

Smart Objects: An Active Big Data Approach

Stephen H. Kaisler, D.Sc.
SHK & Associates
Laurel, MD
Skaisler1@comcast.net

William Money, Ph.D.
The Citadel
Charleston, SC
wmoney@citadel.edu

Stephen Cohen
Microsoft Corporation
Redmond, WA
stcohen@microsoft.com

Abstract

The world of data and information has been steadily evolving due to changes in the expansion of complexity and of the data processed by our systems. Big Data has evolved from data that are numbers and characters conceived and collected by individuals, to unstructured data types collected by a variety of devices. Recent work has postulated that the Big Data evolutionary process is making a conceptual leap to incorporate intelligence.. This paper proposes that Big Data have not yet made a complete evolutionary leap, but rather that a new class of data - a higher level of abstraction is needed to understand and integrate this “intelligence” concept. This paper examines previous definitions, and offers a new definition for *Smart Objects* (SO) that extends this evolutionary path, examines the basic concept of smart data (is it really exhibiting properties associated with or purported to be intelligence?), and identifies issues and challenges associated with understanding Smart Objects as a new software paradigm. It concludes that Smart Objects incorporate new features and have different properties from passive and inert Big Data.

1. Introduction

The changing and evolving nature of data has been greatly driven in recent years by the expanding concept of Big Data. Big Data’s value is widely accepted. Its importance is rapidly maturing as a critical component of the Information Systems, Computer Science, Information Processing, and Data Analytics disciplines. Research has been focused on characterizing Big Data with five Vs: volume, velocity, variety, veracity, and value. The concept of Smart Data as proposed imbues Big Data with additional attributes and property information through the application of analytics. Thus, Big Data are viewed as inert and described by passive attributes that are changed due to manipulation by external events and procedures.

Our research and observations of the growing Big Data field suggest that not all data are passive or

inactive. Other authors have recognized that data do not always behave as passive groups of characters and numbers.

To discuss these new types of data and distinguish them from the standard, passive view of data, we introduce a new concept— *Smart Objects* –focused on data with intelligent and active data structures. We believe that Big Data, through such structures, has properties and characteristics that require another (higher) level of abstraction in which data manipulation tools do not merely exist to describe and build things through external processes. Rather, we believe the tools must co-evolve with the solutions that they build. Smarter and better tools mean better solutions.

From a software perspective, this is akin to having robots that can repair themselves and create clones or improved versions of themselves. We view Smart Objects as precursors to futuristic objects from movies and the edge of science today - such as TRANSFORMERS (cars, trucks that morph into robots and weapons). One can think of the movies: *Transformers*, *Revenge*, *Dark of the Moon*, *Age of Extinction*, etc. to visualize possible end states with this perspective. It is both a frightening and enlightening scenario.

As the theory differentiating Smart Objects from traditional passive data and our understanding of this phenomena evolves, a number of conceptual issues and challenges have emerged that will need to be recognized and clarified to obtain a full understanding of the best value and utilization of Smart Objects.

2. Background and Motivation

Smart Data is a business data processing concept that has historically acquired many definitions as processing has evolved. The illustrative definitions in table 1 are offered by different authors with a key question in mind: *What does it mean for data to be smart?* These definitions all approach, to varying degrees, the extent to which data are viewed as smart – *something has been added or done to data that somehow makes it intelligent*, e.g., “smart”.

Table 1. Smart Data Definitions

<p>“Smart Data” means information that actually makes sense. ... Smart data is data from which signals and patterns have been extracted by intelligent algorithms. https://www.wired.com/insights/2013/04/big-data-fast-data-smart-data/</p>
<p>If big data is the technological foundation for data driven business decision making, smart data is the analytics we use to extract relevant information and insight from big data, and the visualization we use to present the results. https://econsultancy.com/blog/61682-2013-the-year-of-smart-data</p>
<p>Smart data is digital information that is formatted so it can be acted upon at the collection point before being sent to a downstream analytics platform for further data consolidation and analytics. http://whatis.techtarget.com/definition/smart-data</p>
<p>Smart data makes sense out of Big Data It provides value from harnessing the challenges posed by volume, velocity, variety and veracity of big data, in-turn providing actionable information and improve decision making. wiki.knoesis.org/index.php/Smart Data</p>

There are more definitions in the literature similar to those presented in table 1. These definitions all exhibit an underlying difficulty. None tease out or explicitly state what is it (attribute, property, etc.) that makes the data smart. The definitions point to patterns in the data that can be recognized, after the fact analytics, proper analytic formatting, etc. We find this definitional concern to be a critical problem. Using these definitions, the “smartness” still resides in external applications. But we do believe there are Big Data that exhibit some properties that one can reasonably deem “intelligent” where an “outside” program or process is not required to make sense of, or utilize the data.

