S. Distributed Online Judge System for Interactive Theorem Provers. **16,** 1–5 (2014).
2. Schnipper, J. L. *et al.* Rationale and design of the Pharmacist Intervention for low literacy in Cardiovascular Disease (PILL-CVD) study. *Circ. Cardiovasc. Qual. Outcomes* **3,** 212–219 (2010).
3. Kripalani, S., Schmotzer, B. & Jacobson, T. A. Improving Medication Adherence through Graphically Enhanced Interventions in Coronary Heart Disease (IMAGE-CHD): A Randomized Controlled Trial. *J. Gen. Intern. Med.* **27,** 1609–1617 (2012).
4. Oxford English Dictionary. Oxford English Dictionary Online. *Oxford English Dictionary* (2017).
5. Flynn, A. J., Klasnja, P. & Friedman, C. P. MedMinify: An Advice-giving System for Simplifying the Schedules of Daily Home Medication Regimens Used to Treat Chronic Conditions. *AMIA Annu. Symp. Proc.* **2014,** 1728–37 (2014).
6. Osterberg, L. & Blaschke, T. Adherence to Medication. *N. Engl. J. Med.* **353,** 487–497 (2005).
7. Willson, M. N., Greer, C. L. & Weeks, D. L. Medication Regimen Complexity and Hospital Readmission for an Adverse Drug Event. *Ann. Pharmacother.* **48,** 26–32 (2014).
8. Deaton, A. The Great Escape. *Heal. Wealth, Orig. Inequal.* **2,** 376 (2013).
9. Mennini, F. S. *et al.* Cost of poor adherence to anti-hypertensive therapy in five European countries. *Eur. J. Heal. Econ.* **16,** 65–72 (2015).
10. Bosworth, H. B. *et al.* Medication adherence: A call for action. *Am. Heart J.* **162,** 412–424 (2011).
11. Hernandez, L. M. *Standardizing medication labels: Confusing patients less: Workshop summary.* (National Academies Press, 2008).
12. Elliott, R. A. Reducing medication regimen complexity for older patients prior to discharge from hospital: Feasibility and barriers. *J. Clin. Pharm. Ther.* **37,** 637–642 (2012).
13. Claxton, A. J., Cramer, J. & Pierce, C. A systematic review of the associations between dose regimens and medication compliance. *Clin. Ther.* **23,** 1296–1310 (2001).
14. Winer, N. *et al.* Effect of fixed-dose ACE-inhibitor/calcium channel blocker combination therapy vs. ACE-inhibitor monotherapy on arterial compliance in hypertensive patients with type 2 diabetes. *Prev. Cardiol.* **8,** 87–92 (2005).
15. Cohen, C. *et al.* Once-daily antiretroviral therapies for HIV infection: Consensus statement of an advisory committee of the international association of physicians in AIDS care. *J. Int. Assoc. Physicians AIDS Care* **1,** 141–145 (2002).
16. Nachega, J. B. *et al.* Lower pill burden and once-daily antiretroviral treatment regimens for HIV infection: A meta-analysis of randomized controlled trials. *Clin. Infect. Dis.* **58,** 1297–1307 (2014).
17. Gazmararian, J., Jacobson, K. L., Pan, Y., Schmotzer, B. & Kripalani, S. Effect of a pharmacy-based health literacy intervention and patient characteristics on medication refill adherence in an urban health system. *Ann. Pharmacother.* **44,** 80–87 (2010).
18. Stange, D., Levente, K., Von-Wolff, A., Baehr, M. & Dartsch, D. C. Reducing Cardiovascular Medication Complexity in a German University Hospital: Effects of a

Structured Pharmaceutical Management Intervention on Adherence. *J. Manag. Care Pharm.* **19,** (2013).
19. Solomon, D. H. *et al.* Predictors of very low adherence with medications for osteoporosis: Towards development of a clinical prediction rule. *Osteoporos. Int.* **22,** 1737–1743 (2011).
20. Wolf, M. S. *et al.* Helping patients simplify and safely use complex prescription regimens. *Arch. Intern. Med.* **171,** 300–305 (2011).
21. Hansten, P. D. & Horn, J. R. *The top 100 drug interactions.* (H&H Publishing, 2013).
22. Farrish, S. & Grando, A. Ontological approach to reduce complexity in polypharmacy. in *Annual Symposium proceedings of the AMIA Symposium* **2013,** 398–407 (2013).
23. Kwan, J. L., Lo, L., Sampson, M. & Shojania, K. G. Medication Reconciliation During Transitions of Care as a Patient Safety Strategy: A Systematic Review. *Ann. Intern. Med.* **158,** 397–404 (2013).
24. Chhabra, P. T. *et al.* Medication reconciliation during the transition to and from long-term care settings: A systematic review. *Res. Soc. Adm. Pharm.* **8,** 60–75 (2012).
25. Mandl, K. D. *et al.* The SMART Platform: Early experience enabling substitutable applications for electronic health records. *J. Am. Med. Informatics Assoc.* **19,** 597–603 (2012).
26. Nelson, S. J., Zeng, K., Kilbourne, J., Powell, T. & Moore, R. Normalized names for clinical drugs: RxNorm at 6 years. *J Am Med Inf. Assoc* **18,** 441–448 (2011).
27. Bodenreider, O. & Peters, L. RxNav : Browser and Application Programming Interfaces for RxNorm. in *AMIA 2010 Symposium Proceedings* 1330 (2010).
28. Plantin, J.-C., Lagoze, C., Edwards, P. N. & Sandvig, C. Infrastructure studies meet platform studies in the age of Google and Facebook. *New Media Soc.* 146144481666155 (2016). doi:10.1177/1461444816661553
29. Krejcie, R. V & Morgan, D. W. Determining Sample Size for Research Activities. *Educ. Psychol. Meas.* **30,** 607–610 (1970).
30. Brown, L. D., Cai, T. T. & Dasgupta, A. Interval Estimation for a Binomial Proportion. *Stat. Sci.* **16,** 101–117 (2001).
31. Friedman, C. P. & Wyatt, J. *Evaluation Methods in Biomedical Informatics*. (Springer, 2006).
32. Bender, D. & Sartipi, K. HL7 FHIR: An Agile and RESTfulApproach to Healthcare Information Exchange. 1–6 (2015).
33. 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).
34. Kalliokoski, A. & Niemi, M. Impact of OATP transporters on pharmacokinetics. *Br. J. Pharmacol.* **158,** 693–705 (2009).
35. Bergman, E. *et al.* Effect of a single gemfibrozil dose on the pharmacokinetics of rosuvastatin in bile and plasma in healthy volunteers. *J. Clin. Pharmacol.* **50,** 1039–1049 (2010).
36. Norell, S. E. Improving medication compliance: a randomised clinical trial. *Br. Med. J.* **2,** 1031–3 (1979).
37. Lenaghan, E., Holland, R. & Brooks, A. Home-based medication review in a high risk elderly population in primary care - The POLYMED randomised controlled trial. *Age Ageing* **36,** 292–297 (2007).

38. Flynn, A. J., Friedman, C. P. & Klasnja, P. Taking it easy - a needs analysis for computer-generated advice to simplify home medication regimens. in *AMIA Annual Symposium Proceedings2* (2013).
39. Bates, D. W. *et al.* Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inf. Assoc* **10,** 523–530 (2003).
40. Sittig, D. F. *et al.* Grand challenges in clinical decision support. *J. Biomed. Inform.* **41,** 387–92 (2008).
41. Friedman, C. P., Rubin, J. C. & Sullivan, K. J. Toward an Information Infrastructure for Global Health Improvement. *IMIA Yearb. Med. Informatics* 1–8 (2017).
42. Friedman, C. *et al.* Toward a science of learning systems: a research agenda for the high-functioning Learning Health System. *J. Am. Med. Inform. Assoc.* **22,** 43–50 (2015).
43. Greenes, R. A. *et al.* Clinical Decision Support Models and Frameworks: Seeking to Address Research Issues Underlying Implementation Successes and Failures. *J. Biomed. Inform.* (2017). doi:10.1016/j.jbi.2017.12.005
44. Evans, R. S. *et al.* A computer-assisted management program for antibiotics and other antiinfective agents. *N Engl J Med* **338,** 232–238 (1998).
45. Friedman, C., Elstein, A. & Wolf, F. Enhancement of Clinicians' Diagnostic Reasoning by Computer-Based Consultation A Multisite Study of 2 Systems. *Jama* **282,** (1999).

CHAPTER III

Design and Development of Initial Technical Components for the Knowledge Grid

INTRODUCTION

My goal in this dissertation is show how to increase the interoperability of computable biomedical knowledge sufficiently to potentiate the production of well-informed health advice. Well-informed advice reflects current best practices, which are established primarily through the results of empirical science.

Chapter I is the cornerstone of this dissertation. Its conceptual analysis indicates that advice messages are information objects targeted at unmade decisions. In this chapter, we explore new methods to generate and communicate large quantities of well-informed advice.

This chapter is the middle chapter in a trilogy of chapters that complete the principal scientific argument of this dissertation. That argument asserts the feasibility of building a domain-agnostic technical knowledge infrastructure capable of significantly augmenting the degree of interoperability of computable knowledge.

Chapter II set the baseline for improvement. Using the example of the MedMinify advice-giving system, Chapter II exposed a series of limitations to the interoperability of MedMinify's computable knowledge resources and related automated knowledge services. These limitations were a lack of resource and service modularity, a lack of unique identifiers for its resources and services, and a lack of mechanisms to keep its resources and services up to date. These are serious limitations. They explain why MedMinify stands alone due to proprietary elements of its technical architecture. These limitations hinder MedMinify from being easily deployed in real-world settings.

This chapter covers the work of designing and developing new technical means to augment the interoperability of computable biomedical knowledge and related services. This work attempts to overcome the limits of resource and service modularity, identification, and upkeep that were recognized in hindsight by analyzing MedMinify after it was built and tested.

Specifically, this chapter focuses on the design and development of three initial technological components for a new *computable knowledge infrastructure*[1] called the *Knowledge Grid,* or KGrid. Besides helping to organize computable knowledge, KGrid's technical components are intended to make it easier to automatically apply computable knowledge to generate and communicate well-informed advice. In keeping with this intent, KGrid's software components have been designed and developed to overcome the noted deficiencies of MedMinify's system architecture. KGrid overcomes them by augmenting the interoperability of computable knowledge resources and related services through technical means and methods.

As a fledgling knowledge infrastructure for computable knowledge, the Knowledge Grid is designed to enable a paradigm shift away from large, monolithic, *stand-alone advice-giving systems*, which operate in proprietary ways, towards simpler, smaller, modular, *individuated advice-giving services*, which operate in open ways. One benefit of this approach is that open knowledge and advice-giving services can be drawn on by many different systems.

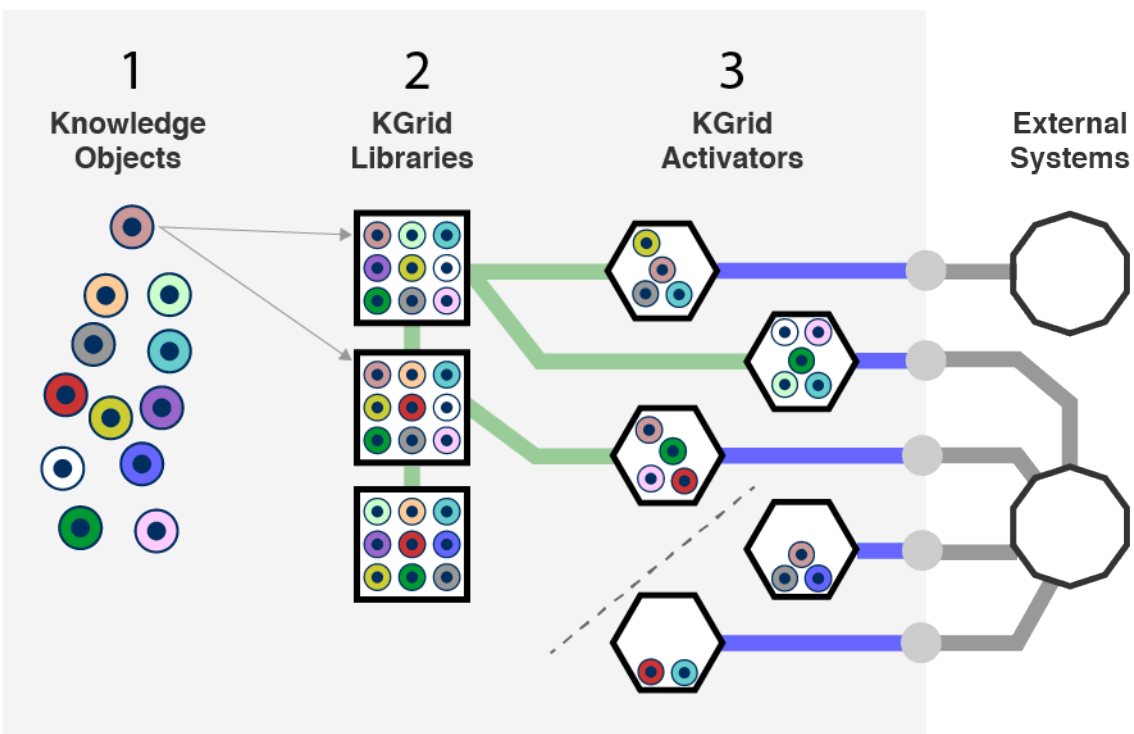
In this case, *open* signifies only an architectural perspective promoting service oriented machine interfaces for generating and communicating advice. Here, the term *open* does not connote free and unlimited access to content or services[2]. Usually, providing access to knowledge content and related services involves some work and cost.

The paradigm shift mentioned above from stand-alone advice-giving systems to individuated advice-giving services is of particular interest in the health domain today. It is hoped that *advice-giving services* may provide a manageable and suitable technical mechanism to generate and communicate the large quantities of well-informed health advice needed to support Learning Health Systems[3–7]. To meet this need is a major challenge.

KGrid's initial three technical components are (1) Knowledge Objects, (2) the KGrid Library, and (3) the KGrid Activator. This chapter briefly introduces these three components below and later provides detailed descriptions of them. These descriptions explain their purpose, design, and stage of development. As a result, this chapter encompasses a body of scientific work establishing KGrid's technical underpinnings as a *testbed system*[8,9]. A testbed system is a platform to conduct tests and studies. KGrid serves as a testbed for research and development on how to use compound digital objects[10] to solve some hard computable knowledge management problems.

KGrid's three technical components are briefly introduced next. Knowledge Objects are compound digital objects that combine an instance of computable knowledge with a service specification and descriptive metadata. The KGrid Library is a conventional digital library for storing and managing Knowledge Objects. The KGrid Activator is a technical service component for deploying the computable knowledge in Knowledge Objects.

Figure 3.1 The Knowledge Grid's Three Core Technical Components



The three core technical components comprising the Knowledge Grid, and the relationships between them, are depicted in the grey box in Figure 3.1 above. On the far left, a group of individual Knowledge Objects are shown in different colors. They can be collected in KGrid Libraries, three of which are portrayed as squares in the middle of Figure 3.1. Each of these KGrid Library instances holds different Knowledge Objects. Each KGrid Library connects over a computer network to another KGrid Library. These network connections are illustrated by short vertical green lines between the KGrid Library boxes.

Next, five instances of the KGrid Activator are depicted as hexagons near the middle of Figure 3.1. Each of these KGrid Activators contains several Knowledge Objects. Green lines indicate that the topmost three KGrid Activators have computer network links to KGrid Libraries. Using these links, KGrid Activators find and access Knowledge Objects. However, such links between KGrid Libraries and KGrid Activators are not required. A thin, angled dashed line helps to indicate that. The bottommost two KGrid Activators are not directly connected to KGrid Libraries in Figure 3.1. This is because KGrid Activators can operate independently from KGrid Libraries. The flexible modularity of these components enhances the overall utility of the Knowledge Grid by increasing their interoperability. In fact, this modular approach makes it possible to incorporate one or more of these components into other platforms and infrastructures besides the Knowledge Grid.[11,12]

Finally, on the far right of Figure 3.1, two external systems are depicted as decagons. The top External System connects on a network to one advice-giving service endpoint (•) provided by a single instance of the KGrid Activator. The bottom External System connects to four similar endpoints provided by the other four KGrid Activators. With this diagram in mind, the origin of the name of the Knowledge Grid can now be explained. This name stems from the links among KGrid Libraries and KGrid Activators that form a network or *grid* of these things.

Work on these three technical components of the Knowledge Grid offers opportunities to rethink the roles that knowledge plays in the world, and the forms that it takes while playing them. To wit, Weinberger has remarked about “knowledge’s new medium”, a reference to global-scale computer networks where knowledge is “never fully settled, never fully written, never entirely done"[13]. Similarly, Edwards et al. assert that, "the basic mechanics by which knowledge is produced and circulated"[1] are undergoing rapid change due to Internet technologies. Examples of some of these changes are discussed next.

Increases in the velocity of knowledge production and circulation are causing its production and circulation to splinter[14]. As one example of this, consider the growing importance of interim research products, such as scientific manuscript pre-prints. Pre-prints are scientific works that precede peer-reviewed articles. They are additional sources of scientific knowledge, some of which will never be peer-reviewed. Further, repositories of pre-prints represent yet another distinct type of resource where pieces of the global scientific record are now found.

As another example of the splintering of scientific resources across the World Wide Web, an example that is very relevant to this dissertation, consider the use of the privately-run GitHub, BitBucket, and SourceForge platforms for versioning and sharing research artifacts, including software code. The fact that three competing version control platforms exist makes their splintering effects obvious. Researchers have encouraged other researchers to use these platforms to help produce and communicate new knowledge[15]. One concern is that none of these privately-run platforms commit to serve as infrastructure for maintaining the persistence of the global scientific record[12]. In contrast, the Knowledge Grid is being developed to bring about a distributed, persistent scholarly computable knowledge infrastructure.

KGrid simultaneously exemplifies and reflects ongoing changes in the mechanics of knowledge production and circulation. It is an attempt to manage computable knowledge artifacts for science via the World Wide Web in ways that maintain their integrity for generations to come. It is also an attempt to make computable knowledge more interoperable. That part of the KGrid story is what mainly motivates this dissertation.

BACKGROUND

Purpose of The Knowledge Grid

The foremost purpose of the Knowledge Grid is to greatly decrease the multi-year latency between the moment new biomedical knowledge is discovered and its widespread uptake and use in practice[16]. The core technical components of the Knowledge Grid are domain-agnostic. However, their development has been specifically motivated by the well-established need to accelerate *global biomedical knowledge flows* for health[17–24]. Without an acceleration of biomedical knowledge flows, it is anticipated that current problems with health care underutilization, overutilization, disparities, and avoidable iatrogenic harms will persist, if not worsen[25].

Reiterating, to improve the flow of biomedical knowledge for health and meet its foremost purpose, the Knowledge Grid provides the technical bases for an emergent *knowledge infrastructure*. This knowledge infrastructure makes computable knowledge easily transferable and rapidly available via highly interoperable webservices accessed via the World Wide Web. In this way, computable knowledge can potentially be used by any machine capable of connecting to the Internet.

Context of Learning Health Systems

As noted above, efforts to develop the Knowledge Grid are situated within the context of a much larger ongoing movement to bring about high-functioning Learning Health Systems[3,25–27]. This movement seeks to combine, in credible and workable ways, capabilities for learning that already exist in the following areas: biomedical research[28–31], health care quality improvement[32], health care practice advancement[33,34], and personal health behavior betterment[35,36]. Through aggregation of these learning capabilities, large-scale Learning Health Systems are ultimately expected to continuously learn from the health experiences of many people while reliably providing the latest biomedical knowledge to most, if not all, people.

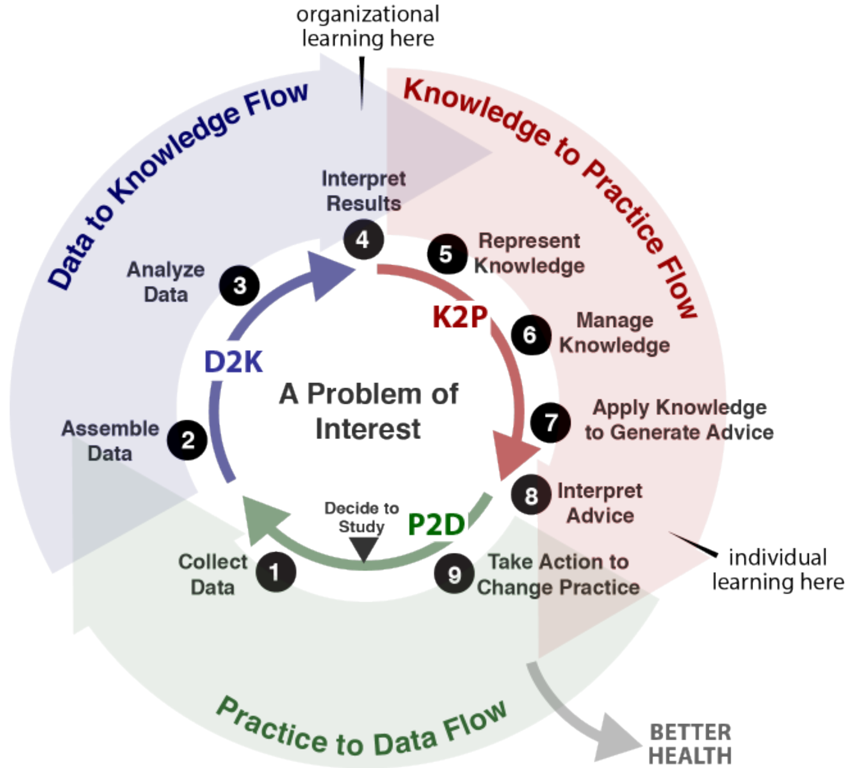
To achieve Learning Health Systems through aggregation of various existing capabilities for learning about health, it is thought that an advanced sociotechnical infrastructure for large-scale system-wide learning is needed[3]. This infrastructure must support ongoing, cyclical processes of learning called *learning cycles*. These ongoing *learning cycles* engage communities in empirical analysis of data relevant to biomedical problems, resulting in discovery of new knowledge via *organizational learning*. Then, *learning cycles* directly apply the new knowledge gained to change health care practices and health behaviors via *individual learning*. Next, *learning cycles* move forward by collecting new outcomes data resulting from updated and changed practices and behaviors. These new outcomes data drive succeeding *learning cycle* iterations, with improvement in health occurring as a byproduct of these interconnected processes. KGrid is an attempt to develop just one portion of this learning infrastructure.

