

An Ontological Approach to Focusing Attention and Enhancing Machine Perception on the Web

Cory Henson*, Krishnaprasad Thirunarayan, Amit Sheth

Kno.e.sis – Ohio Center of Excellence in Knowledge-enabled Computing

Wright State University, Dayton, Ohio, USA

{cory, tkprasad, amit}@knoesis.org

Abstract. Today, many sensor networks and their applications employ a brute force approach to collecting and analyzing sensor data. Such an approach often wastes valuable energy and computational resources by unnecessarily tasking sensors and generating observations of minimal use. People, on the other hand, have evolved sophisticated mechanisms to efficiently perceive their environment. One such mechanism includes the use of background knowledge to determine what aspects of the environment to focus our attention. In this paper, we develop an ontology of perception, IntellegO, that may be used to more efficiently convert observations into perceptions. IntellegO is derived from cognitive theory, encoded in set-theory, and provides a formal semantics of machine perception. We then present an implementation that iteratively and efficiently processes low level, heterogeneous sensor data into knowledge through use of the perception ontology and domain specific background knowledge. Finally, we evaluate IntellegO by collecting and analyzing observations of weather conditions on the Web, and show significant resource savings in the generation and storage of perceptual knowledge.

Keywords. Perception, Observation, Sensor, Ontology, Semantic Web

* Corresponding Author (377 Joshi Research Center, 3640 Colonel Glenn Highway, Dayton, OH 45435 (+1 937-775-5217))

1. Introduction

Look at the image in Figure 1. How quickly were you able to realize the identity of the depicted object? The activity you just engaged in is called perception; and while people are able to perceive their environment almost instantaneously, and seemingly without effort, machines continue to struggle with the task. In the present paper, we investigate how people are able to perceive the world so effectively and show how an approximation of this process can be formalized to better enable machines to perceive.



Figure 1. A red apple.

The study of perception likely began in ancient Greece when Plato first pondered the meaning of shadows dancing on a wall.¹ The Greek word for perception, *intellego*, stems from the Latin *intellegere*: *inter* ("between") and *lego* ("to choose or gather") (Norwich, 1991). To the ancient Greeks, to perceive was to choose from among alternative explanations which account for our observations. Thus, to perceive an apple is to choose from among a range of possibilities, including an apple, orange, or ball. These acts of observation and perception provide the building blocks for all human knowledge (Locke, 1690); they are the processes from which all ideas are born; and the sole bond connecting ourselves to the world around us. Now, with the advent of sensor networks² capable of observation, this world may be directly accessible to machines. Missing from this vision, however, is the ability of machines to glean semantics from observation; to apprehend entities from detected qualities; *to perceive*. The systematic automation of this ability is the focus of machine perception – the ability of computing machines to sense and interpret the contents of their environment (Nevatia, 1982). Despite early successes within narrow domains (e.g., facial recognition (Zhao, Chellappa, Phillips, & Rosenfeld, 2003)), however, a general solution remains elusive. *This state of affairs is the result of difficult research challenges, such as the ability to effectively model the process of perception, to provide an appropriate interpretation of observational data with incomplete information, and to efficiently interpret the growing stream of observational data.* The issue of effectively modeling the process of perception is often investigated within specific application areas, such as machine vision (Aloimonos, Weiss, & Bandyopadhyay, 1988; Diamant, 2007). While much progress has been achieved, this approach also results in a

¹ Plato's Cave, http://en.wikipedia.org/wiki/Allegory_of_the_Cave (accessed May, 2011)

² A sensor network is a group of specialized measuring devices (i.e., sensors) with a communications infrastructure intended to monitor and record conditions within an environment.

fractured assortment of models and algorithms that are effective for narrowly defined problems, such as interpreting sensor data of a single modality. Integrating and interpreting sensor data of multiple modalities for a wide range of applications, however, requires a more encompassing approach. In an attempt to deal with the latter issue, of efficiently interpreting the growing stream of observational data, there is much research within the sensors community to mitigate the effects of observational data overload. Many such efforts concentrate on developing effective schedules and sampling rates for sensor observations (Gürgen, Roncancio, Labbé, & Olive, 2006). Interval-based sampling often generates data (through *brute-force* collection) that is unnecessary for understanding the environment. Another technique for collecting sensor data, called event-based sampling (Pawlowski et al., 2008, 2009), generates observation records only after a particular event is detected (e.g., temperature drops below a certain threshold). Unfortunately, this provides limited additional benefits towards effective analysis of (often incomplete) data. The dichotomic need for reduced information overload and complete information for effective analysis can be addressed through an understanding, and modeling, of the techniques employed by human perception. *More specifically, the techniques employed by human perception, such as the ability to focus attention and seek out additional information from our environment, holds the key to simultaneously minimizing the amount of information needed for perception and enabling graceful degradation of perception with incomplete information.*

Increasingly, sensors and sensor data are being made accessible on the Web (Aberer, Hauswirth, & Salehi, 2006; Botts, Percivall, Reed, & Davidson, 2008; Sheth, Henson, & Sahoo, 2008). In 2008, there were an estimated 40+ billion sensors connected to mobile devices (Nokia, 2008); and it has been predicted that within the next five years, sensor data from such devices will become the dominant type of information on the Web (Higginbotham, 2010). With this future ahead, any proposed solution to the issue of sensor data management and/or interpretation of sensor data should be highly scalable (at Web scale) and able to integrate with standard Web languages.