2.1 The Smart Object Concept

We introduce the concept of a Smart Object to fill this space. A Smart Object is an *object representation that is computationally aware* – meaning *self-defining* and *self-reflecting*, and, possibly, *self-modifying/self-adapting*. Our objective is to use this definition to explicitly identify the characteristics of “smart” and assess the challenges that will face users and automated systems that will utilize smart objects.

Thus, smart objects (1) *embed one or more computational models that enable the associated data to dynamically respond to CRUD (Create, Read/Retrieve, Update and Delete) operations;* (2) *enable higher level actions such as aggregation, negotiation, or collaboration with other smart objects;* and (3) *exhibit intelligent behavior.* Some authors have asserted that this implies artificial intelligence

that emulates/replicates human behavior. We prefer the term *augmented intelligence* – coined by IBM for its Watson system – in that it enhances active data with human expertise rather than trying to replicate human reasoning.

We do not “see” Smart Objects being integers or strings or other simple data structures that are the foundation of most programming languages. Within SOs, simple data structures will represent attributes, rules and procedures. Rather, SOs incorporate a set of capabilities based on intelligent, autonomous, and self-evolving concepts for active software structures.

2.2 Research Questions

Given our definition for a Smart Object, this paper begins to answer two essential questions about it:

- (Q1) What technologies will enable augmented intelligence in objects?
- (Q2) What are the implications of using Smart Objects to extend the capabilities of our organizational systems to exhibit intelligence?
- (Q3) Are there (perhaps un-proposed) theories of data (such as number or set theories) that will aid in our understanding of the characteristics and properties of Smart Objects?

We barely scratch the surface of this new complex paradigm, but we attempt to set the stage for future research in this emerging discipline.

2.3 Technical Approach

To address these research questions, we will first identify functionality and technology that can be used to implement the awareness we seek to tease from Smart Objects. We surveyed the technical literature and reviewed candidate technologies that may enable awareness and intelligence in objects. Section 3 will describe some of the characteristics and features that Smart Objects must possess. Sections 4 addresses some approaches to answering our research questions. Section 5 presents our conclusions and suggestions for future research.

3. Smart Object Features

Our conceptualization of a Smart Object views it as an object that can be represented by a complex object evolving from basic object-oriented concepts. Its physical instantiation should be based on an existing, flexible, and extensible object-oriented programming language, such as Java, Smalltalk, or an object-oriented version of Common Lisp

Wang [12] proposed an initial approach for cognitive informatics that encompassed three models of computing systems that represent the historical evolution of data processing. Current technology has achieved successes in developing autonomic systems, but true cognitive systems have yet to be developed, although IBM’s Watson seems to possess some capabilities. We suggest that an innate self-cognitive ability is essential to developing active Smart Objects. This “self” capability represents the next evolution in software development. We explore what “self” properties would possibly be exhibited below.

Table 2. Wang’s Models of Computing Systems

<i>Imperative:</i> Passive systems based on stored-program controlled behaviors for data processing.
<i>Autonomic:</i> Active systems that are goal-driven and self-decision-driven machines that do not rely on external instructive and procedural information.
<i>Cognitive:</i> Reasoning systems that implement computational intelligence mechanisms by autonomous inferences that emulate the brain.

Kortuem, Kawsar, Fitton and Sundramoorthy [9] discuss Smart Objects, but it is more limited in its definition of smart objects than our definition. It focuses more on physical things attached to the Internet. Their definition incorporates a concept of awareness of environment, but lacks the concepts of self-awareness and self-reflection, self-adaptation, and self-modification, which are essential to our definition. Thus, our definition lies at the Cognitive level of Wang’s model whereas theirs resides at the Autonomic Level.

3.1 Smart Object Representation

A Smart Object (SO) is a self-contained data and information structure that extends existing object-oriented data models with additional embedded capabilities to implement the features discussed in the following sections.

While current systems are very good in certain areas, they are essentially one-shot wonders and must be rebooted and retrained for each new task. In [9], Raia Hadsdel, a Google Scientist, noted that “there is no neural network in the world, and no method right now that can be trained to identify objects and images, play Space Invaders, and listen to music”. An architecture that will address many of the problems listed in the following sections must be able to *multitask*; and have the ability to change focus as events – both internally and externally – occur. As

Hadsell also remarked, “we can’t even learn [to play] multiple games” simultaneously.

Table 3. SO Characteristics and Features

<i>Persistence:</i> The data are continually available and able to respond to messages. Data are encapsulated within a process that is continually running as long as the computer system is running.
<i>Computational Awareness:</i> An SO must exhibit both internal and external awareness – spatial, temporal, and contextual - in order to actively evolve and respond to both internal and external stimuli and changes of state.
<i>Automatic Notification:</i> Under certain circumstances, such as swarm behavior, SOs must notify other SOs of changes to their structure. This implies awareness of a collection of SOs utilizing a common communications mechanism.
<i>Smart Object Complexes:</i> SOs may be organized into complexes – hierarchical, network, or other topologies – based on domain and problem space requirement, derived from the domain ontology
<i>Classes versus Instances:</i> As with objects, SOs occur as two types: classes which describe a collection of instances and define their attributes and value ranges, and instances which explicitly specify entities within the domain of discourse.
<i>Getting the Data:</i> Underlying the issues and challenges in this paper is the need to get the data to solve the problems a Smart Object encounters. Getting the data means that the SO must have many means of accessing data which will involve accepting streaming data from external sources and the ability to query external data sources.