The notion of ongoing *learning cycles* is not new. One historical example makes this point. While writing about organizational science in 1978, Susman and Evered describe a cyclical process for executing "action research", which is research that simultaneously adds to the overall body of general scientific knowledge while contributing real-world solutions to practical problems that people currently face[37]. In their work, they depict a model *learning cycle* with five phases. Their version of the *learning cycle* is akin to the much more commonly encountered *Plan•Do•Check•Act* (PDCA) cycle associated with statistical process control and continuous quality improvement since the 1940s[38].

A model of the *learning cycle* for Learning Health Systems is portrayed with three distinct information flows in Figure 3.2 below. These three flows are (1) a Practice to Data (P2D) flow, resulting in the collection of practice data, including *health outcomes data*, (2) a Data to Knowledge (D2K) flow, resulting in *organizational learning* by interpreting the results of biomedical data analyses, and (3) a Knowledge to Practice (K2P) flow, resulting in *individual learning* via interpretation of advice predicated on current biomedical knowledge. When these three information flows come together in the cyclic manner shown in Figure 3.2, they can result in better human health. Friedman summarized this idea in the following conceptual equation[3]:

$$P2D + D2K + K2P = Better\ Health$$

Figure 3.2 The Learning Cycle with Three Flows and Nine Steps



The Knowledge Grid is knowledge infrastructure purpose-built specifically to support a portion of the Knowledge to Practice (K2P) information flow. To support Step 6 in Figure 3.2, "Manage Knowledge", KGrid provides capabilities to manage large quantities of individual computable knowledge resources which will easily number in the millions of instances[39].

The K2P flow in Figure 3.2 also involves using knowledge to inform and generate *advice* at Step 7, "Apply Knowledge to Generate Advice." At this step of the *learning cycle*, up to date knowledge is combined with an adequate set of contextual information about an unmade decision to generate and communicate well-informed advice for human decision-makers. When used in this way to generate and communicate advice, computable knowledge is, in one step, accessed *and* combined with instance data to compute advice for a specific decision situation.

The Knowledge Grid takes advantage of Existing Information Infrastructure

To accelerate the flow of biomedical knowledge within Learning Health Systems, the Knowledge Grid attempts to support Steps 6 and 7 in Figure 3.2 above. When used for this purpose, its support must unfold on a global scale and it must encompass a huge scope. Its scope includes knowledge about the human body, thousands of diseases, and tens of thousands of medical tests, interventions, processes, and treatments. Yet, the technological components of the Knowledge Grid are not specific to the health domain. They can be used to organize and disseminate any knowledge that has a computable form suitable to be interpreted by machines.

To make the work of *accelerating knowledge flow* effective enough to propel Learning Health Systems, KGrid takes advantage of the scalable, global information infrastructure of the World Wide Web, and the underlying technologies of the Internet that enables the World Wide Web. This existing infrastructure undergirds the Knowledge Grid's technical components and adds considerably to their interoperability. The specific ways that the technologies and standards of the World Wide Web add to KGrid's interoperability are discussed in more detail later in this chapter.

Background about Knowledge Objects

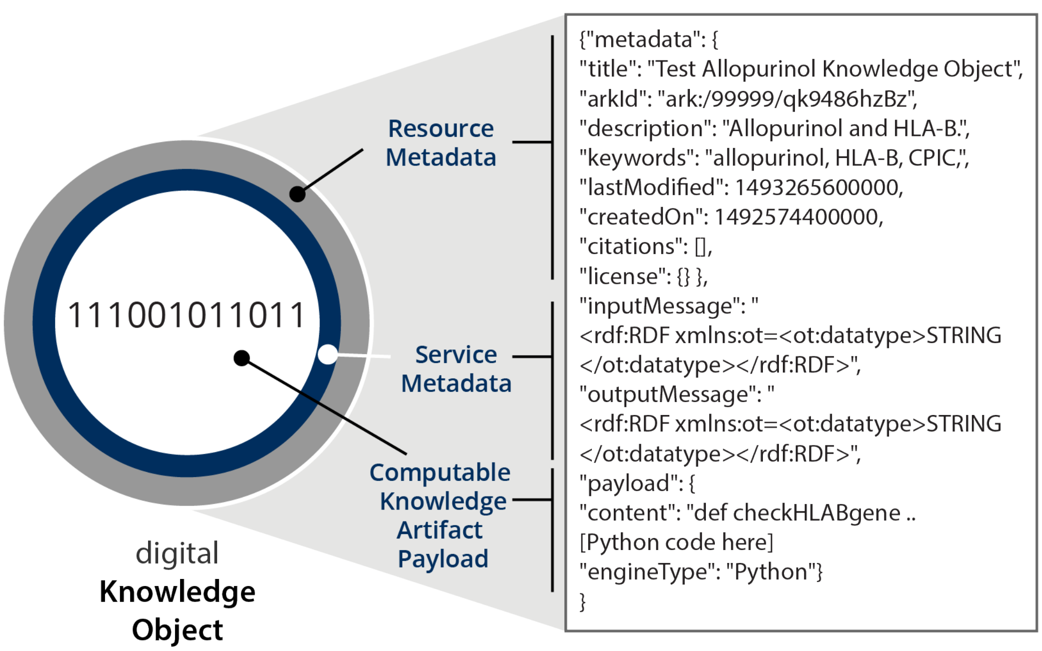
To help manage computable knowledge and use it to generate advice, this chapter puts forward a formal specification for a new compound digital object called a *Knowledge Object*. A Knowledge Object bundles an instance of computable knowledge, also called a *payload*, with useful resource and service metadata. Resource metadata include a persistent unique identifier for each Knowledge Object and statements describing its *payload's* provenance and purpose, among other things. In contrast, service metadata are limited to descriptions of the specific services that the Knowledge Object can enable[40]. Service metadata describe precisely how to interact with a Knowledge Object to get advice.

Yet before a formal specification for *Knowledge Objects* could be created, a more general conceptualization was needed to help us understand what a *Knowledge Object* could be[40]. This general conceptualization is depicted in Figure 3.3 above. In Figure 3.3, resource and service metadata are depicted as the outer two layers of a Knowledge Object. These outer two layers relate to two different portions of the text on the right. This text comprises a simplified example of a Knowledge Object. Also in Figure 3.3, a payload is depicted at the core of the Knowledge Object in 1's and 0's. This payload is associated with its own portion of the example Knowledge Object's text on the right. In the figure, the text of the example Knowledge Object is serialized in the JavaScript Object Notation (JSON) format. Other textual and binary formats are possible.

Real-world examples of computable knowledge *payloads* stored in *Knowledge Objects* include encoded predictive models, computable guidelines, and computable phenotypes.

Early work to develop the Knowledge Grid has focused attention on development of a formal specification that precisely defines what a *Knowledge Object* is. This chapter reports work that we have done to build on the conceptual notion of a Knowledge Object portrayed in Figure 3.3. We published our conceptual notion of a Knowledge Object in 2016[40].

Figure 3.3 Conceptual Knowledge Object model with an example in JSON



Background about the initial KGrid Library component

To support computable knowledge management and dissemination, the Knowledge Grid includes a conventional digital *KGrid Library* component (Figure 3.1). It is for managing, stewarding, archiving, and preserving computable knowledge that is stored and packaged in Knowledge Objects.

From here on, we will use the term *curation* to mean selecting, managing, stewarding, archiving, and preserving Knowledge Objects as a new type of material *artifact*. As indicated above, Knowledge Objects can be collected in KGrid Libraries to facilitate their curation. Obviously, there are many aspects to curation of Knowledge Objects, but most of those aspects are not addressed here. Instead, what is addressed here is the work to make Knowledge Objects as interoperable as possible and then to use them to augment the interoperability of the related knowledge services and advice-giving services that they enable.

One important aspect of Knowledge Object curation that *does* relate directly to the interoperability of Knowledge Objects is the KGrid Library-to-KGrid Library sharing of Knowledge Objects. We believe the ability to share some high-value, highly interoperable Knowledge Objects among KGrid Libraries is critical to the early success of the Knowledge Grid effort for the following two reasons.

First, for the Knowledge Grid to be successful, we have to show how it creates value. The ability to share Knowledge Objects is an initial move toward demonstrating value. Knowledge Object sharing has the potential to bring about economies of scale in computable knowledge creation, management, dissemination, and use. For this reason, we are exploring early uses of the Knowledge Grid to share an authoritative collection of Knowledge Objects. An example of this is work to create Knowledge Objects that encode and encapsulate guidelines about the genetic determinants of drug selection and dosing from the Clinical Pharmacogenetics Implementation Consortium.[41] Further details about this work are outside the scope of this chapter.

Second, one of the basic storage-related tenets of digital preservation is to make and curate multiple independent copies of digital artifacts. This principle has been articulated using the term Lots of Copies Keep Stuff Safe (LOCKSS).[42] The purpose of technology that supports the LOCKSS principle is to have backups on hand throughout the lifecycle of digital artifacts.[42,43] Although this dissertation is not about digital preservation, I recognize that long-term interoperability of computable knowledge artifacts - including Knowledge Objects - depends on thoughtful, systematic approaches to digital preservation.

Background about the initial KGrid Activator component

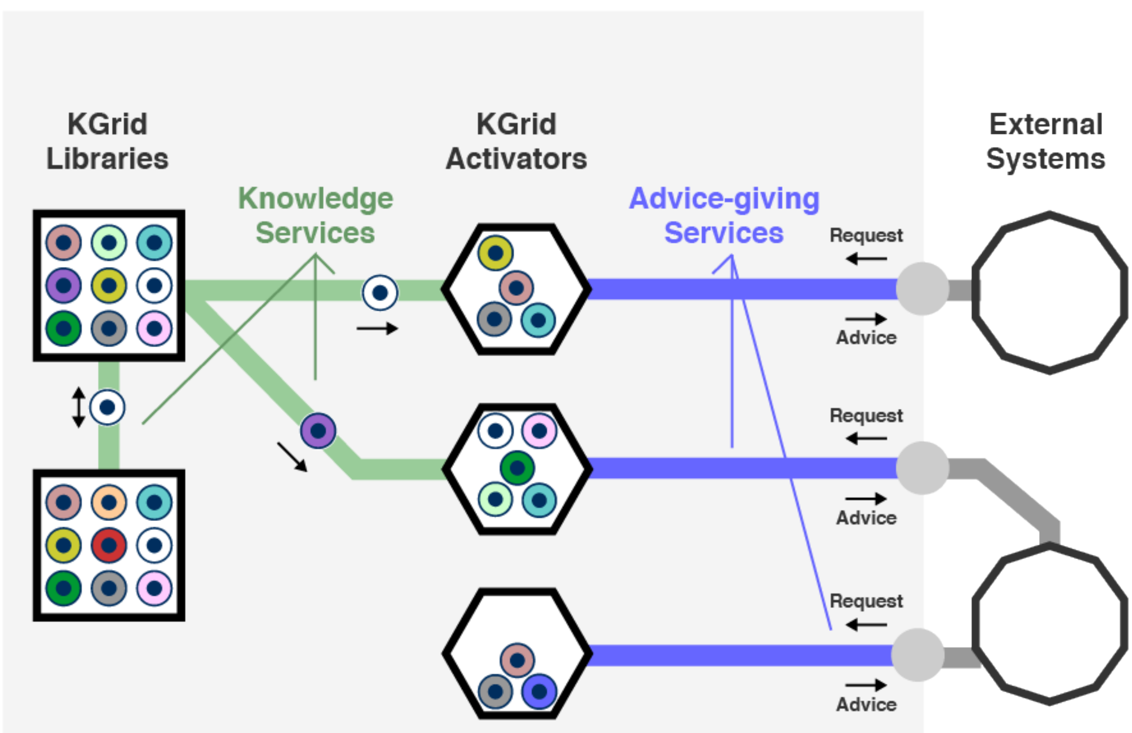
Along with Knowledge Objects and the KGrid Library, the Knowledge Grid also includes a core software component called the *KGrid Activator*. As indicated above in Figure 3.1, a primary role for the *KGrid Activator* is to make the payloads of Knowledge Objects serviceable for the purpose of generating and communicating advice. It does this by providing network access to Knowledge Object payloads in two different ways. Upon request, the KGrid Activator can send whole payloads to external systems where they can be used for a variety of purposes. More relevant here, though, the KGrid Activator can accept a request for advice from an external system and use Knowledge Object payloads to generate advice that is then communicated back to the requestor. For the remainder of this chapter, the KGrid Activator's capability to accept and respond to requests for advice from external systems remains in scope. However, its capability to shuttle Knowledge Object payloads to other systems is out of scope because that use of the KGrid Activator is not primarily about advice-generation.

With the background covered thus far in mind, we can now distinguish more carefully between two types of technical services provided by the Knowledge Grid. As illustrated in Figure 3.4, the Knowledge Grid provides both automated knowledge services (portrayed with green lines) and automated advice-giving services (portrayed with blue lines). Its network-enabled knowledge services shuttle Knowledge Objects, which are resources, from KGrid Libraries to KGrid Activators. Similarly, its network-enabled advice-giving services, which are provided exclusively by KGrid Activators, accept requests for advice from External Systems and meet those requests by combining the payloads stored in Knowledge Objects with inputs to generate and communicate advice. These requests for advice carry instance data, which are data that describe relevant features of a decision-situation related to the advice being sought. For the remainder of this dissertation, the focus is primarily on these network-enabled advice-giving services provided by KGrid Activators and not on the knowledge services that support them.

When scaled up to regional, national or international scales, primarily by making use of the World Wide Web, the loosely coupled interoperable components comprising the Knowledge Grid illustrated in Figure 3.4, have the potential to become large-scale computable knowledge infrastructure. If and when that happens, the change from a prototype platform to an actual knowledge infrastructure will be signaled by an "infrastructural inversion", which is Bowker's term for emergent capabilities, and related positive and negative effects, that arise when new infrastructures take hold in the world.[44]

I must point out, however, that the idea of providing advice-giving services via the World Wide Web is not new. Such services are well described in the scientific literature[45,46]. What is

Figure 3.4 Two Types of Automated Services Provided by the Knowledge Grid



new is that the Knowledge Grid, by virtue of the *KGrid Activator* and the KGrid Activator's Application Programming Interface (API), potentially makes it easier to rapidly and reliably deploy and update automated *advice-giving services* predicated on well-curated computable knowledge.

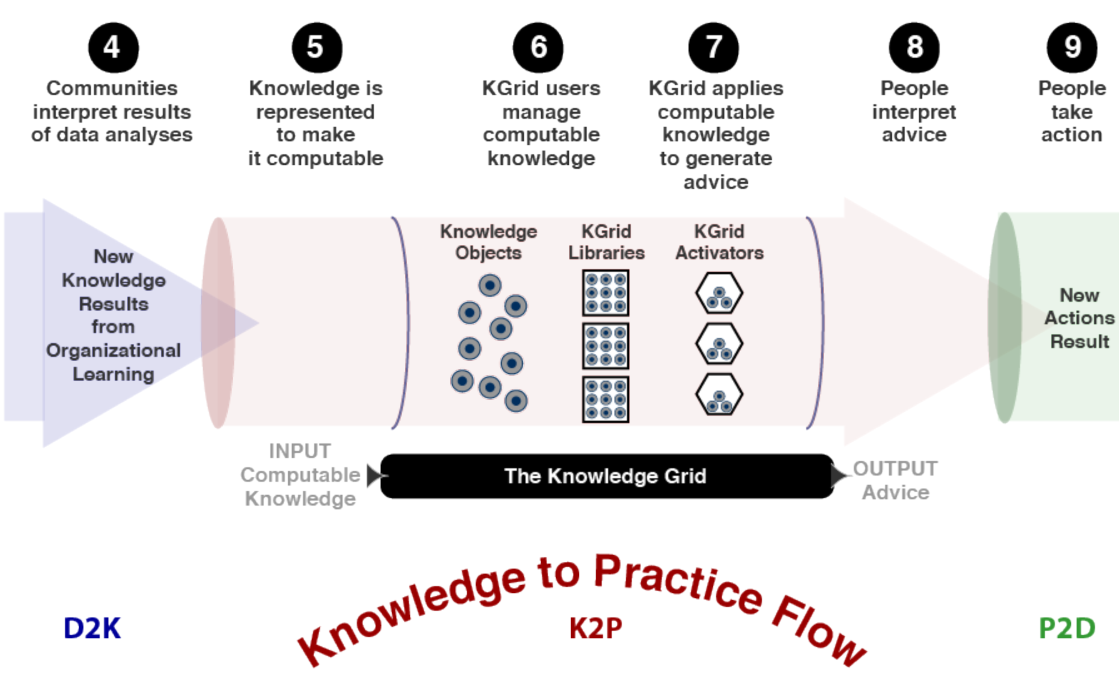
Pipeline view of the core technical infrastructural components of the Knowledge Grid

To put its core technical components in the context of learning cycles for Learning Health Systems, Figure 3.5 below illustrates how the Knowledge Grid provides supportive infrastructure specifically for the Knowledge to Practice (K2P) flow of learning cycles. It does so by using Knowledge Objects, instances of the KGrid Library, and instances of the KGrid Activator to generate advice about health.

In Figure 3.5, following numbered steps that were previously introduced in Figure 3.2, after a knowledge artifact is represented in a computable form (Step 5), then the Knowledge Grid provides the means to package it into a Knowledge Object. Next, Knowledge Objects are collected and managed within deployed instances of the KGrid Library software (Step 6). From deployed KGrid Libraries, instances of KGrid Activators find and access Knowledge Objects, and then *activate* their payloads by using them to bring automated advice-giving services online that generate and communicate advice. When these *advice-giving services* are then engaged by external systems on behalf of people, messages of advice targeted at unmade decisions faced by those people are received by them (Step 7). When people receive and interpret this advice (Step 8), the K2P information flow is completed. As depicted in Figure 3.5, the Knowledge Grid specifically supports Steps 6 and 7. It takes in computable knowledge, and supports both its curation and use to generate and communicate advice targeted at unmade decisions.

Summarizing what has been discussed about the Knowledge Grid thus far, Figure 3.5 depicts how the three core technical components of the Knowledge Grid function together as supportive infrastructure for managing computable knowledge resources and using them to deploy advice-giving services. Now that the core technical components of the Knowledge Grid

Figure 3.5 Pipeline View of the Knowledge Grid's Infrastructure Components



have been reviewed, the following subsection of this chapter provides more relevant but general background about the science of knowledge infrastructure as it pertains to the Knowledge Grid.

Pertinent background about compound digital objects and ontologies

This subsection gives some more detail about several topics related to knowledge infrastructure and the Knowledge Grid. It begins by comparing KGrid's Knowledge Objects to other similar, preceding types of digital objects. Then, ontologies are introduced and their roles within the context of knowledge infrastructure are briefly outlined. Finally, an explanation is given for the purposeful use of conventional digital libraries as part of the Knowledge Grid.

About compound digital objects. The idea of creating compound digital objects to hold and annotate various types of content, including knowledge in various forms, is not new. This idea has been seriously explored for more than 20 years, starting with an effort to develop a framework for distributed digital object services completed by Kahn and Wilensky in the 1990s[10]. Building on their work, and on previous work to develop the Dienst architecture for distributed document libraries[47], Payette and Lagoze built the Flexible Extensible Digital Object and Repository Architecture (FEDORA)[48]. This important groundbreaking work makes it possible to further progress the science and technology of compound digital objects via the Knowledge Grid.

When considering KGrid's Knowledge Objects, two similar types of digital objects deserve special mention, *Learning Objects* and *Research Objects*. These two other types of compound digital objects are forerunners of the Knowledge Objects developed here.

Learning Objects 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 Learning Objects[49]. Learning Object content includes text, images, audio, and video files, and software applications. Learning Objects have mostly been used in the context of Learning Management Systems. These systems are typically used by large organizations to train their members on a variety of mandatory competencies, and to track when required training occurred.

Yet the scope of Learning Objects is so broad that it is difficult to formally define them in a way that can be widely accepted[49,50]. Even so, for the purpose of achieving greater interoperability, the technical Shareable Content Object Reference Model (SCORM) standard has been adopted and used in many Learning Management Systems to date[51]. A "next generation" version of SCORM, based on service-oriented web architectures, is planned[52].

Taking note of this history about Learning Objects and SCORM, we took a different approach with Knowledge Objects. Our approach is to develop an ontology that formally and precisely describes what a Knowledge Object is from the outset. Then it is to put the new ontology through peer review. Finally, it is to make it as widely available as possible.

Our approach involves two methods that the developers of SCORM did not use[52]. The first of these methods is to adopt an upper level ontology. The upper level ontology we used is described later. We used it so that the formal specification of a Knowledge Object can, in theory, interoperate with formal specifications for other entities that share the same upper level ontology. The second method is to define a hierarchy of the entities comprising Knowledge Objects and the properties that relate those entities. We did this to define, prescribe, and limit the classes of *information content entities* that Knowledge Objects can contain. Unlike the SCORM specification for Learning Objects, our specification for Knowledge Objects does not allow Knowledge Objects to hold any content in any format. Instead, Knowledge Objects have constraints. They are compound digital objects comprised of instances of computable knowledge, resource metadata, and service metadata. The main point is that, compared to the way Learning Objects are specified, our work to formally specify Knowledge Objects uses a somewhat stricter specification in an attempt to improve Knowledge Object interoperability. Whether a stricter specification actually improves Knowledge Object interoperability remains an open question.

Another preceding and relevant type of object is the Research Object. A Research Object ontology exists[53]. Per that ontology, a Research Object is an *evolving aggregation* of heterogeneous resources for describing and reproducing research workflows, along with annotations of those resources[54]. The heterogeneous resources that can be placed inside Research Objects include, but are not limited to, data sets, lab notebooks, analytic workflows, intermediate results, draft manuscripts, and publications. Therefore, the scope of the Research Object ontology covers what is needed to bundle essential information relating to experiments and investigations for the purpose of increasing scientific transparency and study reproducibility.

The Knowledge Object Reference Ontology (KORO) introduced in this chapter has a different scope. The scope of KORO covers what is needed to build compound digital objects that store instances of computable knowledge and describe, for each such instance, at least one related service that the instance of computable knowledge supports and enough other details about each instance of computable knowledge and Knowledge Object itself to make them as findable, accessible, interoperable, and reusable as possible[55].

About ontologies. The idea that ontologies have important roles to play within knowledge infrastructures is also not new. When confronted with the challenges of classifying information artifacts that continue to grow in number, Weinstein and Alloway point to the expressiveness and precision of ontologies as reasons why ontologies are useful for automated creation and curation of information artifacts at scale[56]. More recently, ontologies have been developed specifically for the World Wide Web, enabling it to support enhanced data aggregation. As one example of this, the website schema.org provides ontologies to improve search accuracy for the World Wide Web by making it easier to extract relevant information from textual resources facilitating their curation[57,58].

In the context of this chapter, an ontology is defined as a, "standardized representational framework providing a set of terms for the consistent description of data and information across disciplinary and research community boundaries.[59]" In essence, as they are often used in computing and information science, ontologies are formal, explicit, and logically consistent specifications of hierarchies of entities and other properties that relate entities. There are well-established methods to represent ontologies in fully computable forms, and tools that help people to do so[60–62].

Ontologies are used by the Knowledge Grid mostly as structural frameworks for text-based compound digital Knowledge Objects and their constituent textual parts and subparts. The most important ontology for the Knowledge Grid to date is the Knowledge Object Reference Ontology (KORO). Its design and development are described later in this chapter.

We use multiple ontologies, including KORO, as structured models for representing hierarchical "is_a", or subsumption, relationships, "has_part" and "part_of" parthood relationships, and other relationships besides[56]. Once relationships between the entities are established as axioms within a given ontology, machines can then reason using the ontology to label or "tag" entities automatically. Here we use the Web Ontology Language, called OWL, to represent the computable axioms that comprise the KORO ontology for Knowledge Objects.

Within biomedicine specifically, two initiatives towards the coordinated evolution of biomedical ontologies are particularly relevant to our efforts to design and develop the Knowledge Grid. These are the establishment of the National Center for Biomedical Ontology (NCBO)[63] and the efforts of the Open Biomedical Ontologies (OBO) consortium, which have resulted in the OBO Foundry[64].

The NCBO provides an online website known as Bioportal[65]. This website provides a collection of several hundred ontologies and terminologies. The ontologies on the Bioportal enable their users to automatically annotate textual source materials by mapping arbitrary words in texts of interest to ontological terms, thereby "tagging" the texts with those precise terms. The results of this mapping and tagging of texts can then be used to improve search performance. This kind of automated annotation through mapping and tagging is of great interest for improving the findability of Knowledge Objects as work on the KGrid Library progresses. We have uploaded the KORO ontology we created to the public Bioportal website.

The OBO Foundry has an ambitious goal of developing ontologies by using a shared set of design principles. The Foundry's goal is to create ontologies that are interoperable and combinable[64]. The OBO Foundry seeks to use ontologies to standardize data coding schemes so that clinical data sets may cumulate over time and across organizational boundaries. Similar aggregation of structured data over time is also important for the Knowledge Grid. We will return to the topic of ontology-enabled aggregation of knowledge resources in Chapter IV.

Besides aggregation, however, the Knowledge Grid has several other needs for support from ontologies. Ontologies are needed to improve Knowledge Object findability, curation, and interoperability.

Besides using the KORO ontology to specify the overall structure of Knowledge Objects, various other ontologies can give structure specifically to the resource metadata of Knowledge Objects. These include ontologies for describing people, events, and provenance. Using them can improve findability and facilitate curation. However, these important capabilities are not the focus here.

Other ontologies can give structure to the service metadata of Knowledge Objects. This is particularly important for augmenting Knowledge Object interoperability. Ongoing work to upgrade the service metadata of Knowledge Objects to support *type introspection* is an example of this. *Type introspection* is the ability of an arbitrary software program to examine the properties of a digital object, such as a Knowledge Object, in order to put digital objects to use automatically. To bring about *type introspection* for Knowledge Objects requires Knowledge Objects to be able to expose their service metadata in a structured and standardized way to other systems using an ontology. For this purpose, we are presently experimenting with the Open API standard[66]. Open API will allow us to describe, in a machine-processable way, the format of the inputs needed to engage a Knowledge Object based advice-giving service and the format of the advice output the advice-giving service generates and communicates.

Background about the initial KGrid Library component

As previously discussed, development of the Knowledge Grid is situated in the context of supporting Learning Health Systems with better knowledge infrastructure. A core technical component of the Knowledge Grid is a conventional digital library component called the KGrid Library. The purposes of the KGrid Library are to enable the collection and curation of Knowledge Objects, and to provide automated knowledge services, as illustrated above in Figures 3.1, 3.4, and 3.5. Here, the metaphor of the library is being used because of the need to uphold the integrity of Knowledge Objects by curating them.

Yet Internet technologies make it so that all digital artifacts, including Knowledge Objects, are subject to constant revision and repurposing.[11] The fact that digital artifacts can potentially be constantly in flux creates operational problems for health organizations. These problems translate into challenges for the Knowledge Grid. One challenge is that, because the health domain involves high-risk decisions and actions, there are requirements to save the internal system states of health IT systems and to support rollback to previous system states. The legal aspects of this challenge are addressed elsewhere.[67] We are only beginning to consider the technical aspects of this challenge. Therefore, for my purposes here, it is enough to point out that the role of a conventional digital library for managing Knowledge Objects that will be used to generate advice about human health is not yet clear. It may be that such digital libraries will have to uphold a set of health care specific policies, e.g., policies for Knowledge Object version control, in order to handle Knowledge Objects effectively. Future work, beyond what I report in this dissertation, is needed in this area.

For the purpose of establishing trust and for a host of other medicolegal reasons, the health care system essentially requires a persistent record of all versions of Knowledge Objects that are used to generate and communicate advice. Recently, the United States Food and Drug Administration (FDA), which regulates medical devices, shared draft guidance for industry about clinical and patient decision support software. Via its draft guidance, the FDA makes it clear that it values capabilities that enable health care professionals to “independently review” the basis for automatically-generated medical advice about diagnoses and treatments. FDA’s guidance suggests that advice-giving systems for clinical and patient decision support without such capabilities are more likely to be considered medical devices[68]. Assuming this draft FDA guidance holds, there is an opportunity for the Knowledge Grid to demonstrate how curation and versioning of Knowledge Objects promotes the independent review of the evidence upon which they are based.

Looking ahead, KGrid anticipates a future where a host of personal “digital health tests”, akin to laboratory tests, are performed using instances of computable biomedical knowledge. These new digital tests will be done in the manner that laboratory tests and radiological imaging studies are frequently done today. However, unlike laboratory tests or imaging studies, which require tissue to be sampled, specimens to be drawn, or people to be present for imaging, digital health tests will process our health data in the background, and only return results when something of potential clinical relevance is calculated. If this future vision comes to pass, then the KGrid Library will have to function as a conventional digital library for curating the Knowledge Objects that support useful digital health tests. For this to happen, controls over Knowledge Objects will be needed so that the origin of, and rationale for, advice from digital health tests can be completely understood long after advice is given.

It may also be that the broader domain of biomedical science requires fixed and persistent versions of the computable knowledge resources it produces. Today, fixed works of peer-reviewed scholarship are heavily relied upon in clinical practice. For example, oncologists are now required to identify prior published studies that directly support their cancer treatment decisions before treatments are enacted[69]. To support this use of published biomedical knowledge, publications must be findable and accessible to clinicians. In the future, when more biomedical knowledge is available in machine-interpretable forms, it will also need to be made interoperable and reusable. These four principles, to make computable knowledge findable, accessible, interoperable, and reusable, or FAIR, guide the work of developing the Knowledge Grid[55].

Background Summary

The Knowledge Grid's core technical components are designed to overcome several limitations of stand-alone advice-giving systems. These limitations pertain to the curation and use of computable knowledge resources and related services. The rest of this chapter relates a body of scientific work undertaken to design and develop technical knowledge infrastructure components for KGrid which are modular, uniquely and persistently identified, and readily updateable. In the next chapter, technical KGrid components with these properties will be tested to evaluate the degree to which they augment the interoperability of computable knowledge resources and related services.

RESEARCH QUESTIONS

I address the following six research questions (*RQs*) in this chapter. The first four research questions pertain to the design and development of a formal specification of a Knowledge Object. The fifth and sixth research questions pertain to the design and development of the KGrid Library and KGrid Activator components of the Knowledge Grid, respectively.

RQ1. What are required and optional informational 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 an actual material digital Knowledge Object, when concretized, and how may they be represented and made logically consistent in an ontology?

RQ3. What questions about Knowledge Object parthood can an ontology answer competently in order 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?

RQ5. What are a minimum set of necessary and sufficient technical subcomponents, which, when integrated, can bring about a stable, scalable digital library capable of storing Knowledge Objects, making them discoverable and shareable, facilitating their incorporation into various other applications, and supporting both Knowledge Object and digital library management?

RQ6. What constitutes a minimum set of necessary and sufficient technical subcomponents that, when integrated, result in stable, scalable, rapidly deployable, secure, traceable, *activated* Knowledge Objects, thereby enabling KGrid to provide performant and reliable on-demand advice-giving services for health?

METHODS

Methods for formally specifying what a Knowledge Object is

Initial search for relevant existing ontologies. Before creating an ontology of our own to formally specify what a Knowledge Object is, we reviewed ontologies from the NCBO's online Bioportal and from the OBO Foundry to see if an existing ontology might suffice for this purpose. We were unable to identify an existing ontology for compound digital objects intended to help manage and share computable knowledge. We did find the Software Ontology (SWO), which supports annotation of software tools by type, manufacturer, inputs, outputs, and uses. However, SWO is evidently a model for software applications, such as Microsoft Word[70]. SWO does not provide enough detail to model the parts and pieces of compound digital objects like Knowledge Objects.

Choosing an upper level ontology. Prior to developing the Knowledge Object Reference Ontology, called KORO, we addressed the question of which, if any, upper level ontology to use. Recall from above that an upper level ontology potentially enhances the interoperability of all other ontologies that incorporate it by founding them on some shared model of reality[59,71]. Although several choices for upper level ontologies exist, we decided to use the Basic Formal Ontology (BFO) as an upper level ontology for KORO. We chose BFO because it has been purpose-built for scientific work and because it's creators have developed a set of step-wise procedures to follow when using it. BFO is described next.

BFO is an upper level ontology that has been purpose-built to support scientific endeavors[59]. Its purpose is to better describe science. BFO is a small ontology. It makes a few key metaphysical commitments about reality which can be built upon by other ontologies. Those who use it adopt these metaphysical commitments. For example, at its highest level, BFO begins by dividing reality, i.e., our conception of the world, into two views. One view takes the perspective of persistent entities, like buildings. Another view takes the perspective of transitory entities, like the process of entering a building, a process which starts and stops. Next, BFO categorizes persistent entities into immaterial entities, such as concepts, and material entities, such as physical objects. In this manner, what BFO does is lay down a short series of metaphysical commitments in a logically consistent way to support scientific communication.

While investigating BFO, we discovered that some of its creators had already extended it further to create the Information Artifact Ontology (IAO)[72]. IAO incorporates the upper level classes of entities from BFO, and then extends them to model additional classes for information entities. We found these information entity classes to be very useful for formally specifying what a Knowledge Object is with KORO[72]. For this reason, we ultimately based KORO directly on IAO, which in turn is based on BFO.

KORO ontology design and development procedures. A multidisciplinary team at the University of Michigan, led by Flynn and comprised of faculty, graduate students, and developers, collaborated to create KORO. At the outset, we determined that KORO needed to serve as a formal specification describing all of the parts of Knowledge Objects and how those parts relate. For this work, we followed the steps to develop ontologies using BFO given by Arp, Smith, and Spear[59].

We started the work to design and develop KORO with a copy of the Information Artifact Ontology (IAO 1.0[72]) that was previously merged with BFO[59]. In this case, IAO was represented in the Web Ontology Language (OWL) format. OWL is an ontology language that formally describes domains of interest in terms of entities, expressions, and axioms. In OWL, entities are either classes, properties, or individuals; expressions communicate criteria about entities; and axioms communicate assertions of the truth.[73]

BFO and IAO provide super-classes from which we derive upper level KORO concepts and terms. When deriving terms for KORO from BFO and IAO, we used the Minimum Information to Reference an External Ontology Term (MIREOT) method[74]. MIREOT defines the least amount of information needed to incorporate information from one ontology into another ontology. Its purpose is to limit unnecessary duplication and improve precision when deriving entities and properties from those in a preceding ontology.

In cases when external sources other than BFO and IAO inspired terms used in KORO, those external sources were explicitly noted in the documentation for the KORO ontology. For example, we incorporated some fundamental bibliographic concepts of the Functional Requirements for Bibliographic Records model into KORO[75].

For the first version of KORO, we focused our efforts on defining Knowledge Objects and their parts using the class of persistent entities that BFO calls *continuants*. In BFO, continuants are universals that continue or persist in time[59]. Following in the manner of IAO, we modeled the *conceptual* parts of a Knowledge Object Content Package as *information content entity* continuants. We also modeled the *material* parts of actual digital Knowledge Objects, and whole Knowledge Objects, as BFO continuants.

KORO provides Aristotelian definitions for all KORO classes and properties. These definitions follow a precisely defined format (see examples below). After incorporating BFO and IAO classes and their relations, we created KORO-specific content through a process of term identification. We gathered established terms for the parts of Knowledge Objects from the digital library and information technology worlds. For example, the terms "Fact Sheet", for a document that is a collection of key facts and "Log", for a document resulting from automatic recording of activity by a computer, were included in KORO. After assigning existing terms to most parts of KOs, we formalized what is meant by the novel terms in the ontology, e.g., the novel 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 (also called subclasses). Then we added to KORO the minimum number of additional relations needed to represent parthood, notably the relations 'has_part' and 'is_part_of.' 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 1.2 of KORO, serialized in OWL. We refined KORO with the help of feedback from experts. We uploaded our original version (KORO 1.0), and three improved versions (1.1, 1.2, and 1.3) 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[76]. 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[61] software and the automated Pellet reasoner[77], we created instances of 'Knowledge Object' and 'Knowledge Object Part Item' suitable to demonstrate that KORO properly infers answers to the three competency questions above.

Building of Exemplar Knowledge Objects to Illuminate Implementation Decisions. 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 Knowledge Objects. It does not constrain the format(s) of Knowledge Object parts, leaving these to be determined by those who create Knowledge Objects. Hence, to help identify implementation decisions that have to be made when creating Knowledge Objects, we have built exemplar Knowledge Objects of various types. The process of building actual Knowledge Objects allowed us to list an initial set of implementation questions that need to be answered about each part of a Knowledge Object when building new Knowledge Objects. These questions address the formatting and serialization of various parts comprising compound digital Knowledge Objects. They appear in the Results section for *RQ4*.

Methods for KGrid Library design and development

To create an encompassing architecture for the KGrid Library, accounting for all of its *technical components*, we analyzed the system architectures of several other existing digital library platforms with Application Programming Interfaces (APIs) and we screened a number of other related, open source technologies. Here, a *technical component* is a single entity in a larger system architecture, or platform, that can be implemented independently by various technical means. The following two paragraphs highlight some of the systems and technologies that we studied.

The Fedora repository project brings a fundamental component to the KGrid Library's system architecture[78]. Fedora, version 4.x, is open source middleware for digital repositories that combines native Resource Description Framework (RDF) *linked data platform* capabilities with a scalable capacity for *binary file storage*[79]. The Fedora repository middleware can scale up to store millions of digital holdings with their RDF descriptions. Two existing digital library platforms built on the Fedora repository were also studied: Islandora and Hydra, which is now called Samvera[80–82]. In addition to these, we studied the OpenICPSR platform for organizing and curating scientific data sets, which is a Fedora-based research data set repository built using the proprietary Archonnex Architecture[83,84].

The *California Digital Library* provides a needed technical service to the KGrid Library called EZID. EZID is used to create and manage long-term globally unique identifiers (IDs) for digital objects[85]. EZID mints Archival Resource Keys (ARKs[86]) that are used by the KGrid Library to solve the problem of uniquely identifying Knowledge Objects with universal identifiers. ARKs support finding these objects in a universal way, regardless of the KGrid Library instance in which they are held.

During KGrid Library development, we routinely tested and improved the capabilities of the KGrid Library to store and manage actual Knowledge Objects. For this work, KGrid Library holdings consisted of dozens of digital Knowledge Objects that we built using KORO as a formal specification. Through three major development cycles to date, the KGrid Library component has been extended and improved, resulting in a mature prototype.

The KGrid Library prototypes were designed and developed with input and support from University of Michigan faculty Charles P. Friedman, Carl Lagoze, Zach Landis-Lewis, Tim Pletcher, and Douglas Van Houweling. For the KGrid Library platform, the work (a) to design the front-end “look and feel”, (b) to define initial KGrid Library requirements, and (c) to determine what will be the “technology stack” for the KGrid Library is primarily that of Flynn and Boisvert, working in conjunction with Rampton[87]. The work to develop the software to make the KGrid Library capable of meeting its requirements is primarily that of Bahulekar, Boisvert, Gittlen, and Meng[87].

Methods for KGrid Activator design and development

To create an encompassing architecture for the KGrid Activator, one that delineates a set of necessary technical components and is extensible, we analyzed various cloud computing architectures developed in the context of health information systems[45,46,88]. In addition, we also studied the differences between a Remote Procedure Call (RPC), Remote Method Invocation (RMI), and the network-based RESTful architectural pattern for a stateless, scalable, generic networked interface with standard HTTP semantics (i.e., GET, PUT, POST, etc.)[89].

The KGrid Activator is an independent component built with the Spring Framework for Java[90]. Using the Java Development Kit, Spring’s *Web* module, and Maven for Java software for “build automation”, we developed the KGrid Activator by combining a series of interdependent Java classes and object instances to bring about *Knowledge Object based* services. To demonstrate the feasibility of the KGrid Activator, we have successfully deployed it on the Google Cloud and Amazon Web Services platforms.

The KGrid Activator is a stand-alone technical component of the Knowledge Grid that has been developed to work in conjunction with the KGrid Library. Via the KGrid Library’s own application programming interfaces (APIs), the KGrid Library receives requests from the KGrid

Activator for individual Knowledge Objects. Upon request, the KGrid Library then serializes Knowledge Objects in JavaScript Object Notation (JSON)[91], making them easier to transfer, and delivers them to the KGrid Activator. The KGrid Activator then uses the Knowledge Objects it receives to enable knowledge services based on the payloads in those objects.

To increase interoperability, Friedman initially specified that the KGrid Activator should operate as an independent component in the Knowledge Grid. This requirement led the development team to explore how to use APIs effectively to engender the *knowledge services* and *advice-giving services* that shuttle and apply the payloads of Knowledge Objects within the Knowledge Grid (Figure 3.4).

During KGrid Activator development, we routinely tested and improved the capabilities of the KGrid Activator to provide RESTful knowledge services using the actual payloads of Knowledge Objects. For this work, a variety of Knowledge Objects were built to test various KGrid Activator capabilities. In particular, the following two paragraphs describe fundamental knowledge service capabilities of the KGrid Activator that have been rigorously tested.

A capability was tested to demonstrate that the KGrid Activator can accept requests from external systems, reliably execute computations using Knowledge Object payloads encoded in Python or Javascript code, and return the results of those computations to the requesting external systems.

Another capability was tested to demonstrate that the KGrid Activator can accept requests from external systems and, in response, reliably deliver to those external systems a copy of the payload of a Knowledge Object with which the external systems can perform their own computations.

Through two redesign and redevelopment iterations to date, the KGrid Activator has advanced over a period of 18 months from its early prototype stage to its current, mature prototype stage.

The KGrid Activator was designed and developed with support from University of Michigan faculty members Charles P. Friedman, Carl Lagoze, Zach Landis-Lewis, Tim Pletcher, and Douglas Van Houweling. The work (a) to define KGrid Activator requirements, (b) to test and confirm that the KGrid Activator meets those requirements, and (c) to determine what will be the “technology stack” for the KGrid Activator is primarily that of Boisvert, working in conjunction with Flynn[92]. The work to develop the software to make the KGrid Activator capable of meeting its requirements is primarily that of Boisvert, Gittlen, Gross, and Meng[92].

RESULTS

The results of the research and development work to address the six research questions (*RQs*) about the Knowledge Grid covered in this chapter are entirely descriptive in nature. This section describing these results is comprised of the three following subsections, which follow in sequential order: (1) KORO: a formalism for Knowledge Objects and their parts, (2) Overview of the system architecture of the KGrid Library, and (3) Overview of the system architecture of the KGrid Activator.

KORO: a formalism for Knowledge Objects and their parts

Results for the Knowledge Object Reference Ontology (KORO) begin with the definitions of key terms. Many of KORO’s terms come from BFO and IAO. Its terms define unique classes of entities (things), or they define relationships that may exist among and between classes of entities. All KORO definitions are given in the following Aristotelean form:

$$Species\ (S) = def.\, a\ Genus\ (G)\ {\scriptstyle that/whose/with}\ Ds,$$

where “def.” stands for *definition*, and where “Ds” stands for one or more *differentia* that tells what it is about a certain Genus, in virtue of which they are also a Species[59]. To clarify, these three variables, S, G, and Ds are indicated in the first Aristotelean definition given below.

To begin, BFO separates all ‘Entities’ in the world into the following two classes:

CONTINUANT (S) = def. an **ENTITY** (G) that continues or persists through time (D_1)

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 of which KORO includes:

GENERICALLY DEPENDENT CONTINUANT = def. a **CONTINUANT** that is dependent for its existence on 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’, such that the ‘Entity’ can migrate from bearer to bearer as when copies of the same digital file migrate from one hard drive to another hard drive. In contrast, ‘Specifically Dependent Continuants’ cannot migrate in this manner. Instead, they depend on specific - not generic - ‘Independent Continuant’ bearers in the world for their existence. An example of an independent continuant that bears only specifically dependent qualities is a person. Unlike a digital file, a person cannot migrate intact from one material representation of itself to any other. In other words, people cannot exhibit generic dependence.

As noted above, IAO extends BFO, starting with a key subclass of ‘Generically Dependent Continuant’ that is defined by this term:

INFORMATION CONTENT ENTITY (<u>**ICE**</u>) = def. an **ENTITY** that 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

From IAO, KORO incorporates the term ICE (defined above), along with several terms for ICE subclasses, with one underlined 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 <u>some **ENTITY**</u> in a process

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

During our work, we learned from other experts that some definitions in IAO are contested and in flux. The definition for 'Directive Information Entity' is one of these. As a result, the underlined change in the IAO definition of 'Directive Information Entity' adopted by KORO removes an ambiguity. This is our attempt to clarify that members of the class 'Directive Information Entity' are procedures or instructions that can be followed and which result in some *thing*, i.e., some entity being "realized in a process"[72].