3.2 Self-Defining

Self-defining data carries within it its own description. Both XML files and RDF files provide a means to construct self-defining documents although these structures are passive. However, extensions to RDF provide some mechanisms for dynamic self-adjustment, such as SWRL.

Self-adjusting data structures are those that can re-arrange themselves when operations are performed upon them. The degree of re-arrangement depends on the current state of the data structure and the operation performed. For example, AVL trees re-balance themselves every time a node is inserted or removed from the tree. Also, consider a min-heap which, given a set of input data, provides a minimum element. Each time new data is added to the min-heap, it updates itself to deliver, possibly, a new minimum element.

Table 4. Self-Defining: Issues & Challenges

<i>Defining Representation Mechanisms:</i> The mechanism for representing data structures should allow data, processes, and rules to be defined within the model. A good example is the Common Lisp representation of lists

which represent data or functions as a single data structure, namely, S-expressions.
<i>Setting and Measuring Criteria for Self-Adjustment:</i> The criteria are problem-dependent but must be dynamically assessed and applied as each new piece of data is received by the data structure, or existing data is changed or removed from the data structure.
<i>Recognizing and Acting Upon Ontological Evolution:</i> As a result of applying analytical algorithms, an SO may change its internal structure by adding new substructures, new attributes or changing the properties of existing attributes. These changes should be accomplished within accepted and documented ontological practices.
<i>Data Description:</i> Standard mechanisms are required for describing the data required, its metadata, and internal procedures for managing it. Enhanced Description Logic system such as OWL and SWRL are logical candidates for this mechanism.

Note: We differentiate changes in an SO's structure from changes in its reasoning ability due to internally represented rules and procedures.

3.3 Self Reflection

Self-reflection is the ability of an entity to examine, introspect, and modify its own structure and behavior at runtime. A *reflective object* is one that can reason about itself given a symbolic representation of the program, its data, and its metadata. Self-reflection is an essential feature for implementing true machine learning.

Table 5. Self-Reflection: Issues & Challenges

<i>Self-Reflecting Metadata:</i> To support self-reflection, smart data's metadata must be enhanced to support self-description. A standard for self-reflection metadata must be devised in order to support communication between SOs. For example, one SO may query another to ask "what do you know about yourself?"
<i>Class Partitioning:</i> A Class SO may have the ability to examine its collection of instances and determine that this set can be partitioned for more explicit description. A Class SO may have the ability to create two or more subclasses and assign its instances to those subclasses through a process of specialization.
<i>Class Generalization:</i> A Class SO may maintain data regarding other classes, including their attribute set. A class may have the ability to analyze this data and create superclasses encompassing a subset of other classes within its knowledge.

3.4 Self-Modification

An SO may be *self-modifying or self-adapting*, e.g., its behavior may change (evolve, degrade, etc.) by changing the rules and/or procedures by which that behavior is exhibited. Self-modification requires that the SO be self-learning. Self-modification will require

that internal procedure and rules be expressed symbolically.

Self-modification of code was an early feature of many applications because of limited memory capacity. Programmers would use every available byte and often dynamically change a group of instructions to accomplish different functions. Dynamic generation of small programs, such as OS/360's creation of data channel programs is one example. As programming methodology evolved, the concept of self-modification was denigrated as harmful. But, it seems to be the only way to build true machine learning systems that can adapt themselves to changes in their internal and external environments.

Table 6. Self-Modifying Code: Issues & Challenges

<i>Immutable Kernel Schema:</i> One cannot expect to just describe smart objects without some basic schema or ontology. A basic immutable kernel schema is required to be the foundation for concept formulation and learning. How much and how descriptive this kernel schema must be is an open research question.
<i>Immutable Kernel Modification Procedures:</i> The corollary to immutable kernel schema is the existence of a kernel set of procedures that performs fundamental or primitive operations essential to the learning process.
<i>SO Code Description:</i> SO internal code that supports self-modification must itself be modifiable to support self-reflectance. A standard symbolic representation for rules and procedures must be defined to support the self-generation of new procedures and rules. We must avoid the existential problem of infinite recursivity.

Note: We think that statistical machine learning systems modify their behavior in limited ways by selecting from among a pre-programmed set of behaviors, perhaps by setting different parameters during iterative execution. In most cases, ML systems of this sort do not embed any domain knowledge in the application, and so, do not "understand" the data or the computed results. Understanding is essential to learning.