In addition to the terms for the classes defined above, KORO then adds a number of new terms. It uses them to help define a key new subclass of 'Document', which is the subclass termed 'Knowledge Object Content Package.' That class 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** that includes or contains, 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** that is comprised of one or more **KNOWLEDGE CONTENT ENTITIES** that may or may not also have other **ICEs** as its parts

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 is an empirical 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 **ICEs**

The pragmatic definition of 'Knowledge Content Entity' above is informed by prior work of the Computer Supported Cooperative Work (CSCW) community. This prior work focuses on knowledge management and views knowledge as something that is situated in particular social and temporal context[93,94]. The definition of a 'Knowledge Content Entity' also reflects the contended epistemological notion that knowledge is information that has somehow been upgraded[95,96]. 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** that is 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, among **DATA ITEMS** that have been previously observed and demonstrated
>
> THEORETICAL MODEL = def. a **MODEL** that hypothesizes, 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[97]:

> ARTIFACT = def. a **MATERIAL ENTITY** that is 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 relating these many defined terms for entities, KORO formally defines a 'Knowledge Object' and 'Knowledge Object Part Item' in these ways:

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 is a bearer of a concretization of a required or optional part of a **KNOWLEDGE OBJECT**

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 with the Knowledge Grid.

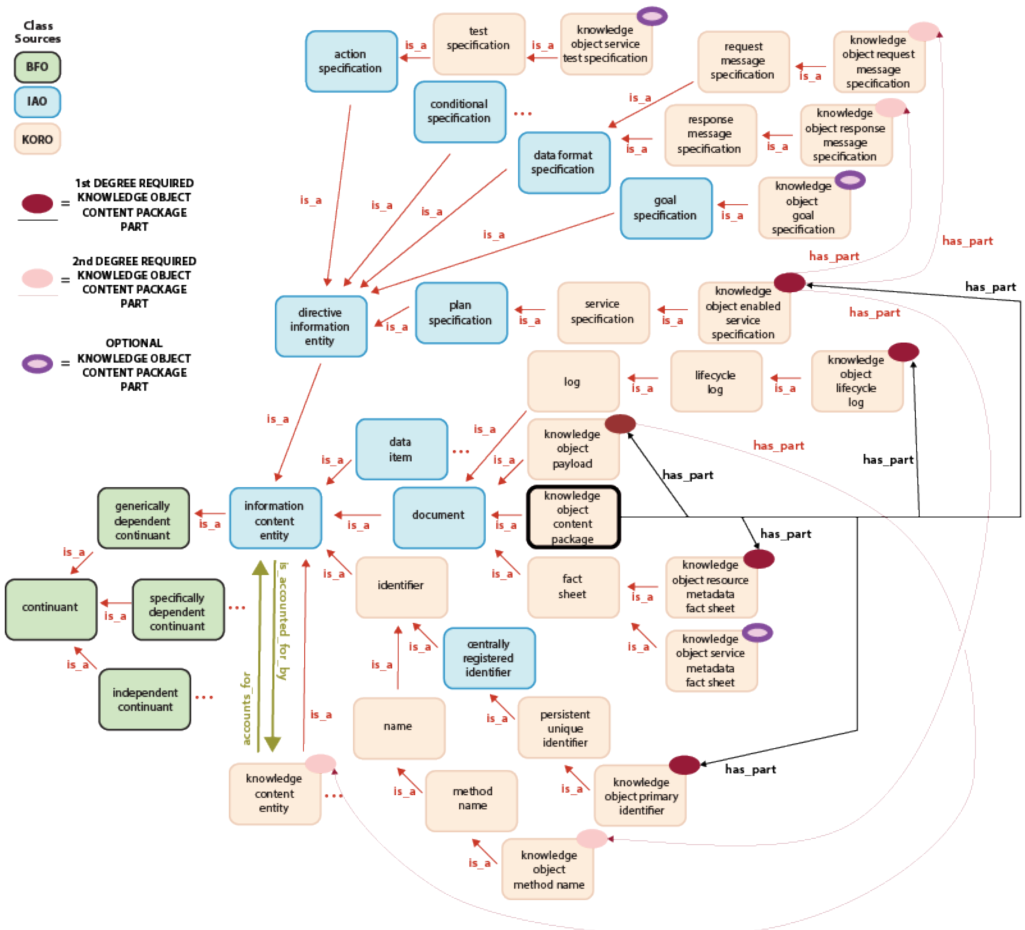
Having reviewed many fundamental terms and their definitions in KORO, we can now proceed to report the remainder of the results about KORO. What follows are results organized as responses to the first four research questions addressed by this chapter.

***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 3.4 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 3.6, the key class, 'Knowledge Object Content Package', appears toward the center. There are five "1st Degree" parts that *must be* included in a 'Document' for it to be an instance of the class 'Knowledge Object Content Package.' Subclasses for these five "1st degree"

Figure 3.6 Portion of KORO Indicating the Required and Optional Parts of a Knowledge Object Content Package



parts are marked with a red oval and connected to the ‘Knowledge Object Content Package’ class in Figure 3.6 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 3.6)

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 that exists 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 that exists between a **KNOWLEDGE CONTENT ENTIT**Y 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[97]. In this regard, we assert, as part of these results, 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, procedures, e.g., "lock the door at 4pm", do not "qualify alethically", meaning they cannot be correctly qualified as true or false[98]. We have been able to create early versions of KORO but we recognize that some things about IAO remain unsettled.

***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."[99] 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[97]. For KORO, we opted to include all four of these subclasses of 'Material Entity.' We did this because the definitions of 'Artifact' and

‘Information Artifact’ are more recent and more specific to our purposes[97]. 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 that exists between a **GENERICALLY DEPENDENT CONTINUANT** and a **SPECIFICALLY DEPENDENT CONTINTUANT**, 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 that exists 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

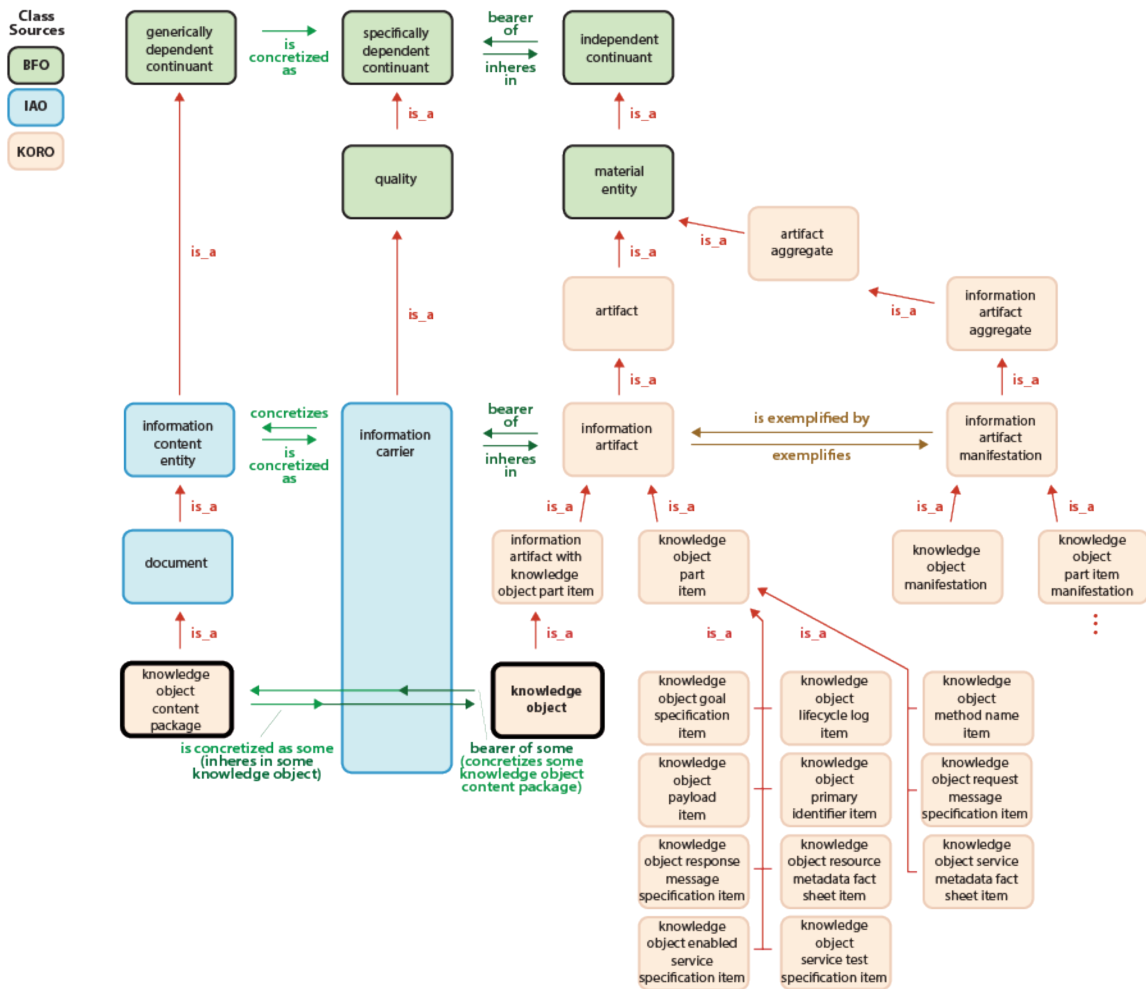
The two relationships defined immediately above have the following two corresponding inverse relationships that are also defined in BFO:

> CONCRETIZES = def. a relationship that exists 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 **INDEPEDENT CONTINUANT**
>
> BEARER OF = def. a relation that exists 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 3.7 below, the inverse relationships between the three subclasses of ‘Continuant’ are illustrated. Next, toward the middle of Figure 3.7, 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 3.7, through instances of ‘Information Carrier’, instances of ‘Knowledge Object Content Package’ are concretized and then made to inhere in instances of material

Figure 3.7 Portion of KORO Indicating the Whole and Parts of a Knowledge Object with Key Relationships



'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 3.7:

ARTIFACT AGGREGATE = def. a **MATERIAL ENTITY** that is made up of a collection of **ARTIFACTS** and whose parts are exactly exhausted by the **ARTIFACTS** that form this collection

INFORMATION ARTIFACT AGGREGATE = def. an **ARTIFACT AGGREGATE** that is made up of a collection of **INFORMATION ARTIFACTS** bearing concretizations of **INFORMATION CONTENT ENTITIES** and whose parts are exactly exhausted by the **INFORMATION ARTIFACTS** that form this collection

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** that is comprised of all instances in reality of some particular **INFORMATION ARTIFACT**

KNOWLEDGE OBJECT MANIFESTATION = def. an **INFORMATION ARTIFACT AGGREGATE** that is comprised of all instances in reality of a particular and uniquely identifiable **KNOWLEDGE OBJECT**

KNOWLEDGE OBJECT PART ITEM MANIFESTATION = def. an **INFORMATION ARTIFACT AGGREGATE** that is comprised of all instances in reality of a particular part item of a **KNOWELDGE OBJECT**

In addition, the following key terms for inverse relationships between 'Information Artifact' and 'Information Artifact Aggregate' are also portrayed in Figure 3.7:

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[75]. We find that the creators of IAO have also been influenced by FRBR[72]. 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 the term '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[100].

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 3.7, the class 'Knowledge Object Part Item' has 11 subclasses to account for this, one for each type of 'Knowledge Object Part Item' in KORO.

As Figure 3.7 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).

Finally, an example of an actual Knowledge Object, built according to an early version of the KORO specification, is shown on the next page in Figure 3.8. This simple example Knowledge Object is a concretization of a Knowledge Object Content Package document. That document has its five mandatory parts per KORO: Primary Identifier, Resource Metadata Fact Sheet, Service Specification, Payload, and Lifecycle Log. Its five parts are indicated by embedded section headers in Figure 3.8. These headers are not part of the Knowledge Object.

The Knowledge Object in Figure 3.8 holds in its payload a computable representation of a guideline about how to use codeine given an individual's phenotype for the gene CYP2D6. This Knowledge Object, which can be made to generate advice using the KGrid Activator, was built by Koki Sasagawa and George Meng in the Knowledge Grid lab in early 2018. Its payload is represented in software code written in the Python computer language.

Figure 3.8 Example of a Knowledge Object Built Using the KORO Specification

{"metadata":{"title":"CPIC CYP2D6 Codeine Recommendations",
PRIMARY IDENTIFIER
"arkId":"ark:/99999/fk4mc97w6m",
RESOURCE METADATA FACT SHEET
"version":"v0.1.0",
"owner":"KGRID",
"description":"",
"contributors":"Koki Sasagawa",
"keywords":"CPIC, CYP2D6, Codeine, Recommendation",
"published":true,"lastModified":1517610998731,
"createdOn":1517610546335,
SERVICE SPECIFICATION
"inputMessage":
<ot:noofparams>2</ot:noofparams>\n
<rdf:li>phenotype</rdf:li>\n
<rdf:li>choice</rdf:li>\n
<ot:datatype>STRING</ot:datatype>\n
"outputMessage":
<ot:returntype>STRING</ot:returntype>\n
"functionName":"execute",
"engineType":"Python"},
PAYLOAD
payload":{"content":"# KGrid CPIC guidelines UGT1A1 Phenotype to Recommendation
Payload\n# Koki Sasagawa \n# Last Updated: 2/2/2018
def execute(pheno):
The following function will return the recommendation
corresponding to the specified phenotype
Dictionary containing Phenotype to Recommendation Information\n\t
pheno_recom =
{*"Ultrarapid metabolizer\":*
{"Implications for phenotypic measures\": \
"Increased formation of morphine following codeine administration,
leading to higher risk of toxicity\","Dosing recommendations\": \"
Avoid codeine use due to potential for toxicity.\",
"Classification of recommendations\": \"Strong\",},
"Normal metabolizer\":
{"Implications for phenotypic measures\":\
"Normal morphine formation\","Dosing recommendations\": \
"Use label-recommended age- or weight-specific dosing.\",
"Classification of recommendations\": \"Strong\"},
"Intermediate metabolizer\":
{"Implications for phenotypic measures\":\
"Reduced morphine formation\","Dosing recommendations\": \
"Use label-recommended age- or weight-specific dosing.
If no response, consider alternative analgesics such as morphine or a nonopioid.\","
Classification of recommendations\": \"Moderate\",},
"Poor metabolizer\":
{"Implications for phenotypic measures\":\
"Greatly reduced morphine formation following codeine administration, leading
to insufficient pain relief\",\n\t\t\t\"Dosing recommendations\": \
"Avoid codeine use due to lack of efficacy.\",
"Classification of recommendations\": \"Strong\",}
options = {\"0\": \"Implications for phenotypic measures\", \"1\": \
"Dosing recommendations\", \"2\": \"Classification of recommendations\"}\n\n\t
recommendation = {\"Info\": \"\", \"Recom\": \"\"}
if pheno[\"choice\"]:
if pheno[\"choice\"] in options:
recommendation[\"Info\"] = options[pheno[\"choice\"]]else:
return (\"Incorrect/invalid option.\")
return str(recommendation[\"Info\"] + \": \" + recommendation[\"Recom\"])",
LIFECYCLE LOG
"logData":"ProvenanceLogData: \n\t
wasAttributedTo: 'George Meng',\n\t
wasGeneratedBy: 'https://dlhs-fedora-dev.med.umich.edu/
fcrepo/rest/tx:d6d045c8-1481-46ee-a4f3-3b4ca27ce113/99999-fk4mc97w6m/Log/
CreateActivity',\n\t wasAssociatedWith: 'George Meng',\n\t
used: 'https://dlhs-fedora-dev.med.umich.edu/
fcrepo/rest/tx:d6d045c8-1481-46ee-a4f3-3b4ca27ce113/99999-fk4mc97w6m/Log/
CreateActivity',\n\t startedAtTime: Fri Feb 02 17:29:05 EST 2018,\n\t
endedAtTime: Fri Feb 02 17:29:06 EST 2018",
"uri":"ark:/99999/fk4mc97w6m"}

***RQ3*. What questions about Knowledge Object parthood can an ontology answer competently in order to demonstrate its logic and potential utility?**

When given an instance of an arbitrary entity, described in the form of Web Ontology Language (OWL) axioms, KORO supports automated reasoning to infer things about the makeup and parts of the arbitrary entity. By so doing, it enables effective, automated 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 enables a machine to answer *CQ1* 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 automated 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 help answer *RQ4*, we have built a simple working example of an instance of 'Knowledge Object.' It is referred to from here on as KO1. KO1 has, as one of its necessary parts, an instance of 'Knowledge Object Payload Part Item.' It is referred to as payload-of-KO1. Payload-of-KO1 bears a concretization of a very simple mathematical model, which is an example based on a previously published lung cancer risk statistical predictive model[101].

For this simple example, required Knowledge Object parts for KO1 are listed in Table 3.1, column 1. Corresponding examples of the information content constituting the parts of KO1 are listed in Table 3.1, column 2.

Creators of Knowledge Objects have innumerable options for formatting the information content stored in their KOs. For the 'Information Content Entity' parts making up a KO, 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 KGrid Library component[87]?

3. How will the information in each KO part be serialized for use as a service-enabler by the KGrid Activator component[92], which allows its users to quickly stand up KO-based automated knowledge services as webservices?

Table 3.1 Implementation Decisions Made for KO1 as an Example Knowledge Object

REQUIRED PART	EXAMPLE CONTENTS	INFORMATION MODEL	LIBRARY SERIALIZATION for use as a RESOURCE	ACTIVATOR SERIALIZATION to enable a SERVICE
Knowledge Object KO1's Primary Identifier Item	ark:/99999/fk4jh3tk9s	Archival Resource Key[86,102]	As RDF† triple	As JSON◆
Payload-of-KO1	Risk score (%) = $V1^{0.4}+V2^{1.4}-V3^{2}$	Python 2.7	As binary file	As JSON
Knowledge Object KO1's Resource Metadata Fact Sheet	Title: Risk Model Description: Predictive Model Creator: A. Person	Dublin Core Metadata Set[103]	As RDF triples	As JSON
Knowledge Object KO1's 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 KO1's Lifecycle Log	Created on: 1-1-2017	PROV-O Ontology[104]	As RDF triples	

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

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 KO1, the answers to these 15 implementation questions are provided in the 3rd, 4th, and 5th columns of Table 3.1.

The results presented in Table 3.1 thus complete the reporting of results to address *RQ1* through *RQ4*. Summarizing, these results establish KORO as a formal, computable ontological model, represented in OWL format, that defines what a Knowledge Object is and describes its parts and their mereology.

Overview of the system architecture of the KGrid Library

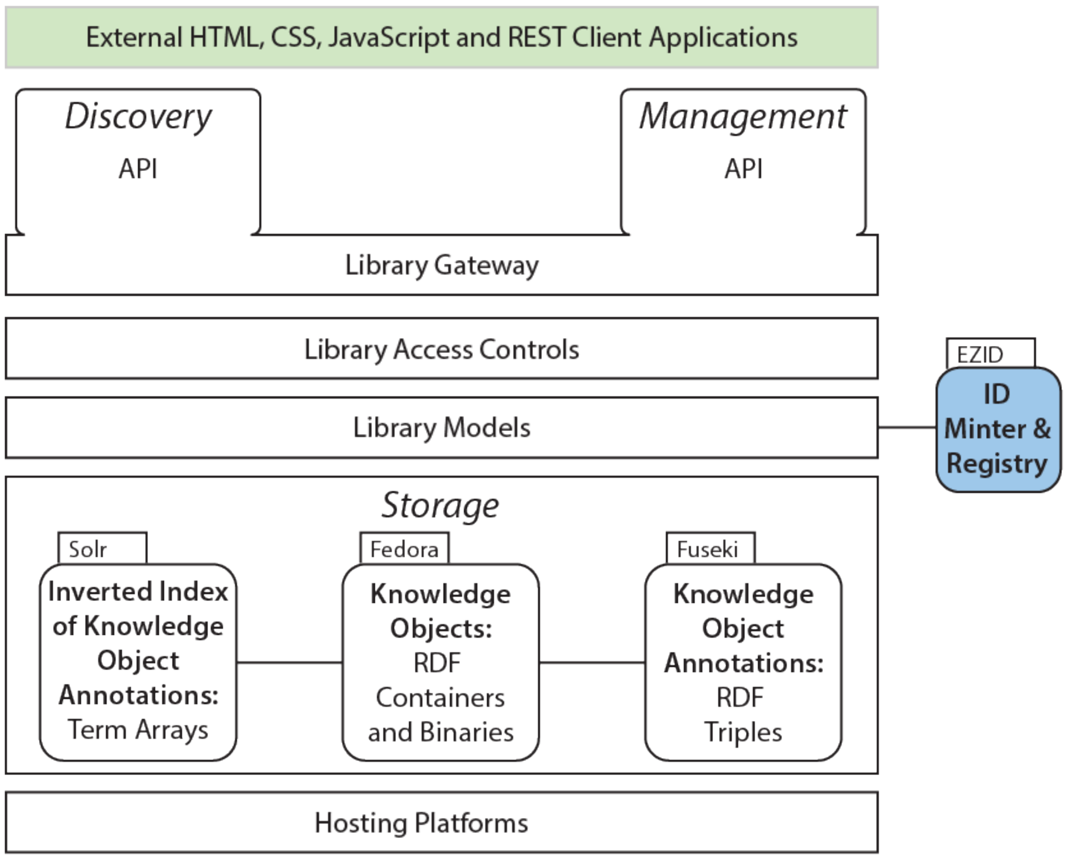
The results reported next address *RQ5*, which is:

***RQ5.* What are a minimum set of necessary and sufficient technical components, which, when integrated, can bring about a stable, scalable *digital library* capable of storing Knowledge Objects, making them discoverable and shareable, facilitating their incorporation into various other applications, and supporting both Knowledge Object and *digital library* management?**

A library must provide a set of core functions[105]. These core functions can be organized into three high-level categories: (1) Storage of holdings, (2) Discovery of holdings, and (3) Management of both holdings and the library itself. A successful digital library adequately addresses all three of these core function categories. A system architecture for a conventional digital library platform holding computable Knowledge Objects was designed with these three functional categories in mind (Figure 3.9). What Figure 3.9 below portrays is the system architecture for the KGrid Library.

What follows is a description of the KGrid Library's current capabilities to curate computable Knowledge Objects. The capabilities described here have all been realized during the initial development phase of the software for the KGrid Library, through which the technical

Figure 3.9 System Architecture for the Digital KGrid Library



system architecture portrayed in Figure 3.9 has been realized. All of these capabilities rely on established World Wide Web hosting platforms, making the KGrid Library a native online, network-ready conventional digital library technology.

Storage. Knowledge objects held by an instance of the KGrid Library are stored in an instance of Fedora version 4.x using a tree structure comprised of (a) RDF Containers, (b) Binary Containers, and (c) the branches between them. In Fedora 4.x, RDF Containers hold each object's metadata annotations and links to evidence and licenses while Binary Containers hold the executable software payloads of each object.