3.5 Autonomous Operation

Several authors have drawn an analogy between smart cars, which are autonomous, self-driving machines and a similar capability for data. From this viewpoint, data becomes self-routing - not only across networks, but also self-navigating through analytics pipelines. Today, we look for data via search and query engines, such as Google, Bing, etc., where we manually and explicitly have to fetch the data. In a self-routing data environment, we would specify what data we seek and smart, autonomous, self-reflecting

data would seek us out and direct itself for us to access and store. Of course, the devil is in the details and this is just an initial concept, but a tantalizing one, nevertheless.

Such a capability has several advantages. We would not have to open many files in search of documents that we know are somewhere in a filing system – whether locally or remotely. It may also ensure the use of data beyond its primary usage once accessed. Data might literally be at our fingertips if it “knew” when it was needed. There is significant potential for the way we develop applications and services across the Web and how we conduct business operations.

Autonomous decision-making systems that are situated in dynamic environments must be able to handle the expected decisions, but also respond to the unexpected decisions. Some humans are very good at this problem, while others are not. Studying this long-term behavior may provide insights into solving the problem of handling unexpected decisions. Moreover, symbolic machine learning must be coupled with handling the unexpected decisions in order to improve the systems expected utility.

Supporting this capability, autonomous decision-making systems will require functionality to perceive states of the world/external environment, frame the decisions to be made (possibly parameterized from templates), make inferences to generate and maintain beliefs about the world, identify and gather observations, and fitting these into internal models, identify and execute the best set of actions consistent with internal principles (think Asimov’s Three Laws of Robotics [1]).

Table 7. Autonomous SOs: Issues and Challenges

<p><i>Learning from Computer/Biologic Viruses:</i> Many viruses of either type are able to direct themselves to receptors where they can anchor and multiply. Is this a good model for self-navigation in a network of SOs and other objects? What attributes must complementary SOs present in order to attract and link with other SOs?</p>
<p><i>Broadcasting Data Needs:</i> Mechanisms and “bulletin boards” are needed to advertise what data are needed and when.</p>
<p><i>Non-Stop Operation:</i> Autonomous systems need to be “fail-safe” which seems to imply non-stop operation, particularly in complex life-threatening environments. Easy functional degradation to some minimal capability is a critical requirement. This complex research area involves both hardware and software advances.</p>
<p><i>Information Highway Maps:</i> In order for data to navigate around the Internet, a mechanism for identifying sites that advertise for data and maps for navigating to such sites is required.</p>

Data Self-Routing: A mechanism is required for data to literally “pick itself up” and transfer itself from one server to another along some route. One of us (Cohen) has noted that this is how many viruses work. Thus, he suggests that access controls need to be enforced at the data element/object levels and across network/subnet regions and topologies.

Economic Issues: There are many possible business models and economic and security issues to be addressed in such an ecosystem. Describing the ecosystem of SOs is a research problem. Understanding how SOs will fit in existing ecosystems (playing nice, so to speak) is also a research problem.

Handling Time, Synchronicity, and Streams of Events: Real-world decision-making requires time-dependent dynamics of belief and action. Systems must understand concurrent streaming events – both synchronous and asynchronous, and the critical notions of persistence and dynamics of world states.

3.6. Understanding: The Re-emergence of AI

Underlying the concept of “smart” is the need to understand – itself, its environment, how it computers and solves problems, etc. For a system to be truly intelligent, it is not enough to simply know what it knows, it must know what this knowledge means and what to do in a given situation. Thus, situational awareness is about the knowledge state that’s achieved—either knowledge of current data elements, inferences drawn from these data, or predictions that can be made using these inferences. Klein, Moon, and Hoffman [8] note that sensemaking (e.g., understanding) is “a motivated, continuous effort to understand connections (which can be among people, places, and events) in order to anticipate their trajectories and act effectively”.

Artificial Intelligence (AI) – the attempt to emulate human reasoning in a computer system – flourished in the early 1970s to early 1980s before crashing for several reasons. Among these were fragility of successful systems, lack of robustness, poor performance due to inadequate hardware, and inability to deal with open world environments. The so-called “AI Winter” ensued which has slowly thawed over the past decade. In the intervening thirty years or so, advances in technology, methodology, hardware and software, and cognitive models have sparked a renewed interest in AI.

AI encompasses many sub-disciplines which are viewed, today, as maturing disciplines in their own right – natural language processing, image and video processing, clustering and classification, among others. A generative agent that is autonomous should take actions based on the expected utility of its decisions (in the service of its symbiotic partner(s)).

This principle seems easy to state, but it is difficult to implement in practice. More research into the origin(s) of utility measures relative to a particular decision, and the assignment of utility to sub-problem models decomposed from high-level goals in order to assess the value contributions of individual and aggregate data elements and their transformations for decision-making is required.

Cognitive computing, as defined by IBM for its Watson system, is a technology that makes human kinds of reasoning for solving problems computable. It encompasses technology – both hardware and software – for addressing complex situations that are characterized by ambiguity and uncertainty. In dynamic, information-rich, and morphing situations, data tends to change frequently, and it is often conflicting. It assumes that the goals of users will evolve as they learn more and redefine their objectives. However, we should remember that prediction and prescription are not cognition, merely computation.

Metacognition is “cognition about cognition” or, basically, being aware about how you think. It extends self-reflection with information about using different strategies for learning and problem-solving. We consider metacognition as the basis for the active control engine at the foundation of a Smart Object.

One research question is whether the control engine is immutable or not. From human biology, we know the brain adapts and evolves throughout one’s life. However, there is a fundamental question of whether there are immutable structures and processes in the human brain that control learning (and forgetting!). Rather than attempting to duplicate or replicate human brain functioning, we should seek to emulate it in an appropriate set of technologies that better fits the digital domain.

Human behavior and intelligence arises from a complex organ - the human brain – which contains millions of neurons. Clearly, complexity has something to do with behavior and intelligence. Our conceptual model of Smart Objects focuses on complex objects representing data sets rather than individual instances of data. To emulate human behavior and intelligence, we suggest that a swarm of SOs working in concert may be necessary to achieve this objective.

Human decision-making is founded upon the concept of “bounded rationality” as defined by Simon [11]. Russell and Norvig [10] noted that it is constrained by the knowledge or information one possesses, the cognitive limitations one has, and the

time limit within which a decision needs to be made. Thus, it is really a search through a set of finite options. Generally, humans assume that other people are rational or we wouldn’t be able to function in our complex world. Imbuing an SO with the ability to make assumptions, but to also question its beliefs and those of others is a difficult problem. “Trust, but Verify” must be embedded into the SO’s cognitive processes.

Humans can explain their decisions and the processes by which they made them. But many of the systems we use today are opaque and only offer rudimentary explanations, if at all. Neural networks are an example of inscrutability. We know how to build them and how to supply them with data, but extracting the explanation or process for the results they yield is very difficult and often unexplained. To put it bluntly, a system will only have real intelligence if it can show how it works to obtain results, meaning it can explain itself.

Table 8. Understanding: Issues and Challenges

<i>Numerical vs. Symbolic Reasoning:</i> These represent the two camps – if you will – of how to perform human-like reasoning. Since we do not yet fully understand how the brain works to store information, process data, and retrieve data/information, this debate will continue.
<i>Emulation vs. Replication:</i> Do machines have to reason like humans or can they reach the same results with alternate methods within similar time frames (or better)? It is unclear whether machines can exactly reproduce human reasoning despite the promise of IBM’s Synapse chip.
<i>Approximate vs. Exact Results:</i> Humans are often very tolerant of inexact results. Should we build this capability into our cognitive systems or must we always insist on exactitude in computing? And, how would we define approximate logic and support it with programming languages?
<i>Symbiosis vs. Servitude:</i> How do we want cognitive software to interact with us – as a symbiotic partner or as merely an intelligent, but active software agent?

3.7 Semantic Annotation

Semantic annotation is the process of enriching data through the attachment of additional information that will make it easier for a computer to use the data in algorithms and reasoning processes. Usually, annotations are written in a machine-interpretable formal language – such as a RDF and OWL, among others. Further research is required to determine how to extend RDF and OWL to support automatic Smart Object annotation. As data volumes increase and the number of information sources increases, manual annotation becomes unrealistic except in special cases.

An *ontology* is a schema for knowledge about a domain. Developing, maintaining, and validating ontologies are difficult research problems in their own right. As Benjamins, Contreras, Corcho, and Gomez-Perez [2] point out, there are six key challenges for the Semantic Web which must be addressed for the Semantic Web to be a successful knowledge repository.

Table 9. Semantic Annotation: Issues & Challenges

<i>Role of AI/ML:</i> Given the volume increase, semi-automatic annotation will need to be accomplished through AI modules that capture domain knowledge as well as ML modules that can deduce/abduce/induce new information.
<i>Validation/Verification:</i> Manual annotation can be fraught with errors due to lack of domain knowledge, amount of training, personal motivation, etc. Tools are required for automatic validation and verification of annotations across the data sets.
<i>The Role of Ontologies:</i> Ontologies make explicit the semantics of the Semantic Web content. They should exist independently of Smart Objects as they represent entire domains, but an SO must include a subset of an ontology to support computational awareness and reasoning.
<i>Linguistic Standardization:</i> A standard ontology description languages (ODL) is required in order to support effective matching and merging of separate ontologies.
<i>Non-textual Data Annotation:</i> Annotation is generally thought of as textual enrichment of data, but there is no reason why it cannot be other forms of media. Arguments against non-textual annotation generally focus on the substantial increase in storage requirements.
<i>Scalability:</i> It is possible that annotation metadata and augmenting data may exceed the amount of original data comprising an SO.