Digital preservation is supported by Fedora's native fixity checking. This feature checks the binary files stored in Fedora 4.x to detect digital file alteration or corruption. When alternations or corruption are detected, Fedora 4.x can alert its users to restore or fix a binary file.

It is possible to query Fedora directly through Fedora's own Application Programming Interface (API). This capability supports integration of the storage sub-components that are included in this architecture. RDF Triples are automatically extracted from Fedora and loaded into a separate Fuseki "triple store"[106]. Fuseki is a technology that enables SPARQL queries to be executed on the metadata describing Knowledge Objects. An additional component, which also stores RDF metadata coming from Fedora, is Apache Solr[107]. Solr supports finding and discovering Knowledge Objects in the KGrid Library by automatically generating an inverse index from Knowledge Object annotation metadata.

Discovery. Besides its integrated Storage components described immediately above, the KGrid Library's system architecture includes several other software components that we have built to support discovery, or findability, of Knowledge Objects.

The first of these is the Library Models component, which is used to validate that Knowledge Objects have all of their required parts according to a predefined model of Knowledge Objects, namely KORO. The Library Models component also applies a specific provenance model – the W3C PROV-O ontology[104] – to consistently author RDF statements that record key events in the lifecycle of Knowledge Objects.

A Knowledge Object specific metadata model is used by the Library Models component to enable users to create, read, update, and delete Knowledge Object annotations. Fields used for these annotations include several Dublin Core elements[103], along with links to citations of published scientific works, and links to Knowledge Object licenses.

An external system, EZID[85], is used by the Library Models component to request and receive unique Archival Resource Key[102] (ARK) IDs for every Knowledge Object added to the holdings in an instance of the KGrid Library, and to register the whereabouts of each Knowledge

Object on an external EZID server. This enables EZID, working in conjunction with the Name-to-Thing Resolver, to become a persistent locator service for Knowledge Objects, allowing them to be easily found and accessed over the World Wide Web.

Another KGrid Library subcomponent that we have built is the Library Access Controls component. It is used to control access to the holdings stored in an instance of the KGrid Library.

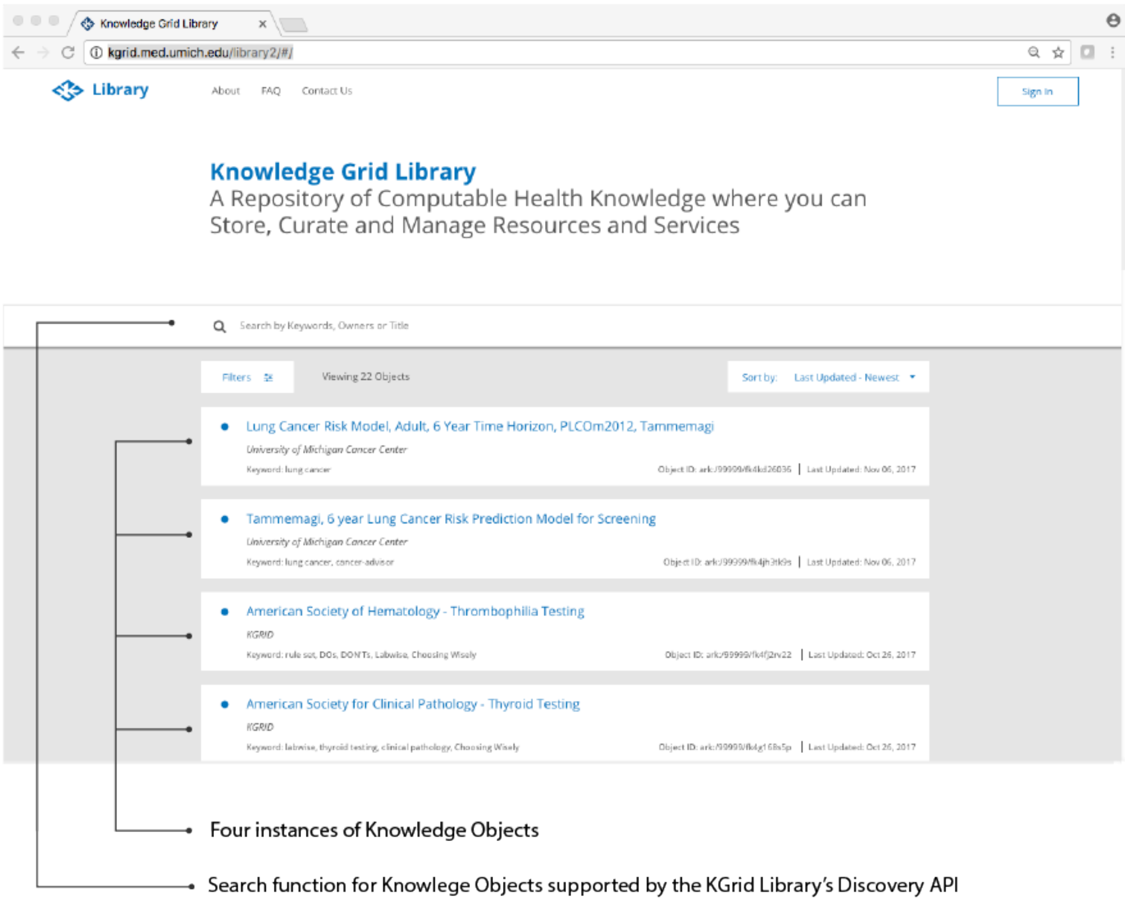
In addition, as subcomponents of the KGrid Library's system architecture we have also built a Library Gateway component with two new APIs, one enabling external systems to do Knowledge Object discovery (the Discovery API) and another to let external systems import, export, and update Knowledge Objects (the Management API). The Library Gateway implements these two APIs by making them available for use by external systems.

Management. In the KGrid Library system architecture (Figure 3.9), the Management API applies mainly to Knowledge Object management and less to overall KGrid Library management. The basic Knowledge Object Management capabilities supported by the Management API consist of mechanisms enabling external systems to (1) create/upload Knowledge Objects into the holdings of an instance of the KGrid Library, (2) "checkout" or read/download holdings from an instance of the KGrid Library, (3) edit/update Knowledge Objects held within an instance of the KGrid Library, and (4) delete or remove Knowledge Objects from the holdings in an instance of the KGrid Library.

In addition, the Library Models component includes logic to govern all four of the mechanisms supported by the Management API listed above. This logic enables user configurable options for configuring access to, and editing of, Knowledge Objects held in an instance of the digital KGrid Library.

A development instance of the KGrid Library, built according to the system architecture described here and portrayed in Figure 3.9, is publically available online at: http://kgrid.med.umich.edu/library2/. An image of the KGrid Library's current user interface, or "front-end", which has been built primary using HTML and CSS, is shown below in Figure 3.10.

Figure 3.10 Image of an Instance of the KGrid Library's Online User Interface



Overview of the system architecture of the KGrid Activator

The results reported next address *RQ6*, which is:

RQ6. **What constitutes a minimum set of necessary and sufficient technical components that, when integrated, result in stable, scalable, rapidly deployable, secure, traceable, *activated* Knowledge Objects, thereby enabling KGrid to provide performant and reliable on-demand knowledge services for health?**

To make instances of computable knowledge active, i.e., to activate them, means to enable them to be accessed and applied directly via a service interaction provided by an API. The purpose of the KGrid Activator is to bring about such API-enabled advice-giving service interactions by using the payloads of Knowledge Objects. This means the Activator is a tool for quickly establishing performant, on-demand, online advice-giving service capabilities as part of the Knowledge Grid. For this reason, the answer to *RQ6* is given next by describing the system architecture of the current KGrid Activator.

Minimum Functional Requirements for the KGrid Activator. The two fundamental service functions of the KGrid Activator are (1) *to transfer* computable knowledge resources wholesale upon request to other systems and (2) *to apply* computable knowledge, as part of performing an automated advice-giving service, by accepting relevant instance data as input, combining those instance data with computable knowledge, performing an execution step, and returning a computed result as new advice to inform an unmade decision. The KGrid Activator can perform both of these fundamental functions.

In addition, in our work with the KGrid Activator we are evolving a specific model of computable biomedical knowledge *activation*. This emerging model of *activation* stands to help us further extend beyond the two fundamental KGrid Activator functions above by including a host of other functions we believe are minimally necessary to deploy computable biomedical knowledge safely, effectively, and efficiently to generate advice for improving health. These other activation-related functions include, but are not limited to, *access control*, *Knowledge Object fixity checking*, *activation policy setting*, *activation policy enforcement*, and *Knowledge Object utilization logging*.

A system architecture for the KGrid Activator, which is capable of *activating* the payloads of Knowledge Objects, has been designed with these functions, in mind (Figure 3.11).

Payload Agnostic Activator Functions. The KGrid Activator provides access controls for all users. It also provides several functions for "Knowledge Management Administrators." These functions are general knowledge management functions that apply in the same way to all

Knowledge Objects. Hence, these functions are payload agnostic. These payload agnostic KGrid Activator functions are provided via the four APIs depicted to the left of the vertical dashed orange line in Figure 3.11 below.

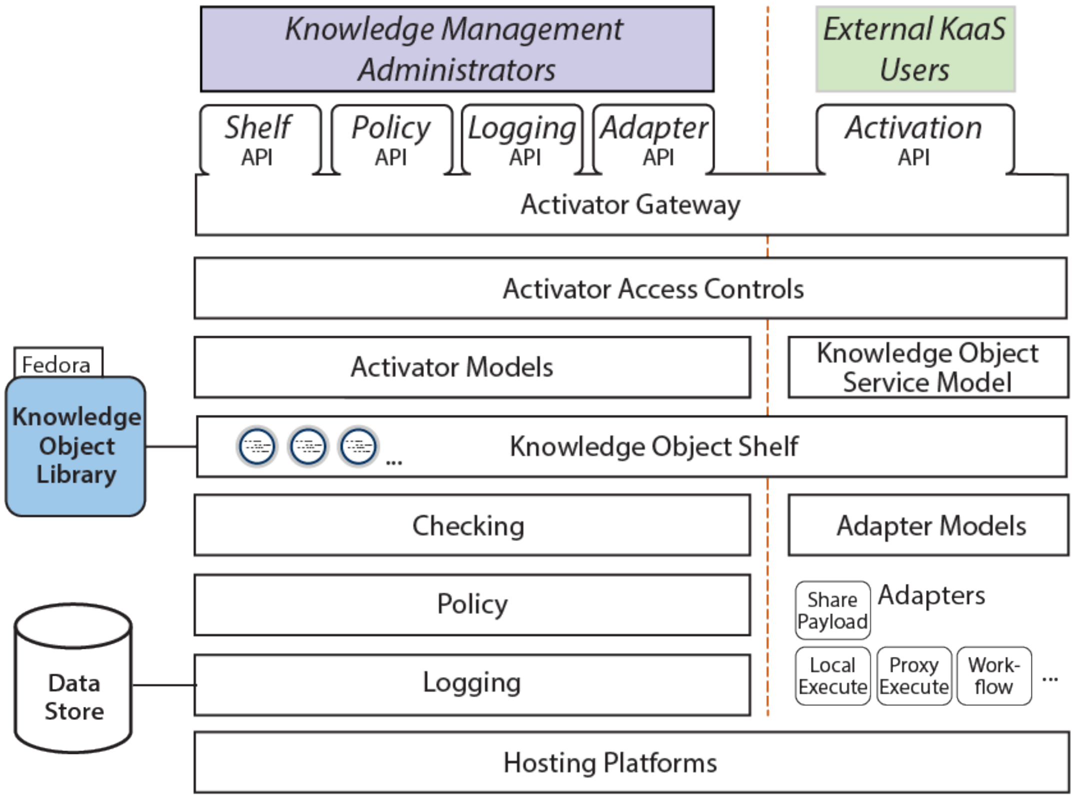
The KGrid Activator's "Shelf API" enables users to load and unload Knowledge Objects into and out of an instance of the KGrid Activator, respectively. The KGrid Activator maintains a persistent shelf for storing Knowledge Objects with which it interacts. The KGrid Activator shelf is currently implemented very simply as a dedicated file folder. The Shelf API allows KGrid Activator users to inspect the Knowledge Objects currently loaded on an instance of the KGrid Activator by reporting the contents of its shelf to a user upon request.

The KGrid Activator's "Policy API" enables users to set, edit, remove, and apply policies that either govern aspects loading Knowledge Objects onto a shelf or serve to control the services that are provided via the KGrid Activator. For example, the Policy API can be used to set and then enforce a rule that only Knowledge Objects from certain, predefined and trusted sources can be loaded and used by a particular instance of the KGrid Activator.

The KGrid Activator's "Logging API" provides users with a mechanism to collect data about automated knowledge service utilization. If logging is turned on within the KGrid Activator, a new data stream is implemented and produced. As a result, log data are created and stored outside of the KGrid Activator in a separate data store. From there these log data can be analyzed and studied to improve performance of the KGrid Activator, or for other reasons.

The KGrid Activator's "Adapter API" makes the KGrid Activator extensible with regard to its ability to execute source code in various languages. Here, an "Adapter" is a software plug-in that enables the KGrid Activator to participate effectively in desired knowledge and advice-giving *service interactions* to fulfill its two fundamental functions outlined above. To date, several "Local Execute" Adapters have been written enabling the KGrid Activator to execute source code written in Python, Java, Javascript, and R.

Figure 3.11 System Architecture for the KGrid Activator



More specifically, the Adapter API allows users to plug one or more Adapters in to a deployed instance of the KGrid Activator. Four specific types of Adapters are shown in Figure 3.8. The simplest type is the "Share Payload Adapter". It sends the payload of a Knowledge Object out to an external system upon request. In contrast, "Local Execute Adapters" are software language-specific ones that, when they are plugged-in, enable the KGrid Activator to execute Python, JavaScript, and other types of software code found in Knowledge Object payloads. In a similar way, "Proxy Execute Adapters" are a type of Adapter capable of engaging external services to execute source code written in various software languages. Finally, within the system architecture of the KGrid Activator, "Workflow Adapters" provide a mechanism to orchestrate execution of the multiple Knowledge Object payloads in repeatable ways.

One set of functions that are not controlled by an API are the internal "Checking" functions of the KGrid Activator. For safety and security, the KGrid Activator has capabilities to check the integrity of any Knowledge Object on its shelf, using various fixity and checksum software routines. The KGrid Activator can also check that incoming instance data are a match for the specifications of any given Knowledge Object's individual instance data.

Payload Specific Activator Functions. The KGrid Activator's Knowledge Object enabled, payload-specific capabilities are made available via its "Activation API", which is directly supported by a series of technical components that are depicted to the right of the dashed orange line in Figure 3.11.

When a standard HTTP request (i.e., GET, PUT, POST, etc.) is received by the Activation API, that request is handled by the Knowledge Object Service Model layer of the system architecture (Figure 3.11). If the needed Knowledge Object is already on the KGrid Activator's Shelf, and the needed Adapter to provide the requested service is already plugged into the KGrid Activator, then the Checking functions are performed and, if all checks are passed, the request for service is fulfilled.

If the needed Knowledge Object is not on the Shelf, then the KGrid Activator can automatically initiate a search for the Knowledge Object at the KGrid Library to which it is connected (Three Knowledge Objects are depicted on the Shelf in Figure 3.8 but many more can be added.) The Activation API also provides a set of standard error responses that describe anticipated technical problems in ways that support troubleshooting whenever a service request cannot be fulfilled by the KGrid Activator.

A development instance of the KGrid Activator, built according to the system architecture portrayed in Figure 3.11, is publically available online at: http://kgrid.med.umich.edu/stack2/shelf. Clicking on this link will open a browser window which shows the contents of the shelf for the development KGrid Activator instance serialized as human readable, but technically oriented, text in JSON format.

DISCUSSION

Today, with the three core technical components of the Knowledge Grid reviewed in this chapter, it is possible to create Knowledge Objects, place them in a collection held by a deployed instance of the KGrid Library, load them onto the shelf of a deployed instance of the KGrid Activator, and engage their payloads as advice-giving services using common methods for API-enabled webservices that conform to the RESTful architectural style.[89,108].

A small number of approximately 70 trial Knowledge Objects have been created and tested thus far. Text editors have been used to develop each component of these objects in a part-by-part manner, including their payloads, metadata, and service specifications. These first Knowledge Objects made it possible for others to receive and use computable knowledge resources through simple mechanisms of file transfer.

Insight gained from what we have learned by creating Knowledge Objects part by part raises the possibility of fully automating the process of creating Knowledge Objects. We are using these insights to conduct some early experiments to create Knowledge Objects automatically. The results of one of these experiments is reported in Chapter 4.

Currently, the KGrid Activator has the potential to augment the interoperability of computable biomedical knowledge held in Knowledge Objects in a variety of ways. These ways will be explored in depth in Chapter 4.

The first real-world test of the KGrid Activator API was passed in 2017 when a Health Information Exchange organization[109] connected an instance of the KGrid Activator to its existing internal information technology infrastructure and successfully engaged the KGrid Activator to identify predefined patterns in its electronic prescription data. The results of this first test have not yet been published.

Meanwhile, a body of future work is planned for the Knowledge Grid. The KGrid Library software is currently being upgraded so that Knowledge Objects can be organized into collections within the KGrid Library. In addition, this software is being prepared to support the first deployment of a production instance of the KGrid Library. Studies of initial user responses to the KGrid Library are now being planned.

Limitations and Next Steps

The technical research and development work covered in this chapter is limited. It is work that has resulted in a fledgling knowledge infrastructure and certainly not a robust or proven one. The Knowledge Grid is formative. Time will tell if any or all of the core technical components of the Knowledge Grid will come into significant real-world use to support Learning Health Systems. Meanwhile, work presented in the following chapter tests some of the ideas surfaced by doing research and development work on the Knowledge Grid to see whether it has the potential to augment the interoperability of computable biomedical knowledge in new and helpful ways.

CONCLUSION

Early versions of three core technical components of the Knowledge Grid knowledge infrastructure have been designed, developed, and tested. These are Knowledge Objects, the KGrid Library and the KGrid Activator. When used together these core components support the curation of computable knowledge resources. These components also support highly individuated, interoperable knowledge services and advice-giving services enabled by the payloads embedded in Knowledge Objects.

As it stands, the Knowledge Grid is a working prototype. The final chapter of this dissertation presents work we have completed to demonstrate that the Knowledge Grid has the potential to increase the interoperability of computable knowledge in a manner that also increases the interoperability of knowledge services and advice-giving services, ultimately resulting in a capability to scale up production of well-informed health advice[3,110].

References

1. Edwards, P. N. *et al. Knowledge infrastructures: Intellectual frameworks and research challenges*. (2013).
2. Lagoze, C. & Van de Sompel, H. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. *Proc. first ACM/IEEE-CS Jt. Conf. Digit. Libr. - JCDL '01* 54–62 (2001). doi:10.1145/379437.379449
3. Friedman, C. P., Rubin, J. C. & Sullivan, K. J. Toward an Information Infrastructure for Global Health Improvement. *IMIA Yearb. Med. Informatics* 1–8 (2017).
4. Chambers, D. A., Feero, W. G. & Khoury, M. J. Convergence of implementation science, precision medicine, and the learning health care system: A new model for biomedical research. *Jama* **315,** 1941–1942 (2016).
5. Etheredge, L. M. A rapid-learning health system. *Health Aff.* **26,** (2007).
6. Greene, S. M., Reid, R. J. & Larson, E. B. Implementing the learning health system: From concept to action. *Ann. Intern. Med.* **157,** 207–210 (2012).
7. Friedman, C. & Rigby, M. Conceptualising and creating a global learning health system. *Int. J. Med. Inform.* **82,** (2013).
8. Hauser, S. E., Demner-Fushman, D., Ford, G. & Thoma, G. R. A testbed system for mobile point-of-care information delivery. *Proceedings. 17th IEEE Symp. Comput. Med. Syst.* 147–152 (2004). doi:10.1109/CBMS.2004.1311706
9. Frew, J. *et al.* The Alexandria digital library architecture. *Int. J. Digit. Libr.* **2,** 259–268 (2000).
10. Kahn, R. & Wilensky, R. A framework for distributed digital object services. *Int. J. Digit. Libr.* **6,** 115–123 (2006).
11. Lagoze, C. J. Lost identity: The assimilation of digital libraries into the web. *Diss. Abstr. Int. Sect. A Humanit. Soc. Sci.* **71,** 755 (2010).
12. Plantin, J.-C., Lagoze, C., Edwards, P. N. & Sandvig, C. Infrastructure studies meet platform studies in the age of Google and Facebook. *New Media Soc.* 146144481666155 (2016). doi:10.1177/1461444816661553
13. Weinberger, D. The Machine that would Predict the Future. *Sci. Am.* **305,** 52–57 (2011).
14. Plantin, J.-C., Lagoze, C. & Edwards, P. N. Re-integrating scholarly infrastructure: The

ambiguous role of data sharing platforms. 1–14 (2018). doi:10.1177/2053951718756683

15. Ram, K. Git can facilitate greater reproducibility and increased transparency in science. *Source Code Biol. Med.* **8,** 1–8 (2013).
16. Morris, Z. S., Wooding, S. & Grant, J. The answer is 17 years, what is the question: understanding time lags in translational research. *J. R. Soc. Med.* **104,** 510–520 (2011).
17. McGlynn, E. A. *et al.* Quality of health care delivered to adults in the United States. *N Engl J Med* **348,** 2635–2645 (2003).
18. González-González, A. Information needs and information-seeking behavior of primary care physicians. *Ann. Fam. ...* 345–352 (2007). doi:10.1370/afm.681.Jos
19. Pauker, S. G., Gorry, G. A., Kassirer, J. P. & Schwartz, W. B. Towards the simulation of clinical cognition. Taking a present illness by computer. *Am. J. Med.* **60,** 981–96 (1976).
20. Covell, D. G., Uman, G. C. & Manning, P. R. Information needs in office practice: are they being met? *Ann. Intern. Med.* **103,** 596–9 (1985).
21. Beasley, J. W. *et al.* Information Chaos in Primary Care: Implications for Physician Performance and Patient Safety. *J Am Board Fam Med* **24,** 745–751 (2011).
22. Markman, M. Information overload in oncology practice and its potential negative impact on the delivery of optimal patient care. *Curr. Oncol. Rep.* **13,** 249–51 (2011).
23. Lesar, T. S., Briceland, L. & Stein, D. aniel S. Factors related to errors in medication prescribing. *JAMA* **277,** 312–317 (1997).
24. Leape, L. L. Preventing adverse drug events. *Am J Heal. Syst Pharm* **52,** 379–382 (1995).
25. Friedman, C. P. *et al.* The science of Learning Health Systems: Foundations for a new journal. *Learn. Heal. Syst.* **1,** e10020 (2017).
26. Olsen, L., Aisner, D., McGinnis, J. M., Editors & Medicine, R. on E. *The learning healthcare system: workshop summary*. (2007).
27. Friedman, C. *et al.* Toward a science of learning systems: a research agenda for the high-functioning Learning Health System. *J. Am. Med. Inform. Assoc.* **22,** 43–50 (2015).
28. Fleurence, R. L. *et al.* Launching PCORnet, a national patient-centered clinical research network. *J. Am. Med. Informatics Assoc.* **21,** 578–582 (2014).
29. Hripcsak, G. *et al.* Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers. *Stud. Health Technol. Inform.* **216,** 574–578 (2015).

30. Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K. Y. & Kitano, H. Software for systems biology: From tools to integrated platforms. *Nat. Rev. Genet.* **12,** 821–832 (2011).
31. Azencott, C. A. *et al.* The inconvenience of data of convenience: Computational research beyond post-mortem analyses. *Nat. Methods* **14,** 937–938 (2017).
32. Montie, J. E., Linsell, S. M. & Miller, D. C. Quality of care in urology and the Michigan Urological Surgery Improvement Collaborative. *Urol. Pract.* **1,** 74–78 (2014).
33. Westfall, J. M., Mold, J. & Fagnan, L. Practice-based research - 'Blue highways' on the NIH roadmap. *J. Am. Med. Assoc.* **297,** 403–406 (2007).
34. Frankovich, J., Longhurst, C. A. & Sutherland, S. M. Evidence-based medicine in the EMR era. *N Engl J Med* **365,** 1758–1759 (2011).
35. Powell, A. C., Landman, A. B. & Bates, D. W. In search of a few good apps. *JAMA - J. Am. Med. Assoc.* **311,** 1851–1852 (2014).
36. Wicks, P. *et al.* Sharing health data for better outcomes on patientslikeme. *J. Med. Internet Res.* **12,** 1–12 (2010).
37. Susman, G. I. & Evered, R. D. An Assessment of the Scientific Merits of Action Research. *Adm. Sci. Q.* **23,** 582–603 (1978).
38. Moen, R. & Norman, C. Evolution of the PDCA Cycle. *Society* 1–11 (2009).
39. MEDLINE®/PubMed® Baseline Database Distribution: File Names, Record Counts, and File Size. *U.S. National Library of Medicine Website* (2016). Available at: https://www.nlm.nih.gov/bsd/licensee/2016_stats/baseline_med_filecount.html. (Accessed: 30th December 2017)
40. Flynn, A., Shi, W., Fischer, R. & Friedman, C. P. Digital Knowledge Objects and Digital Knowledge Object Clusters: Unit Holdings in a Learning Health System Knowledge Repository. *Hawaii Int. Conf. Syst. Sci. 2016* (2016).
41. Relling, M. & Klein, T. CPIC: Clinical Pharmacogenetics Implementation Consortium of the Pharmacogenomics Research Network. *Nature* **89,** 464–467 (2011).
42. Maniatis, P., Roussopoulos, M., Giuli, T. J., Rosenthal, D. S. H. & Baker, M. The LOCKSS peer-to-peer digital preservation system. *ACM Trans. Comput. Syst.* **23,** 2–50 (2005).
43. Moulaison Sandy, H. & Corrado, E. M. Bringing Content into the Picture: Proposing a Tri-Partite Model for Digital Preservation. *J. Libr. Adm.* **58,** 1–17 (2018).

44. Bowker, G. C. *Science on the Run, Information Management and Industrial Geophysics at Schlumberger, 1920-1940*. *Inside Technology* (1994). doi:10.1177/016224399502000408
45. Grolinger, K., Capretz, M. A. M., Mezghani, E. & Exposito, E. Knowledge as a Service Framework for Disaster Data Management. *2013 Work. Enabling Technol. Infrastruct. Collab. Enterp.* 313–318 (2013). doi:10.1109/WETICE.2013.48
46. Abdullah, R., Eri, Z. D. & Talib, A. M. A model of knowledge management system for facilitating knowledge as a service (KaaS) in cloud computing environment. *2011 Int. Conf. Res. Innov. Inf. Syst.* 1–4 (2011). doi:10.1109/ICRIIS.2011.6125691
47. Lagoze, C. & Davis, J. R. Dienst - An Architecture for Distributed Document Libraries. *Commun. ACM* **38,** 47 (1995).
48. Payette, S. & Lagoze, C. Flexible and Extensible Digital Object and Repository Architecture (FEDORA). *Proc. Second Eur. Conf. Res. Adv. Technol. Digit. Libr. Lect. Notes Comput. Sci.* **1513,** 41–59 (1998).
49. Polsani, P. R. Use and Abuse of Reusable Learning Objects. *J. Digit. Inf.* **3,** 2–7 (2003).
50. Brugger, N. Website history and the website as an object of study. *New Media Soc.* **11,** 115–132 (2009).
51. Sinclair, J., Joy, M., Yau, J. Y.-K. & Hagan, S. A Practice-Oriented Review of Learning Objects. *IEEE Trans. Learn. Technol.* **6,** 177–192 (2013).
52. Haag, J., Poltrack, J., Hruska, N. & Johnson, A. The Next Generation of SCORM : Innovation for the Global Force. *Interservice/Industry Training, Simulation, Educ. Conf. 2012* 1–9 (2012).
53. Bechhofer, S., De Roure, D., Gamble, M., Goble, C. & Buchan, I. Research Objects: Towards Exchange and Reuse of Digital Knowledge. *Nat. Preced.* (2010). doi:10.1038/npre.2010.4626
54. Corcho, O. *et al.* Research Object ontology. *Worflow 4Ever* (2017). Available at: http://wf4ever.github.io/ro/2016-01-28/. (Accessed: 31st December 2017)
55. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3,** 160018 (2016).
56. Weinstein, P. & Alloway, G. Seed ontologies. *Proc. Second ACM Int. Conf. Digit. Libr. - DL '97* 83–91 (1997). doi:10.1145/263690.263799
57. Fensel, D., Horrocks, I., Harmelen, F. Van & Mcguinness, D. OIL Ontology Infrastructure

to Enable the Semantic Web. *Intell. Syst. IEEE* **16,** 38–45 (2001).

58. Guha, R. V., Brickley, D. & MacBeth, S. Schema.org: Evolution of Structured Data on the Web. *Queue* **13,** 10–37 (2015).
59. Arp, R., Smith, B. & Spear, A. D. *Building ontologies with basic formal ontology.* (MIT Press, 2015).
60. Noy, N. F. & McGuinness, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowl. Syst. Lab.* 25 (2001). doi:10.1016/j.artmed.2004.01.014
61. Gennari, J. H. *et al.* The evolution of Protégé: An environment for knowledge-based systems development. *Int. J. Hum. Comput. Stud.* **58,** 89–123 (2003).
62. Drummond, N., Rector, A. & Stevens, R. Sequences in Protégé OWL. *9th Intl. Protégé Conf.* 3–6 (2006).
63. Musen, M. A. *et al.* The National center for biomedical ontology. *J. Am. Med. Informatics Assoc.* **19,** 190–195 (2012).
64. Smith, B. *et al.* The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* **25,** 1251–1255 (2007).
65. Whetzel, P. L. *et al.* BioPortal: Ontologies and integrated data resources at the click of a mouse. *CEUR Workshop Proc.* **833,** 292–293 (2011).
66. Sill, A. The Design and Architecture of Microservices. *IEEE Cloud Comput.* **3,** 76–80 (2016).
67. Rosenbaum, S. & Painter, M. W. *Assessing Legal Implications of Using Health Data to Improve Health Care Quality and Eliminate Health Disparities*. (2005).
68. U.S. Food and Drug Administration. Clinical and Patient Decision Support Software: Draft Guidance for Industry and Food and Drug Adminstration Staff. 1–13 (2017).
69. Neuss, M. N. *et al.* 2016 Updated American Society of Clinical Oncology/Oncology Nursing Society Chemotherapy Administration Safety Standards, Including Standards for Pediatric Oncology. *J. Oncol. Pract.* **12,** 1262–1271 (2016).
70. Malone, J. *et al.* The Software Ontology (SWO): A resource for reproducibility in biomedical data analysis, curation and digital preservation. *J. Biomed. Semantics* **5,** 1–13 (2014).
71. Smith, B., Kumar, A. & Bittner, T. Basic Formal Ontology for Bioinformatics. *J. Inf. Syst.* 1–16 (2005).

72. Goldstein, A. *et al.* Information Artifact Ontology IAO. *Version 2017-03-27* Available at: https://github.com/information-artifact-ontology/IAO. (Accessed: 28th December 2017)
73. Bock, C. *et al.* OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. *Syntax* (2008). doi:urn:nbn:nl:ui:29-303961
74. Courtot, M. *et al.* MIREOT: The minimum information to reference an external ontology term. *Appl. Ontol.* **6,** 23–33 (2011).
75. Tillett, B. *Functional requirements of bibliographic records (FRBR): final report.* (1998).
76. Ren, Y. *et al.* Towards competency question-driven ontology authoring. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **8465 LNCS,** 752–767 (2014).
77. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A. & Katz, Y. Pellet: A practical OWL-DL reasoner. *Web Semant.* **5,** 51–53 (2007).
78. Lagoze, C., Payette, S., Shin, E. & Wilper, C. Fedora: an architechture for complex objects and their relationships. *Int. J. Digit. Libr.* **6,** 124–138 (2006).
79. Wilcox, D. Stewarding Research Data with Fedora. in *IFLA WLIC 2017* 1–4 (2017).
80. Foundation, I. Islandora. (2016). Available at: www.islandora.ca.
81. Yeh, S.-T., Reyes, F., Rynhart, J. & Bain, P. Deploying Islandora as a Digital Repository Platform: a Multifaceted Experience at the University of Denver Libraries. *D-Lib Mag.* **22,** (2016).
82. Awre, C. & Green, R. From Hydra to Samvera: an open source community journey. *Insights UKSG J.* **30,** 82–88 (2017).
83. Murphy, T. & Ummerpillai, H. The Architecture of Data Science and Archiving - Archonnex Architecture and Technology Stack. in *IASSIST 2016* (2016).
84. Murphy, T. & Ummerpillai, H. Archonnex - Digital Asset Management Systems Architecture. (2016).
85. Kunze, J., Janée, G. & Starr, J. EZID: Easy identifier and metadata management. in *DCMI 15 - Proceedings of the International Conference on Dublin Core and Metadata* (2015).
86. Kunze, J. & Rodgers, R. *The ARK Identifier Scheme*. *CDL Staff Publications* (2008).
87. Flynn, A. J. *et al.* Architecture and Initial Development of a Digital Library Platform for Computable Knowledge Objects for Health. *Stud. Health Technol. Inform.* **235,** 496–500 (2017).

88. Ryan, A. & Eklund, P. W. The Health Service Bus an architecture and case study in achieving interop. **2010,** 922–926 (2010).

89. Fielding, R. T. Architectural Styles and the Design of Network-based Software Architectures. *Doctoral Dissertation* **54,** (2000).

90. Johnson, R. *et al.* Spring Framework Reference Documentation. *Spring Framew. Version 3.2.17 Release* 819 (2013).

91. Crockford, D. Introducing JSON. *JSON: Javascript object notation* (2006). Available at: http://www.json.org/. (Accessed: 28th December 2017)

92. Flynn, A. J. *et al. Architecture and Initial Development of a Knowledge-as-a-Service Activator for Computable Knowledge Objects for Health*. (2018).

93. Carstensen, P. H. & Schmidt, K. Computer Supported Cooperative Work: New Challenges to Systems Design. *K. Itoh (Ed.), Handb. Hum. Factors* 619--636 (1999). doi:10.1.1.43.5157

94. Ackerman, M. S., Dachtera, J., Pipek, V. & Wulf, V. Sharing Knowledge and Expertise : The CSCW View of Knowledge Management. *Comput. Support. Coop. Work* **22,** 531–573 (2013).

95. Floridi, L. Semantic information and the network theory of account. *Synthese* **184,** 431–454 (2012).

96. 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).

97. Smith, B. & Ceusters, W. Aboutness: Towards Foundations for the Information Artifact Ontology. in *International Conference on Biomedical Ontology* 47–51 (2015).

98. Floridi, L. Semantic Information and the Correctness Theory of Truth. *Erkenntris* **74,** 147–175 (2011).

99. Smith, B. & Grenon, P. BFO Basic Formal Ontology. *Institute for Formal Ontology and Medical Information Science (IFOMIS) at Unversität des Saarlandes* (2017). Available at: http://ifomis.uni-saarland.de/bfo/. (Accessed: 28th December 2017)

100. Peroni, S. & Shotton, D. FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. *J. Web Semant.* **17,** 33–43 (2012).

101. Tammemägi, M. C. *et al.* Selection Criteria for Lung-Cancer Screening. *N. Engl. J. Med.*

368, 728–736 (2013).

102. Kunze, J. Towards electronic persistence using ARK identifiers. *ECDL Web Arch. Work.* 1–9 (2003).

103. Weibel, S. The Dublin Core: A Simple Content Description Model for Electronic Resources. *Bull. Assoc. Inf. Sci. Technol.* (1997). doi:10.1002/bult.70

104. Missier, P., Belhajjame, K. & Cheney, J. The W3C PROV family of specifications for modelling provenance metadata. *Proc. 16th Int. Conf. Extending Database Technol. - EDBT '13* 773 (2013). doi:10.1145/2452376.2452478

105. *No brief candle: Reconceiving research libraries for the 21st century*. (2008). doi:10.2307/805220

106. Apache Jena Fuseki 2 Documentation. (2017). Available at: https://jena.apache.org/documentation/fuseki2/. (Accessed: 30th December 2017)

107. The Apache Software Foundation. *Apache Solr Reference Guide Covering Apache Solr 7.2*. (2017).

108. O'Reilly, T. What is Web 2.0. (2005). Available at: http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html. (Accessed: 1st January 2018)

109. MiHIN Shared Services. *Michigan Health Information Network Shared Services* (2018). Available at: https://mihin.org/. (Accessed: 1st January 2018)

110. Friedman, C. *et al.* Toward a science of learning systems: a research agenda for the high-functioning Learning Health System. *J. Am. Med. Informatics Assoc.* 43–50 (2014). doi:10.1136/amiajnl-2014-002977

CHAPTER IV

ScriptNumerate: Study of a Promising Modularized Data-to-Advice Pipeline

INTRODUCTION

This final chapter describes work to achieve the ultimate specific aim of this dissertation. It shows, using a particular case example, how augmenting the interoperability of computable knowledge enables modular advice-giving services. Modular advice-giving services move us a step closer to the primary goal of finding better methods to produce large quantities of well-informed advice for health.

Chapter I conceptualized messages of advice as information objects targeted at unmade decisions. Here we are concerned with advice, and especially with advice about human health. We want all such advice to be well-informed and to communicate the least possible amount of uncertainty.

Chapter II surfaced several things that limit the interoperability of computable knowledge. These things are (1) a lack of modularization, (2) a lack of unique identifiers for instances of computable knowledge and related services, and (3) a lack of mechanisms to keep computable knowledge and related services up to date.

Chapter III introduced the Knowledge Grid, a platform resulting from technical work to design and develop new means of augmenting the interoperability of computable knowledge and related services. The Knowledge Grid is an attempt to overcome the limits of resource and service modularity, identification, and upkeep that were surfaced in Chapter II. With the Knowledge Grid, advice-giving services can be modularized, uniquely identified and more easily kept up to date.

This chapter is the last in a trilogy of chapters that completes the principal scientific argument of this dissertation. That argument asserts the feasibility of building a domain-agnostic technical knowledge infrastructure capable of significantly augmenting the degree of interoperability of computable knowledge. This chapter reports a study where the interoperability of computable knowledge is successfully augmented in a way that facilitates the generation and communication of well-informed advice.

Specifically, this chapter brings in the concept of a *pipeline* to demonstrate the utility of more highly interoperable computable knowledge. In computing, a pipeline is a linear sequence of specialized modules used to execute instructions. Pipelines are so pervasive in computing that the verb pipelining is used to mean setting up a step-wise organization of computer processes.

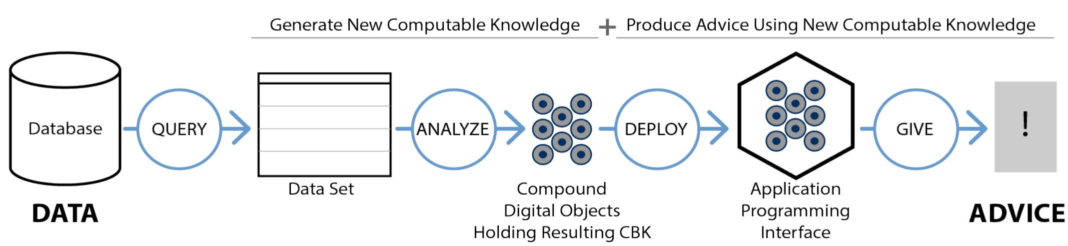
One type of computing pipeline becoming more relevant in the biomedical science and health domains is the *data analysis pipeline*. A data analysis pipeline is a linear sequence of data transformations that convert "raw" or unprocessed data into analytic results that manifest new knowledge. Example biomedical data analysis pipelines include *INKA*[1], for determining cancer drug targets by making inferences about enzyme activity from tumor cell samples, *lumi*[2], for processing Illumina DNA microarray data to discover the functional roles of genes, and *VirusHunter*[3], for identifying novel viruses from DNA sequence data.

Data analysis pipelines end when new knowledge has been generated and communicated. However, as we have seen, after it has been communicated, computable biomedical knowledge can then be applied to produce advice targeted at specific health-related decisions.

To show how augmenting the interoperability of computable knowledge can be useful, this chapter explores the addition of another important step to data analysis pipelines. That added step is one that produces advice from the knowledge that data analysis pipelines create. We call a data analysis pipeline that has been extended to produce advice in this manner a *data-to-advice pipeline*.

Figure 4.1 below illustrates the general case of a data-to-advice pipeline. The pipeline combines four processes to generate new computable knowledge and then use it to produce advice. We partially automated these four processes using various tools, some of which already existed and others that we developed. However, these four processes could be fully automated in a sequence of computer-executed actions. Starting on the left with a Database, the first process is to run a query to get a Data Set. The second process analyzes that Data Set using a method that produces compound digital objects to store the results as machine-interpretable (i.e., computable) knowledge. The third process then deploys those same objects via an Application Programming Interface (API) mechanism. The final process uses the API to give advice automatically.

Figure 4.1 General case of a data-to-advice pipeline



A key feature of the general data-to-advice pipeline portrayed in Figure 4.1 is that the same computable knowledge format is used to communicate knowledge resulting from analysis and to deploy knowledge to produce advice. Compound digital objects are used to bridge these two processes. Compound digital objects have a number of characteristics. They are formally structured digital objects designed to be used by machines. They are modular and, by definition, they each have a unique identifier[4].

This chapter shares the design and development of a new data-to-advice pipeline called ScriptNumerate. ScriptNumerate uses a common digital object-based format both to communicate the analytic results it generates and also to apply those results, which constitute computable biomedical knowledge (CBK), to produce advice. Specifically, ScriptNumerate leverages Knowledge Objects, which are compound digital objects, and the KGrid Activator component of the Knowledge Grid to augment the interoperability of computable biomedical knowledge. In this manner, it attempts to overcome the three limitations of stand-alone advice-giving systems surfaced in Chapter II and reiterated above.

The story of designing, developing, and conducting the first experiment with the ScriptNumerate data-to-advice pipeline unfolds next. In the end, ScriptNumerate's capabilities overcome the limitations of computable knowledge modularity, identifiability, and updateability that pertain to computable knowledge resources and related services in stand-alone advice-giving systems like MedMinify, which was discussed in Chapter II.

The reader may wonder why this chapter presents ScriptNumerate as an example of augmenting the interoperability of computable biomedical knowledge instead of presenting an upgraded version of MedMinify (see Chapter 2). The reason for this is straightforward. The three types of computable biomedical knowledge used by MedMinify do not directly result from an initial or primary analytic process. Instead, the computable biomedical knowledge in MedMinify results from secondary analysis of the scientific literature or from looking up facts about drug products. In contrast, ScriptNumerate includes an embedded primary analytic process, a process that analyzes raw prescription data collected from the field. In the future, MedMinify could potentially be upgraded to have more highly interoperable CBK. For now, however, ScriptNumerate is a more feasible starting point to show the advantages of augmenting the interoperability of CBK by using a data-to-advice pipeline with an embedded analytic process.

BACKGROUND

Data-to-advice pipelines and interoperability

Data-to-advice pipelines, like the pipeline portrayed above in Figure 4.1, augment the interoperability of computable knowledge, including CBK. Recall that interoperability is the ability for arbitrary things to function jointly[5]. Interoperability varies by degrees in the things that exhibit it. In Chapter II, a theoretical framework for assessing the degree to which two things exhibit interoperability was introduced. It proposed five levels of interoperability:

Not: Two things cannot, under any circumstances, be made to jointly and suitably perform a desired set of functions

Slightly: Given two things, to make them jointly and suitably perform a desired set of functions, it requires less work to replace or entirely remake one or both of them than to modify and upgrade one or both of them

Modestly: Given two things, to make them jointly and suitably perform a desired set of functions, it requires about the same amount of work to replace or entirely remake one or both of them than to modify and upgrade one or both of them

Highly: Given two things, to make them jointly and suitably perform a desired set of functions, it requires more work to replace or entirely remake one or both of them than to modify and upgrade one or both of them

Perfectly: Two things can perform a desired set of functions suitably without any further work either to make it or keep it so

Data-to-advice pipelines exhibit interoperability in multiple ways at each step or stage in the pipeline. Indeed, this step-wise interoperability is the chief characteristic of any pipeline. In Figure 4.1, a high degree of internal interoperability exists between the database and the query function, the data set and the analysis function, and the compound digital objects and the deployment function. Otherwise the pipeline would not function.

In addition, to be successful, data-to-advice pipelines have to achieve highly interoperable advice-giving services. Therefore, data-to-advice pipelines use an API to expose a technical advice-giving service to external systems beyond the pipeline. Via an API, external information systems interoperate with the endpoint of the pipeline to get the advice it produces.

Paepcke et al. argue in general that to augment interoperability for data, information, or knowledge resources, multiple approaches need to be pursued in light of known tradeoffs[6]. They recount some of these tradeoffs: Strong standards are helpful but hard to develop, institute, and maintain. Families of standards offer flexibility at the price of greater variability and higher support costs. External mediation between competing standards usually results in some information loss. Specification-based API interactions are useful, but also complex. Finally, mobility is facilitated by modularization and containerization of information resources. All of these approaches are used to augment the interoperability of CBK for ScriptNumerate.