3.8 Machine Learning

Machine learning techniques automate model building to iteratively learn from data and to find hidden insights without being explicitly programmed where to look. However, they do need a kernel of functions or rules to initiate the process of learning. Early machine learning (ML) systems focused on discovering rules for explaining data and modifying those rules to improve system performance. Recently, ML has evolved to learn parameters and descriptions from and make sense of massive amounts of data. These dichotomous approaches can be used to enhance the evolvability of Smart Objects.

The objective for Smart Objects is self-learning, e.g., unsupervised learning, without resorting to “training sets”, but which incorporates extensive domain knowledge to “seed” the effort and to support

decision-making. We will need to assess whether autonomous, unsupervised learning will yield useful and usable results within a discipline

Table 10. Machine Learning: Issues & Challenges

<i>Sufficiency of Data:</i> The amount of data necessary to form a critical mass for ML algorithms varies widely. Mechanisms are required to determine how much data should be available to initiate, sustain, and maintain continuous ML
<i>Tuning the ML Engine(s):</i> Until enough experience is gained in developing autonomous ML engines, the path is likely to require a period of manual intervention to tune ML engines. This will require a unique skill set – not just data scientists – but domain experts with new analytics skills yet to be realized.
<i>Cost of Implementation vs ROI:</i> ML programs are still brittle. Statistical ML programs do not change their behavior, only their computing parameters. Thus, development, training and maintenance of such systems will be costly and must be weighed against the Return On Investment in terms of value to the decision-making process.
<i>Ethical Implications:</i> It is easy to abuse access to Big Data as recent revelations have made that abundantly clear. It will also be easy to assume that a cognitive system is always right. And, as systems become more “intelligent”, the effect on knowledge-based, white collar jobs will be significant.
<i>Deep Search is a Panacea:</i> Deep search as a learning paradigm has worked well for selected problems, such as Chess, but has failed spectacularly in other areas, such as Go (until May 2017). Chess and Go have relatively simple rules, but complex strategies. Determining how Deep Search will help in systems with complex rules and complex strategies is an open research area.

3.9 Computational Awareness

Awareness in computing systems focuses on two environments: internal computational awareness and external contextual awareness. Table 10 presents issues and challenges from both aspects.

There are several approaches to “computational awareness described in the literature. Devanur and Fortnow [3] see awareness as a decision making process to enumerate entities that can resolve a problem such as buying a used car. In their sense, awareness is a process of identification of data that can be used in a decision making process. They cite a paper by Halpern [5] that suggests the use of “awareness structures to allow for natural reasoning” by agents.

Alternatively, “awareness is a mechanism for detecting an event” [14]. Zhao notes that the detected event may not make sense, but may be analyzed to extract information that can improve understanding. He notes that computational awareness is a process for

determining “what is happening, what has happened, and what is going to happen”. Note that these actions are captured in Big Data as descriptive, diagnostic, and predictive analytics.

Neither of these seems to be quite satisfactory in our opinion because they still focus on something happening to data. Thus, they view computational awareness as a process of continually updating the data’s record of its provenance and its environment as represented by operations upon it.

We suggest that it is a driver for initiating internal operations to enhance the data through operators that assess quality; derive additional data based on an ontological model; extend the ontological model; and seek to acquire additional data through querying external data sources.

Zhao [14] notes that many types of aware systems have been described based on the type of events to be aware of, such as situations, intentions, preferences, location, etc. These do not seem to be very helpful as there is substantive overlap among the event types and they do not help to clearly distinguish the operators.

3.10 Contextual Awareness

Computational awareness is awareness of self, but contextual awareness is awareness of the environment around self. As humans, we label our perceptions of the external world (and our internal world as well) with descriptive names or phrases. For example, there is no “red” outside us, only different wavelengths of light. And, to answer the age old question, there is no sound when an unobserved tree falls in the forest, only vibrations of molecules in the air. Through our human sensory mechanisms, we perceive these phenomena and label them.

The idea of contextual awareness arose from ubiquitous computing or pervasive computing where changes in the environment initiated changes in a device or computer system that affected its response to its environment. [13]. Over the past decade, it has become a major focus of research in mobile device technology. Contextual awareness requires sensing, reacting and affecting abilities. Autonomous, intelligent robots are a good example. Within information systems, autonomous, intelligence malware that seeks to identify and exfiltrate data is also an example.

An SO will receive inputs of new data from its external environment supplied in many ways. To support its computational awareness model, it must also develop and maintain a perceptual awareness of

its external environment and the effects of its actions and outputs to that external environment.