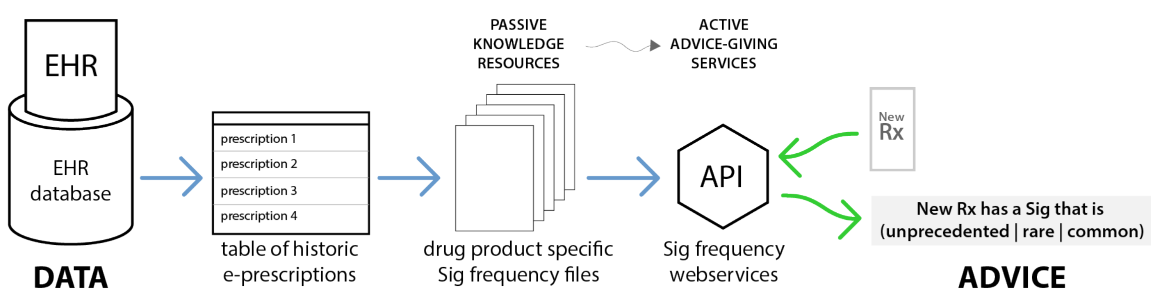
Van de Sompel and Nelson report on 15 years of interoperability efforts for World Wide Web-based systems[7]. They describe the evolution of several technical, web-centric efforts towards better information resource interoperability. They point out a key lesson learned, which is that the technical methods of HTTP's "uniform interface" for APIs on the World Wide Web (i.e., GET, POST, PUT, DELETE), combined with typed hyperlinks, defined media types, and the Resource Description Framework (RDF), can significantly augment the interoperability of compound digital objects. Amongst other methods to augment the interoperability of CBK, ScriptNumerate leverages the architecture and methods of the World Wide Web (WWW) for its outward-facing advice-giving service API.

Summarizing, ScriptNumerate is used to demonstrate that several limitations of stand-alone advice-giving systems, which have been previously noted, can be overcome through the development of working data-to-advice pipelines.

What ScriptNumerate does

ScriptNumerate is portrayed conceptually in Figure 4.2 immediately below. Its purpose is to detect and highlight odd, strange, i.e., *atypical*, electronic prescriptions or e-prescriptions.

Figure 4.2 Conceptual diagram for the ScriptNumerate data-to-advice pipeline



A key reason to detect and highlight atypical e-prescriptions is so that clinicians, particularly pharmacists, can be prompted to evaluate whether or not what is atypical about an e-prescription is warranted. Sometimes what is atypical about an e-prescription is indicative of a problem or a prescribing error, e.g., indicative of an e-prescription for the wrong drug.

As a byproduct of the widespread adoption of Electronic Health Records (EHRs), most prescriptions in the U.S. are now e-prescriptions[8]. E-prescriptions are digital text objects represented using either proprietary or open data schemas. These e-prescription data schemas define several discrete data elements constituting the parts of an e-prescription. These parts include a named drug product, the active ingredients in the drug product, the quantities of those active ingredients, and the *Sig*, a Latin term for a label, which indicates the instructions for use of a drug product (e.g., Take 100mg every morning).

The e-prescriptions used as a data source for ScriptNumerate are represented using a proprietary data schema of one EHR vendor. This proprietary schema represents e-prescriptions using the following seven discrete data elements:

- local unique drug product identifier (e.g., “310”)
- drug product name (e.g., acyclovir 400mg tablet)
- generic ingredient name (e.g., acyclovir)
- ingredient strength (e.g., 400 mg)
- route of administration (e.g., oral)
- dosage form (e.g., tablet)
- Sig (e.g., take 400 mg 2 times daily)

ScriptNumerate is so named because it automatically counts historical e-prescription Sigs to generate statistical Sig frequencies for a given drug product.

The stages of the ScriptNumerate data-to-advice pipeline are described next. Starting on the left of Figure 4.2, as a byproduct of prescribing, e-prescription data are collected in an EHR database. Those data are queried to form a table of historical e-prescriptions. That tabular data set is analyzed, on a drug product specific basis, resulting in counts of unique Sigs by drug product. These Sig counts are represented in the Python computer language. In this format, they constitute a simple form of CBK about historical prescribing patterns. This CBK is stored in numerous drug product specific Sig frequency files. These files are passive knowledge resources. They are not capable of responding programmatically to answer questions about their content.

ScriptNumerate continues beyond this analytic endpoint by converting passive drug product specific Sig frequency files into active advice-giving services. It adds software code to these files and then exposes that code via an API. By so doing, ScriptNumerate's API enables on-demand interrogation of the CBK in its drug product specific Sig frequency files. As the two green arrows on the right side of Figure 4.2 indicate, advice-giving services provided by ScriptNumerate's API are engaged to generate advice about new e-prescriptions. This advice takes the form of messages indicating whether new e-prescription Sigs are common, rare, or unprecedented, in comparison to the Sigs of historical e-prescriptions for the same drug product.

ScriptNumerate uses RxNorm

RxNorm is a terminology from the National Library of Medicine[9]. It enables e-prescription data to be normalized. RxNorm assigns Concept Unique Identifiers called RXCUIs to drugs and drug products. The Semantic Clinical Drug (SCD) subclass of these identifiers pertains to this project. Every SCD in RxNorm has three terms, active ingredient + strength + dosage form. For example, 'Acyclovir + 400mg + Oral Tablet' is an RxNorm SCD with RXCUI 197311. RxNorm SCDs are used by ScriptNumerate to normalize locally acquired e-prescription data at the drug product level.

SIGNIFICANCE

ScriptNumerate is primarily significant because it demonstrates that a data-to-advice pipeline can be built which augments the interoperability of CBK enough to overcome several previously noted limitations.

ScriptNumerate is also significant in its own right because of the advice it provides about e-prescriptions. The advice from ScriptNumerate can help to improve medication safety. ScriptNumerate builds on the previously established potential for recognizing atypical e-prescriptions systematically[10]. It offers a new means to expand both the scope and the scale of this capability to identify atypical e-prescriptions. Adding to its significance in this regard, screening for atypical e-prescriptions is one of several core capabilities needed to partially automate e-prescription review. This automation is of interest to better support pharmacy practice[11].

RESEARCH QUESTIONS

The following three research questions are addressed by the study presented in this chapter:

RQ1. What are the required technical functions of a data-to-advice pipeline capable of computing advice about atypical e-prescription Sigs based on historical e-prescribing patterns?

RQ2. What types of atypical e-prescription Sig advice, and how much of each type, can be generated by using ScriptNumerate to compare contemporary e-prescriptions for an elderly adult population of hospital inpatients with historical e-prescribing patterns arising from treatment of a similar population during the previous year?

RQ3. How can ScriptNumerate aggregate computable biomedical knowledge about historical e-prescriptions from multiple EHR sources to form combined collections of Sig Frequency Knowledge Objects?

METHODS

Use of the Knowledge Grid platform

This study of ScriptNumerate uses the Knowledge Grid (KGrid) platform to help convert passive CBK resources into active CBK-enabled advice-giving services. Using KGrid technology, a special type of compound digital objects holding CBK, called Knowledge Objects (KOs), are automatically created and then deployed within ScriptNumerate to engender advice-giving services. These KOs are built in accordance with the Knowledge Object Reference Ontology (KORO), a formal specification that we developed which is described in Chapter III[12].

Once they are built automatically, ScriptNumerate's KOs are then "activated" by loading them into a deployed instance of the KGrid Activator. The Activator provides the API that gives rise to ScriptNumerate's advice-giving capabilities. The Activator oversees computer execution of the CBK held in the KOs. Thanks to the KGrid's Activator, ScriptNumerate's API can process e-prescription data and compute e-prescription-specific advice.

Software development for this ScriptNumerate study

To build the additional software pieces of ScriptNumerate needed to complement available components of the Knowledge Grid, a set of software scripts were written using PERL (5.18.2) and Python (2.7.10). In addition, we used Python and JavaScript (ECMAScript 5) to encode CBK for KOs later made serviceable by KGrid's Activator.

The first additional software script I developed and tested as part of ScriptNumerate, called *RxCounter*, converted tables of raw prescription data into drug product specific Sig frequency files. Using 170 lines of code in PERL, the RxCounter script processed rows of individual prescription data to generate *drug product specific SIG frequency files* with like-indexed arrays of Sigs and Sig counts *in Python code*. Here is a relevant snippet from one of those files showing the Sigs and Sig counts in two corresponding Python arrays:

```
medicationname = ['ACYCLOVIR 400 MG TABLET']
Sigs = ['400 MG 2 TIMES DAILY','400 MG ONCE DAILY','400 MG 5 TIMES DAILY','400 MG EVERY 12 HOURS','400 MG 3 TIMES DAILY','800 MG ONCE DAILY']
Sigfrequencies = [123,8,3,2,4,1]
```

The second software script I developed and tested for ScriptNumerate is called *PayloadMaker*. This script accepted *drug product specific SIG frequency files* as inputs and added additional lines of Python code to make functional Knowledge Objects suitable for use with Knowledge Grid platform components. A core element of this additional code are the conditional logic statements needed to determine whether a new prescription is common, rare, or unprecedented compared to the Sigs and Sig counts in each Knowledge Object:

```
if found == 0:   # NEW PRESCRIPTION SIG NOT FOUND
  answerSentence = 'RESULT: UNPRECEDENTED '
elif found == 1 and ratiorx > 0.05:  # THRESHOLD VALUE
  answerSentence = 'RESULT:COMMON '
else:
  answerSentence = 'RESULT:RARE '
```

As a core part of ScriptNumerate, I developed and tested a third script in PERL, called the *MetadataMaker*, to further process *drug product specific SIG frequency files* and generate JavaScript Object Notation (JSON) metadata files for each Knowledge Object. Here is an example of such an JSON file:


```
{"payload": {
  "content": "",
  "engineType": "Python",
  "functionName": "rxcompare"},
      "metadata":
      {"title":"197311 For ACYCLOVIR 400 MG TABLET",
              "owner":"Michigan Medicine",
```

"description":"One of a collection of prescription frequency knowledge objects with prescription data for inpatients 75 and older at UM Hospitals from 2016",
"contributors":"DLHS DLKS Knowledge Grid Team 2016-18",
"keywords":"197311 RXNORM SCD",
"published":false,
"lastModified":1515942979000,
"createdOn":1515942979000,
}
}

I wrote a fourth script comprising a core part of ScriptNumerate in Python code and called it the *KOMaker*. The KOMaker was used to combine corresponding Payload and Metadata files generated by the previously described scripts into working Knowledge Objects serialized in JSON. These Knoweldge Objects were then used by the KGrid Activator, version 0.58, to engender advice-giving services.

Finally, I developed and tested several other script besides the core software scripts comprising ScriptNumerate. One of these, called *EngageActivator*, I used to tag large numbers of individual prescriptions automatically as having common, rare, or unprecedented Sigs.

Data source and data characterization

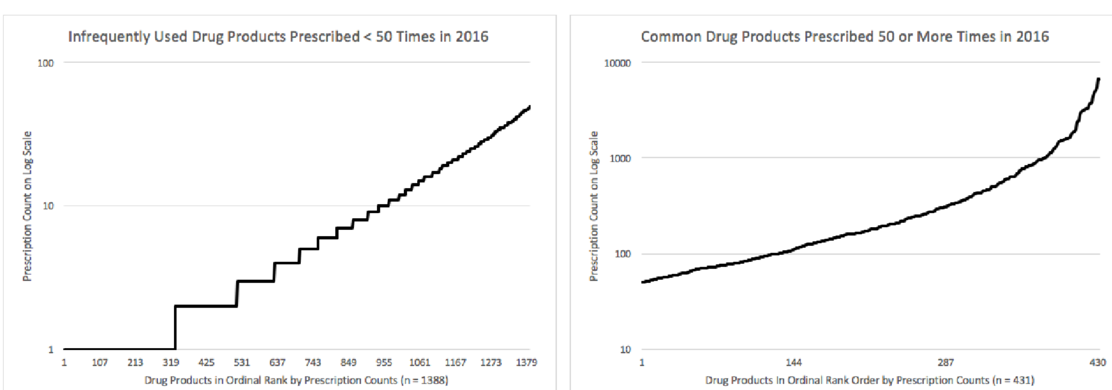
For this feasibility study of ScriptNumerate, after gaining IRB approval, we used two limited data sets each containing one year's worth of e-prescriptions. Every e-prescription had a drug product associated with an RxNorm SCD code. Every e-prescription was actually placed in the EHR for a hospital inpatient age 75 years or older. We ran queries to extract these historical e-prescription data from the University of Michigan's EHR data repository (Clarity, EpicCare, Epic, Verona, WI.).

A sub-population of elderly adults, age 75 and older, was chosen for these experiments because drug dosages, and hence e-prescription Sigs, are specific to this subpopulation[13]. In practice, comparisons of e-prescription Sigs to historical prescribing patterns stratified by subpopulation are required because prescribing is population-specific.

The first data set included e-prescriptions from calendar year 2016 (n = 232,203). From these data, historical prescribing patterns, in the form of statistical Sig frequencies, were determined on a drug product basis for a population of elderly adults. A historical e-prescription prescribing pattern was generated for the 431 drug products that were prescribed at least 50 times for this population during 2016. This cutoff of 50 times was used as an approximate way to distinguish between drug products used more than once a week in 2016 and drug products used less than once a week in 2016. It is assumed that drug products prescribed less than once a week are reasonably thought of as infrequently used drug products according to this arbitrary but pragmatic criteria. In this analysis, e-prescriptions for drug products prescribed less than once a week, on average, are treated as e-prescriptions for infrequently prescribed drug products.

Figure 4.3 below shows two graphs of counts, by drug product, of the number of e-prescriptions in the 2016 data set. A total of 1819 unique drug products are included. The y-axes have a log scale. The graph on the left shows the number of e-prescriptions placed for 1388 drug products prescribed fewer than 50 times for the study population in 2016.

Figure 4.3 Prescription counts per rank-ordered drug product from 2016 for hospital inpatients 75 years and older are given on a log scale for 1388 infrequently prescribed and used drug products (left) and 431 commonly prescribed drug products (right)



Note that 332 drug products were only prescribed once in 2016 for this population. The graph on the right shows the number of e-prescriptions registered for 431 commonly prescribed drug products. Of these, 129 drug products were prescribed between 50 and 100 times in 2016 for the study population, while 302 others were prescribed more than 100 times.

The second e-prescription data set, spanning all of calendar year 2017, was similar to the first. It included e-prescriptions for the same 1819 unique drug products (n = 251,928). Using ScriptNumerate's API, every e-prescription from 2017 was automatically compared to the historical prescribing patterns derived from the previous year's e-prescription data. The 2017 data were processed automatically to ascertain, for all e-prescriptions for 431 commonly prescribed drug products (n = 236,320), whether or not each e-prescription's Sig was common, rare, or unprecedented in comparison to the Sigs for the same drug products from 2016. ScriptNumerate also advised in all cases when an infrequently prescribed drug product was prescribed in 2017 (n = 15,608).

Generation and quantification of advice

We used a software script to compute advice for 251,928 e-prescriptions from 2017 using ScriptNumerate. This batch process took 80 minutes to run on a laptop computer. We used a spreadsheet to quantify the number of e-prescriptions with unprecedented, rare, or common Sigs. Because the definition of a rare Sig is arbitrary, we chose two different thresholds to define what a rare Sig is. Common Sigs are also arbitrary and are defined as the obverse of rare Sigs.

RESULTS

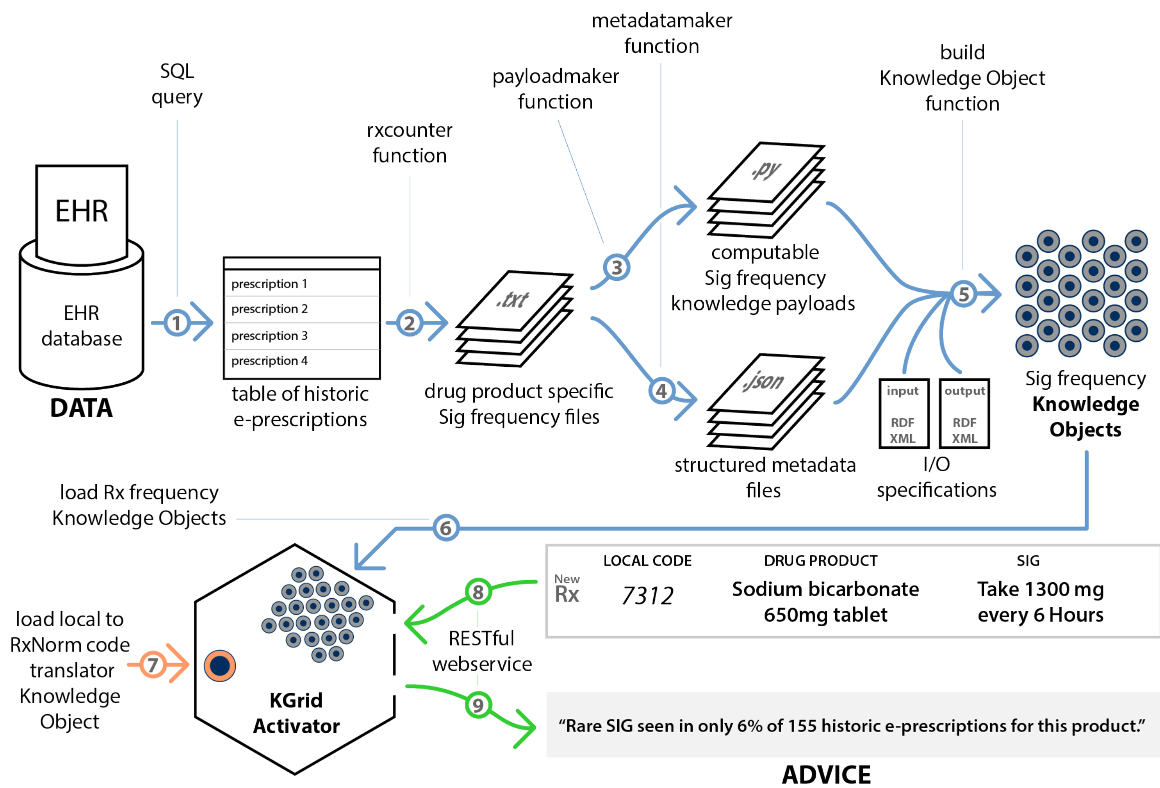
To begin this section, results are given for a single drug-product to explain and clarify what ScriptNumerate does. The exemplar drug product used is Acyclovir 400mg Oral Tablet, an antiviral drug product.

Table 4.1 Results from Using ScriptNumerate Only for the Drug Product Acyclovir 400mg Tablet where the Threshold for Determining a Rare Sig is that it Appears in <=10% of Historical Sigs

Count of 2016 E-prescriptions	Six Sigs used for Acyclovir 400mg Oral Tablet in 2016 (with corresponding statistical frequencies)		Results from comparing 190 **2017** e-prescription Sigs		
			Unprecedented	Rare	Common
141	400 MG 2 TIMES DAILY (123) 400 MG 3 TIMES DAILY (4) 400 MG EVERY 12 HOURS (2)	400 MG ONCE DAILY (8) 400 MG 5 TIMES DAILY (3) 800 MG ONCE DAILY (1)	10	21	159

As shown above in Table 4.1, 141 historical e-prescriptions for this product were placed during 2016 with six different Sigs. The most common Sig was "400 MG 2 TIMES DAILY", which was used for 123 of 141 e-prescriptions. ScriptNumerate stores these six Sigs and their frequencies in a drug product Sig frequency file. This file is then encoded in Python, making it an instance of CBK. Next, using ScriptNumerate's API and this instance of CBK, 190 e-prescriptions from 2017 were processed to assess their Sigs. The results show that, of these 190 e-prescriptions, 159 had a common Sig, 21 others had a rare Sig, and 10 more had unprecedented Sigs, which are Sigs that do not appear at all in the list of Sigs from 2016. An example of an unprecedented Sig from 2017 is "200 MG 5 TIMES DAILY." In this case, instead of prescribing half of a 400 mg oral tablet, there is a 200 mg tablet that could be prescribed.

Figure 4.4 ScriptNumerate data-to-advice pipeline showing its components and its 9 essential technical functions



***RQ1.* What are the required technical functions of a data-to-advice pipeline capable of computing advice about atypical e-prescription Sigs based on historical e-prescribing patterns?**

Figure 4.4, immediately below, is a technical depiction of the ScriptNumerate data-to-advice pipeline. It portrays nine essential components required for the ScriptNumerate data-to-advice pipeline to function as intended. These components are described next.

The first technical function required in the ScriptNumerate pipeline, illustrated in Figure 4.4 and indicated by ①, is to run a SQL query against an EHR database resulting in a table of historical e-prescriptions from the EHR.

The next required function, indicated by ②, is the rxcounter. It does what a "pivot table" function does by automatically generating counts, specific to each drug-product, resulting in the frequencies of the Sigs used in e-prescriptions for that drug product. The result of the rxcounter function is a series of drug product specific Sig frequency text files.

The next two necessary technical functions, indicated by ③ and ④, are the payloadmaker and metadatamaker functions. These two functions convert Sig frequency text files into executable Python-encoded "payload" files and produce a corresponding structured metadata file in JavaScript Object Notation (JSON), respectively. Each payload is an instance of CBK that will become the core component of a unique Knowledge Object. Similarly, each structured metadata file serves as a framework to organize and describe the multiple components comprising each KO.

The build Knowledge Object function, indicated by ⑤, is a pivotal ScriptNumerate function. It automatically unites four digital components: payloads, structured metadata files, an input specification, and an output specification, to form individual Sig Frequency Knowledge Objects, each with its own unique identifier. The compound digital Knowledge Objects created at this step conform to KGrid's formal specification of what a Knowledge Object is[12].

The next required technical function, indicated by ⑥, involves loading Knowledge Objects into the KGrid Activator. This is a matter of moving digital copies of these Knowledge Objects into a special folder for the Activator.

The seventh technical function, indicated by ⑦ on the lower left of Figure 4.4, involves a special Knowledge Object to further enhance interoperability. This KO holds a lookup table that maps local medication ID codes from our EHR to corresponding RxNorm SCDs. Thus, this special KO acts as a code translator, enabling e-prescriptions carrying only a local medication ID code to be compared with appropriate Sig Frequency KOs in ScriptNumerate.