Table 11. Awareness: Issues & Challenges

<i>The Act of Forgetting:</i> What mechanisms should be defined to allow an SO to forget that it “knows” something, what are the criteria, and what is the ripple effect within the SO of “forgetting” something.
<i>The Act of Correcting:</i> What mechanisms should be defined to correct faulty data/information whether accepted as input or due to faulty processes or incomplete deductive, abductive and inductive mechanisms.
<i>The Act of Knowing:</i> What mechanism will allow an SO to know what it knows and what it doesn’t know? How will this knowledge be represented?
<i>The Act of Understanding:</i> As with knowing, how will an SO assess the assimilation of new knowledge that improves its understanding of a situation?
<i>The Act of Discovery:</i> An SO, knowing what it does know, must be able to discover data. One method is to know of and be able to query external data sources. Another method is to be able advertise what its data needs are to its external environment.

4. Approaches to Research Questions

Our research questions represent complex problems that are not easily answerable, if at all, in a few paragraphs. Thus, we examine several aspects of each question and reserve the right to address these and additional questions in future work. Our research questions were:

(Q1) What technologies will enable augmented intelligence in objects?

Section 3 has identified several classes of technologies that will enable Smart Objects. We selected these technology classes as implementation mechanisms for the elements of the Smart Objects definition. Table 12 maps the definition elements to the technology classes.

Table 12. Technology Classes for SO Elements

Self-defining	Semantic Annotation Self-defining Representation
Self-reflection	Computational and Contextual Awareness Autonomous Operation
Self-modification	Artificial Intelligence Machine Learning

An example of an application using Smart Objects may help to understand how these technologies can be used in an implementation.

Apple's Siri, a virtual assistant, provides an individual – from kids to adults – with a helper that can perform all sorts of tasks. One can ask questions from the mundane, such as “what is your favorite dog?” to the profound, such as “what is free will?” Within the limits of its programming, Siri answer these as best as it can. This ability to try to help is essential to the successful adoption of personal assistants such as Siri, Amazon Echo, and Microsoft's Cortana.

What these systems lack, however, is autonomy – they are reactive rather than proactive, generally; they are not self-reflective, and they show a modicum of learning.

Suppose we could extend these systems with Smart Objects? What are examples of such technology evolution? First, they would be autonomous and symbiotic – meaning they continually operate to learn about things in which their user had shown interest, such as stock market forecasts. Second, they would evolve to infer other areas of interest from the topics the user had previously asked about. Third, they would develop and continually refine models of their user and of themselves in order to critique their reasoning, improve their tactics and strategies for answering questions and performing tasks; and suggest to their user actions that should be taken to accomplish goals inferred from user interaction.

(Q2) What are the implications of using Smart Objects to extend the capabilities of our organizational systems to exhibit intelligence?

The evolution of technology – software, hardware, and networks – underlying large-scale applications has increased the complexity of such systems faster than our ability to manage and maintain them. The Smart Object abstraction will embed more functionality that can support maintenance and evolution of applications within the object itself. As a result, we believe that, while a different skill set will be required to produce SO, overall reusability will be increased and personnel resources may be decreased. A significant learning phase will be required to understand and adapt this technology to the design, development and deployment of new systems. Further, a robust framework and infrastructure will be required to develop, implement, test, and deploy Smart Objects. This infrastructure will need to support consistent object persistence beyond the execution of an application.

Philosophically, there is a deeper question to address: Will Smart Objects be “conscious” given the integrated set of enabling technologies that somehow

create it. Franklin [4] began to address this in his book, *Artificial Minds*. His recent work on IDA (www.ccrp.memphis.edu) focuses on creating a cognitive entity that he suggests may be the first artificially conscious entity. Absent good criteria for “consciousness” or “conscious behavior”, we cannot agree with this claim. It is an open research topic on what it means to be “conscious” in humans and is equally difficult to define in machines.

The original Turing Test focused on whether a computer program could emulate human conversation without revealing itself. Over the past decade, new thresholds for the Turing Test have emerged with the success of several applications, such as Apple's Siri, IBM's Watson, etc. We are not suggesting that instances of SOs be able to hold conversations to pass the Turing Test, but we are suggesting that applying modern Turing concepts to SOs will aid in the identification of metrics to validate its concepts.

(Q3) Are there (perhaps un-proposed) theories of data (such as number or set theories) that will aid in our understanding of the characteristics and properties of Smart Objects?

This research question remains unaddressed in this paper. We include it to suggest that other researchers seek to discover as yet unknown ontologies and relationships between varying Big Data and Big Data sets, to envision tools that can be better used to understand Smart Objects and to prove the usefulness of the relationships in the new phenomena.

5. Conclusion and Future Work

We believe that it is time to focus on the next evolution of Big Data –the Smart Objects Paradigm. To that extent we have given a specific definition that is both functionally and technically focused. As with object-oriented programming, the software architecture of complex application systems will be reduced by encapsulating data and functionality in SOs. Several advantages will accrue to using SOs as depicted in table 12. WE surmise that SOs will have properties analogous to the concept of “self.”