The last two functions are linked. They are indicated by the green arrows and the numbers ⑧ and ⑨ in Figure 4.4. These are the functions of ScriptNumerate's API, which is enabled by the KGrid Activator. This API is a RESTful webservice using the HTTP-based WWW architecture. It allows external systems to send e-prescription Sigs to ScriptNumerate (⑧) for which ScriptNumerate computes and returns a message of advice about each Sig (⑨).

Summarizing, ScriptNumerate has nine required technical functions. Of these, functions ①②③④⑤ and ⑦ in Figure 4.4 were developed for this study. The KGrid Activator was built beforehand in our lab. It directly supports functions ⑥⑧ and ⑨. These nine functions comprise a working data-to-advice pipeline that computes e-prescription advice.

RQ2. **What types of atypical e-prescription Sig advice, and how much of each type, can be generated by using ScriptNumerate to compare contemporary e-prescriptions for an elderly adult population of hospital inpatients with historical e-prescribing patterns arising from treatment of a similar population during the previous year?**

ScriptNumerate computes four different types of advice. First, it advises whether an e-prescription is for an infrequently used or frequently used drug product. Three more advice types arise when e-prescriptions are for frequently used drug products. Then ScriptNumerate advises whether a Sig is common, rare, or unprecedented in comparison to statistical frequencies of past Sigs for the same drug product prescribed for a similar population.

Results from our first ScriptNumerate experiment are reported in Table 4.2 below. We found that 15,608 (6.2%) out of 251,928 e-prescriptions placed in 2017 were placed for one of 1388 infrequently used drug products. No determination of whether the Sigs for these e-prescriptions were unprecedented, rare, or common could be made. However, in these cases, advice to pharmacists and others could indicate when an infrequently used drug product has been prescribed.

Table 4.2 Results from Comparing 2017 E-prescriptions Using Sig Frequencies as Historical Prescribing Pattern Data from 2016

2017 e-Prescriptions Compared to Data from 2016	For an Infrequently Used Drug Product (%)	For one of 431 Frequently Used Drug Products (%)				
251,928	15,608 (6.2%)	236,320 (93.8%)				
		↓				
		Unprecedented Sig (%)	Rare Sig defined at < = 5% Threshold		Rare Sig defined at < = 10% Threshold	
			Common Sig (%)	Rare Sig (%)	Common Sig (%)	Rare Sig (%)
		14,456 (6.1%)	180,568 (76.4%)	41,297 (17.5%)	157,422 (66.7%)	64,443 (27.3%)

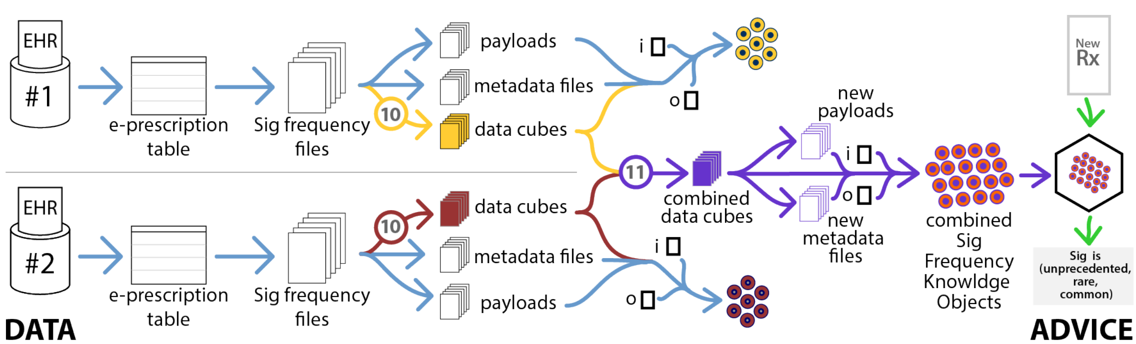
For 431 commonly prescribed drug products, another 14,456 e-prescriptions had Sigs that were unprecedented. Unprecedented Sigs were Sigs not seen during the prior year for a given drug product. In these cases, interruptive alerts advising that a Sig is unprecedented could be warranted to improve e-prescription review. While not every unprecedented Sig suggests a problem, pharmacists are responsible for evaluating whether an e-prescription carrying an unprecedented Sig is warranted or whether, as sometimes happens, the prescriber has made a mistake.

For e-prescriptions placed for the same 431 commonly prescribed drug products, two different thresholds were used to determine whether or not Sigs were "rare." The first threshold defined rare as a Sig appearing in 5% or fewer of historical Sigs for the same drug product. At a 5% threshold, 41,297 of 236,320 e-prescriptions Sigs were considered rare (17.5%). When the threshold for rare Sigs is raised to be less than or equal to 10% of historical Sigs for the same drug product, then 64,443 of 236,320 e-prescription Sigs were found to be rare (27.6%). While interruptive alerts may not be warranted for rare Sigs, tagging e-prescriptions with advice that a Sig is rare could prove helpful in some cases.

***RQ3.* How can ScriptNumerate aggregate computable biomedical knowledge about historical e-prescriptions from multiple EHR sources to form combined collections of Sig Frequency Knowledge Objects?**

One of the potential benefits of using compound digital objects, like Knowledge Objects, to package, manage, and deploy CBK is that components of compound digital objects can be designed to be automatically aggregated. To address RQ3, we added an aggregation capability to the ScriptNumerate data-to-advice pipeline. To do this, we applied the RDF Data Cube vocabulary (www.w3.org/TR/vocab-data-cube/) to represent Sig frequency files as data cubes. This vocabulary is based on the SDMX ISO 17369 standard for statistical data exchange. It enables publishing of multi-dimensional statistical data using RDF triples. This addition to ScriptNumerate is portrayed in Figure 4.5 below.

Figure 4.5 How ScriptNumerate aggregates data about e-prescription Sigs from multiple EHRs (i = input spec, o = output spec)



On the left side of Figure 4.5, the beginnings of two ScriptNumerate pipelines are shown. The top pipeline starts with data from EHR database #1 and the bottom one starts with data from EHR database #2. However, what is illustrated moving from left to right in the middle of Figure 4.5, starting at ⑪, is a third combined ScriptNumerate pipeline built up from aggregating the results in Sig frequency files from the top and bottom ScriptNumerate pipelines.

To both pipelines in Figure 4.5, a tenth technical function (⑩) has been added to represent Sig frequency files as data cubes. The top pipeline's data cubes are in yellow and the bottom pipeline's in red. An eleventh technical function (⑪) automatically combines the yellow data cubes with the red data cubes to form purple combined data cubes. These combined data cubes are aggregates of the Sig frequency data from both the top and bottom pipelines. From them, new payloads and metadata files can be generated, giving rise to a third collection of combined Sig Frequency Knowledge Objects for computing advice based on CBK from both EHRs. This additional capability to aggregate Sig frequency files represented using the RDF Data Cube vocabulary enables CBK from many sources to be combined.

DISCUSSION

Augmented interoperability

Here is how ScriptNumerate overcomes the specific limitations noted with the interoperability of the CBK used by the stand-alone advice-giving system MedMinify.

To augment the interoperability of CBK, first and foremost, ScriptNumerate modularizes instances of CBK using Knowledge Objects. Having Knowledge Objects also allows ScriptNumerate to modularize the advice-giving services it provides. It does this by using an Application Programming Interface (API) mechanism provided to it by the KGrid Activator. Further, because the KGrid Activator can be run in any suitable Java environment, including cloud environments, we also gained a high-degree of mobility for ScriptNumerate's API.

Second, ScriptNumerate assigns unique identifiers to each Knowledge Object it creates. These unique identifiers carry over to advice-giving services that each Knowledge Object enables and supports. In other words, specific services are called on by using unique identifiers of Knowledge Objects.

Third, as a data-to-advice pipeline, ScriptNumerate demonstrates how the problem of keeping computable biomedical knowledge updated can be overcome by formatting and packaging computable biomedical knowledge at its inception using compound digital objects. ScriptNumerate shows how Knowledge Objects, in particular, serve as a common resource to bridge the processes of knowledge communication and advice production effectively. Knowledge Objects are automatically generated to hold and communicate new analytic results within the ScriptNumerate pipeline. These same Knowledge Objects are then moved into the KGrid Activator, where its API mechanism enables them the CBK they hold to be accessed and executed to produce advice.

Importance of data-to-advice pipelines

This work is generally important because more highly interoperable CBK is needed for advice-giving systems to remain current as the volume, velocity and variety of CBK all increase. We expect large increases in these things as ‘Big Data’ analytics, machine learning, and other computational methods are applied to large data sets to improve health. We believe these increases will result, in part, from automated data analysis pipelines that are fed regularly with new data and rapidly compute results which manifest CBK. Here we have shown that, by augmenting the interoperability of CBK in multiple ways, it is possible to extend data analysis pipelines to form data-to-advice pipelines that are capable of both analyzing data to create CBK and providing advice by using the CBK they create. Such extended pipelines have the potential to mitigate the shortage of health advice that has motivated the work of this dissertation.

Potential to replace knowledgebases with distributed compound digital object services

Now that ScriptNumerate has been shown to work, it is possible to consider future moves towards a novel computable knowledge infrastructure paradigm that was first envisioned by Kahn and Wilensky[4]. Whereas today, computable knowledge artifacts tend to be collected in knowledgebases, which are often proprietary, in the future this may not be so. The modularity of Knowledge Objects and KGrid Activators enable a competing knowledge infrastructure paradigm of *distributed compound digital object services*. If a digital ecosystem of uniquely identifiable, readily testable, sound and incorruptible Knowledge Objects could be created using upgraded Knowledge Grid components, then it would be possible to break up monolithic knowledgebases into distributed Knowledge Object collections supporting specification-based advice-giving services accessed via APIs.

Indeed, as uniquely identifiable entities, *collections* of Knowledge Objects could be made findable, accessible, interoperable, and reusable. Such collections could even be made readily deployable by encapsulating them in KGrid Activator instances to form modular and mobile advice-giving service kits. These are just some of the possible next steps enabled by the initial components of the Knowledge Grid infrastructure.

The role of the API in this scenario cannot be overemphasized. Coming as it does at the end of a data-to-advice pipeline, an API like that made possible by the KGrid Activator is locus of interoperability between a data-to-advice pipeline and external systems wanting to gain advice. While a whole host of approaches have been used in this study to augment the interoperability of CBK within the context of a data-to-advice pipeline, it is the API at the end of the pipeline that begins to solve the problem of providing sufficient quantities of well-informed health advice to meet global needs.

Implications for pharmacy practice

Our findings from running ScriptNumerate also have important implications for pharmacy practice. They illustrate what could result from deployment of a data-to-advice pipeline for broad atypical e-prescription screening covering hundreds of drug products. In this year-over-year analysis, for an elderly subpopulation of inpatients, we found that 6.2% of e-prescriptions were for infrequently used drug products. We also found that another 6.1% were for commonly used products but had unprecedented Sigs not seen the year before. These results show that ScriptNumerate can highlight atypical e-prescriptions to help pharmacists and others who review them for safety. Therefore, we plan to further develop and then trial ScriptNumerate for multiple subpopulations at one or more sites.

Limitations

This study has a number of limitations. Perhaps its greatest limitation is that it does not include results from an implementation of ScriptNumerate in practice. In the future, we plan to integrate the Knowledge Grid Activator with EHRs and then execute a real-world trial of ScriptNumerate. Another limitation is technical. In its current version, ScriptNumerate requires a Sig contained within a simple JSON text object be the input to its API. It does not yet include desired data pre-processing capabilities to directly parse e-prescriptions formatted using the most common e-prescription data schemas. We look forward to adding these capabilities to a future version.

CONCLUSION

The work presented in this chapter demonstrates three things. First, that compound digital objects can be used as a common format both to communicate new computable knowledge and to provide advice with it. Second, that the Knowledge Grid's initial components, particularly Knowledge Objects and the KGrid Activator, are sufficiently interoperable to support a data-to-advice pipeline that takes in raw data, analyzes it to generate results manifesting new computable knowledge, and deploys the new computable knowledge that it manifests directly to bring about advice-giving services. Third, that it is feasible to build a domain-agnostic technical knowledge infrastructure that is modularized and readily updateable by using uniquely identifiable compound digital objects. Here, through teamwork, we have demonstrated how to increase the interoperability of computable biomedical knowledge sufficiently to potentiate the production of well-informed health advice. Having done so, the work of this dissertation is complete.

References

1. Pham, T. V *et al.* INKA , an integrative data analysis pipeline for phosphoproteomic inference of active phosphokinases. (2018). doi:10.1101/259192
2. Du, P., Kibbe, W. A. & Lin, S. M. lumi: A pipeline for processing Illumina microarray. *Bioinformatics* **24,** 1547–1548 (2008).
3. Zhao, G. *et al.* Identification of Novel Viruses Using VirusHunter -- an Automated Data Analysis Pipeline. *PLoS One* **8,** 1–11 (2013).
4. Kahn, R. & Wilensky, R. A framework for distributed digital object services. *Int. J. Digit. Libr.* **6,** 115–123 (2006).
5. Chen, D., Doumeingts, G. & Vernadat, F. Architectures for enterprise integration and interoperability: Past, present and future. *Comput. Ind.* **59,** 647–659 (2008).
6. Paepke, A., Chang, C.-C. K., García-Molina, H. & Winograd, T. Interoperability for digital libraries workdwide. *Commun. ACM* **41,** 1–23 (1998).
7. De Sompel, H. Van & Nelson, M. L. Reminiscing about 15 years of interoperability efforts. *D-Lib Mag.* **21,** 1–1 (2015).
8. Gabriel, M. H. & Swain, M. E-Prescribing Trends in the United States. *ONC Data Br. No. 18* **July,** (2014).
9. Nelson, S. J., Zeng, K., Kilbourne, J., Powell, T. & Moore, R. Normalized names for clinical drugs: RxNorm at 6 years. *J Am Med Inf. Assoc* **18,** 441–448 (2011).
10. Woods, A. D. *et al.* Clinical decision support for atypical orders: detection and warning of atypical medication orders submitted to a computerized provider order entry system. *J Am Med Inf. Assoc* (2013). doi:10.1136/amiajnl-2013-002008
11. Flynn, A. J. Opportunity cost of pharmacists' nearly universal prospective order review. *Am. J. Health. Syst. Pharm.* **66,** 668–70 (2009).
12. Flynn, A., Boisvert, P., Friedman, C. P., Landis-Lewis, Z. & Lagoze, C. Knowledge Object Reference Ontology. *National Center for Biomedical Ontology Bioportal* (2017). Available at: http://bioportal.bioontology.org/ontologies/KORO. (Accessed: Jan 25 2018)
13. Elinder, C. G., Bárány, P. & Heimbürger, O. The use of estimated glomerular filtration rate for dose adjustment of medications in the elderly. *Drugs and Aging* **31** (2014).

CONCLUSION

I primarily examined two related things in this dissertation, advice and the interoperability of computable knowledge. I have demonstrated some mechanisms to increase the interoperability of computable knowledge and thereby facilitate advice generation. I have taken advantage of these mechanisms to potentiate the production of medication-related advice.

At the outset, I used a systematic literature review to develop a scientific conceptualization of advice. Then, through a trilogy of chapters, I related work to improve the interoperability of computable knowledge by telling the stories of three systems, MedMinify, the Knowledge Grid, and ScriptNumerate. MedMinify is a stand-alone advice-giving system for medication adherence that could be improved by upgrading the interoperability of its computable biomedical knowledge. The Knowledge Grid is a prototype technical knowledge infrastructure for upgrading the interoperability of any computable knowledge. ScriptNumerate is a data-to-advice pipeline for identifying atypical e-prescriptions that demonstrates the potential of the Knowledge Grid to increase the interoperability of computable knowledge and thereby scale up computerized advice generation.

IMPLICATIONS FOR ADVICE STUDIES

Throughout this dissertation, I have conceptualized advice as *an information object targeted at an unmade decision*. A host of scientific paradigms about advice inform this conceptualization. In the broad context of information and document science, I organized these advice paradigms into three high-level categories by relating them to human-to-human advice, machine-to-machine advice, and ICT-mediated advice from machines to humans.

Human-to-human advice is associated with a *communication paradigm*, emphasizing the impact of advice messages on their recipients, a *discourse paradigm*, emphasizing linguistics and analyzing advice as a speech act, and a *psychological paradigm*, emphasizing the social processes of decision-making and how advice changes those processes.

Machine-to-machine advice is associated with an *artificial intelligence paradigm*, emphasizing computed advice that predicts an event, a *theoretical computer science* paradigm, emphasizing advice in the form of software code which provides a common solution to multiple computer systems, and a *computer network paradigm*, which allows machines to be loosely coupled and, consequently, to share advice with one another.

ICT-mediated advice is associated with a dominant *decision support* paradigm. This paradigm examines the impacts of advice generated by technical advice-giving systems and advice-giving services on the decisions that people make after they receive it.

I have built on the existing literature from *advice studies* to provide a new content-based framework for advice. This framework spans the dimension of *environmental uncertainty*. It categorizes advice into four content types: definitive facts, probabilities for random variables, possible alternatives, and acknowledgments of the unknown. Using this advice content framework, human advisors and advice-giving system developers can identify the type of advice they intend to give according to its content. Then, depending on what type it is, they can take advantage of a variety of methods to communicate it to people. For example, if the advice content is a probability, then visual representations of the probability could help communicate it more effectively. Also, this framework helps an advisor to anticipate whether a given message of advice could increase or decrease uncertainty in the mind of its recipient. This is important because advisors have a responsibility to understand the likely impacts of the advice they give on their advisees.

In addition, scientists can use the advice content framework to study the relationships between advice content type and advice effectiveness. In studies of *advice psychology*, especially those that use the Judge-Advisor System method, there is a need to broaden the content types of the advice studied to better understand its effects.

Another implication is that there are interactions between scientific advice paradigms. Some of these interactions are not yet fully understood. As one example, the *artificial intelligence* and *network* paradigms for machine-to-machine advice interact with the ICT-mediated advice paradigm of *decision support*. The Knowledge Grid infrastructure enables this interaction. Its modular Knowledge Objects and networkable KGrid Library and KGrid Activator components enable machine-to-machine advice to be aggregated and combined on a computer network in ways that could ultimately result in richer ICT-mediated advice for people.

The work on ScriptNumerate shows that, with the help of a technical infrastructure for increasing the interoperability of computable knowledge, data analysis pipelines can be extended to become data-to-advice pipelines. This is a momentous result. It anticipates a future where large-scale data analytic processes, resulting in profuse quantities of new general knowledge, are linked to large-scale advice production processes, which particularize new general knowledge to individual cases via advice generation. This link between analytics and advice production involves computable knowledge formatted in particular ways, perhaps using compound digital objects like Knowledge Objects. For Learning Health Systems, this result implies that new biomedical knowledge, so long as it is created and communicated in the digital manner of ScriptNumerate, can be used immediately to generate advice and guide practice as intended.

IMPLICATIONS FOR COMPUTABLE KNOWLEDGE INTEROPERABILITY

This work shows that computable knowledge interoperability can be increased in a variety of ways. We did several things to increase the interoperability of computable knowledge. We modularized computable knowledge using compound digital objects called Knowledge Objects. We also modularized two technical components for managing and deploying computable knowledge, the Knowledge Grid Library and the Knowledge Grid Activator components. In addition, we minted unique identifiers for the Knowledge Objects in the Knowledge Grid ecosystem. Persistent unique identifiers support interoperability by enabling people and systems to find, select, and access the instances of computable knowledge they wish to use. Finally, in ScriptNumerate, we used Knowledge Objects to bridge an analytic process to an advice-giving process, enabling these processes to interoperate to produce advice.

There are other moves that need to be made to increase computable knowledge interoperability further. One of these moves is to extend and enrich the metadata that are used to describe Knowledge Objects so that they become more discoverable. Doing so could link Knowledge Objects and the Knowledge Grid to existing public and private knowledge infrastructure platforms, like PubMed or Google Scholar respectively.

Another item for future work is to improve interoperability measurement. Here I provided a coarse, five-level scale for measuring interoperability. This scale is based on a theory of work and interoperability. This theory proposes that the more work that is necessary to make two arbitrary things interoperate to suitably perform a desired set of functions, the less interoperable those two things are. Thus, there appear to be opportunities to apply models and measures of work to better estimate interoperability.

IMPLICATIONS FOR HEALTH

This work is ultimately for improved human health. I developed MedMinify to address the significant health problems associated with a lack of adherence to prescribed medications for three common chronic diseases. The technical work to develop the Knowledge Grid makes it possible to return to MedMinify and potentially re-architect its computable biomedical knowledge components to make a more highly interoperable advice-giving service that could be used by many computer systems, including Electronic Health Records.

I developed ScriptNumerate, which uses Knowledge Grid components developed by others, to address medication safety. In practice, pharmacists look for atypical e-prescriptions because these e-prescriptions sometimes signal prescriber confusion or unwarranted prescribing. In those cases, pharmacists can intervene to protect patients. If it were deployed in practice, ScriptNumerate would help identify atypical e-prescriptions systematically on a widespread basis. This capability could cause adverse drug events to be averted.

While these two medication-related advice-giving systems are potentially important, there are greater implications for health from this work. In part because of its modular design, the Knowledge Grid could evolve into an underlying infrastructure supporting an ecosystem for computable knowledge that makes it easier to combine different classes of computable knowledge for effect. It is possible that, by enabling computable biomedical knowledge, in particular, to be aggregated and combined, the Knowledge Grid, or something like it, will one day support much more comprehensive advice-giving services and advice-giving systems than the ones we use today.

The methods to develop ScriptNumerate, and the technology of the Knowledge Grid on which they are based, have the potential to be used in a wide variety of other ways. For example, as the analytic methods of machine learning become more established in the health domain, automated pipelines like ScriptNumerate can apply machine learning methods to large and diverse data sets and thereby set in motion new and more manageable advice-giving services enabled by machine learning. Indeed, what ScriptNumerate demonstrates is the possibility to take primary health data of many sorts, as they accrue, and analyze them automatically using any of a large number of analytic processes. Then, once that is done, ScriptNumerate shows how to add additional information to the results of those analytic processes in a manner that supports their direct application, as computable biomedical knowledge, to generate advice on demand.

Looking toward the future, I end this work with an observation and an assertion of my own beliefs. I observe that warranted and useful advice-giving systems for health are being developed but are not being widely adopted and used. I believe that one thing keeping advice-giving systems from being widely adopted and used is that the scope of the computable biomedical knowledge supporting each system is, in and of itself, far too narrow. Therefore, I believe it is important to work toward greatly expanding and diversifying the scope of computable biomedical knowledge that is available to advice-giving systems and advice-giving services. By producing this dissertation, I have tried to take a small step in that direction.