A key question is “what happens to systems and programs?” Are SOs a replacement for the systems of today? We do not think so. Rather, SOs lead to new software paradigms [6] for constructing more capable, intelligent and autonomous complex systems which represent the next step beyond Big Data passivity. As previous paradigm shifts and technology developments have shown, adoption and adaptation will be anything but seamless and automatic.

Significant turmoil in the marketplace and the workplace can be expected.

Adoption and usage of Smart Objects will be predicated on business managers understanding how they can contribute to business operations and the bottom line. Table 13 proposes some advantages of Smart Objects.

Table 13. Advantages of Smart Objects

<i>Proactive v reactive:</i> An SO would have a lifecycle that existed beyond the scope of any specific instance allowing it to change processing of future instances based on out of band predictive analytics.
<i>Collaborative:</i> SOs would seek other smart objects from a discoverable set of SOs to extend or expand their problem solving ability through shared inputs and outputs as well as an ability to selectively iterate on a problem.
<i>Seek efficiency:</i> SOs would be self-limiting when solving complex problems using time, cost, and statistical data to determine recursion depth and scale of brute force calculations.
<i>Hide complexity:</i> SOs present simple interfaces for complex processing. “Find nearest foo”, expressed as FindFoo(near), internally could perform a geospatial calculation against historical selections and queries to public sources.
<i>Conversational, not directed:</i> SOs may request additional information from their users on systems available to them. By creating an event, that other smart objects, user proxies, or subsystems could respond to, they could use the newly discovered input in another iteration.

Thus, organizations need to understand SO technology and develop a compelling case for making this strategic investment. They need to understand where the best value lies for early implementations until a cadre of skilled personnel is trained, deployed and provided with appropriate support mechanisms. They need to show how implementation will unfold and how to measure SO’s value to business operations using proof-of-concept projects. And, they need to focus on a few key metrics that capture the expected value to the business operations and map those against various types of investments.

We are at the cusp of a Smart Objects Revolution in which active data will take a leading role in its analysis, management and response to supporting business operations. SOs will only comprise a small portion of the total data available, but will represent the most complex data required to support large, complex problem solving.

We are accompanying this paper at HICSS-51 with a half-day tutorial on Smart Objects that will describe and demonstrate a prototype using SOs.

Acknowledgement: Several colleagues have contributed to the formulation of these ideas through discussion and other interactions. We also thank the anonymous reviewers for their insightful comments which forced us to rethink the structure of the paper.

6. References

- [1] Asimov, I. 1950. *I, Robot*, Doubleday & Company, New York, NY
- [2] Benjamins, V. R., J. Contreras, O. Corcho, and A. Gomez-Perez. 2002. “Six Challenges for the Semantic Web”, KR2002 Semantic Web Workshop
- [3] Devanur, N. R. and L. Fortnow. 2009. “A Computational Theory of Awareness and Decision Making”, *Theoretical Aspects of Rationality and Knowledge (TARK)*, pp. 99-107
- [4] Franklin, S. 1995. *Artificial Minds*, MIT Press, Cambridge, MA
- [5] Halpern, J. 2001. “Alternative Semantics for Unawareness”, *Games and Economic Behavior*, 27(2):321-339
- [6] Kaisler, S. 2005. *Software Paradigms*, John Wiley and Sons, New York, NY
- [7] Kaisler, S., F. Armour, A. Espinosa, and W. Money. 2013. *Big Data: Issues and Challenges Moving Forward, 46th Hawaii International Conference on System Sciences*, Maui, HI, IEEE Computer Society
- [8] Klein, G.; Moon, B; Hoffman, R.R. (2006). "Making sense of sensemaking 1: Alternative perspectives". *IEEE Intelligent Systems*. 21 (4): 70–73
- [9] Kortuem, G., F. Kawsar, D. Fitton, and V. Sundramoorthy. 2010. “Smart Objects as the Building Blocks for the Internet of Things”, *IEEE Intelligent Computing*, 14(3):44-51.
- [10] Russell, S. and Norvig, P. 2010. *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ.
- [9] Rusu, A. A., N. C. Rabinowitz, G. Desjardins et al. 2016. *Progressive Neural Networks*, arxiv.org/abs/1606.04671
- [11] Simon, H. 1982. *Models of Bounded Rationality*, MIT Press, Cambridge, MA.
- [12] Wang, Y. 2007. “Theoretical Framework of Cognitive Informatics”. *International Journal of Cognitive Informatics and Natural Intelligence*, 1(1), 1–27.
- [13] Weiser, M., R. Gold, and J. S. Brown, 1999. “The Origins of Ubiquitous Computing Research at PARC in the late 1980s”, *IBM Systems Journal*, 38(4)
- [14] Zhao, Q. 2013. “Computational Awareness: Another Way towards Intelligence”, *Studies in Computational Intelligence*, Spring-Link, Vol. 465, pp. 3-10.