We propose to address such challenges, as discussed above, through the development of an ontology of perception, *IntellegO*. This ontology is derived from well-established cognitive theories of perception and establishes a formal semantics for machine perception. Before introducing IntellegO, we will first explore and draw inspiration from several cognitive theories of perception (Section 2). With this background, the concepts, relations, and processes of IntellegO are described and formally defined (Section 3). To demonstrate the utility of IntellegO, we will employ the ontology to interpret sensor data on the Web, that is encoded in the Resource Description Framework (RDF) (Manola & Miller, 2004) language. In order to integrate with this sensor data, we will use the

Web Ontology Language (OWL) (Hitzler, Parsia, Patel-Schneider, & Rudolph, 2009) to map terminology from IntellegO to a standard sensor ontology (Section 4). Next, we will provide three evaluations of IntellegO (Section 5). In the first, we evaluate the sensing resources required for generating observations and perceptions, and show how focus (of attention) can lead to improved efficiency. In the second, we evaluate the expressivity of IntellegO along two dimensions, including (1) the ability to degrade gracefully with incomplete information, and (2) the ability to improve explanations of observational data based on new information. Specifically, IntellegO's capacity to embody these abilities is compared with current approaches, such as the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) and first-order logic. In the third, we evaluate the resources required for storing sensor observations and perceptions, and show that for some applications, perception can lead to significant resource savings. We will finish with a description of related works (Section 6) and concluding remarks (Section 7).

Additional details of IntellegO, including all ontologies and datasets used in the evaluation – to allow validation, repeatability, and further experimentation – are accessible at: <http://wiki.knoesis.org/index.php/Intellego>.

2. Background

What are perceptions and how are they formed? What can be perceived? How do perceptions relate to reality? In order to bestow onto a machine the ability to glean semantics from observation, such questions require explicit, implementable, answers. As perceptual beings, people are constantly inundated with sensory input; yet we are able to make sense out of our environment with relative ease. We have a remarkable aptitude for comprehending the world around us; for subconsciously analyzing sensory input and apprehending mental conceptions with efficiency and precision. For centuries, thinkers have endeavored to understand the mechanisms underlying this phenomenon, and through such investigation have advanced complex theories of perception within the fields of philosophy, psychology, physiology, cognitive science, and machine vision. One idea that has emerged from such investigations includes the ability to utilize background knowledge to determine what aspects of the environment to focus our attention (Bajcsy, 1988; Gregory, 1968, 1997; Neisser, 1976), which enables a perceiver to efficiently make sense of the environment.

In the late 1970's, the idea of perception as a cyclical process began to take form. The most famous expression of this idea was proposed by Ulric Neisser in 1976 (Neisser, 1976). Neisser's Perception Cycle is divided into three stages: (1) sampling (or observing) the environment, (2) modifying our knowledge (schema) of the environment, based on our newly acquired observations, and (3) directing our attention for further exploration.

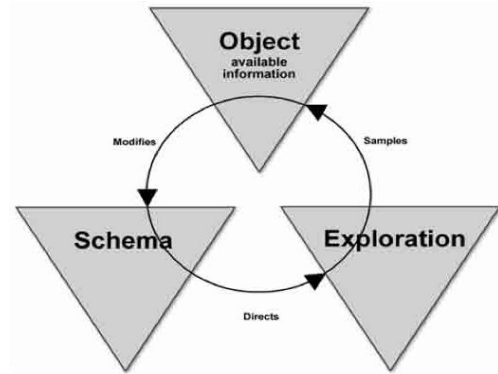


Figure 2. Neisser's Perception Cycle.

Figure 2 shows a graphical representation of Neisser's Perception Cycle. In contemporary research, this general model is called Active Perception (Bajcsy, 1988). The idea of perception as an active, cyclical process laid the ground work for our current understanding of human perception. Shortly after, from the early 1980's and into the late 1990's, Richard Gregory's theories of perception began to take shape and were continually refined through investigations into the nature of illusions (Gregory, 1997). From this work, he showed that the perception cycle iteratively generates and tests hypotheses that explain our observations. As we progress through these tests, by focused attention on our environment, the number of distinct explanatory hypotheses invariably diminishes; leading to more precise, unambiguous hypotheses. In addition, Gregory was able to demonstrate that the hypothesize and test cycle is driven by a-priori background knowledge. In other words, our perception of the world is highly dependent on our knowledge of the world. More recently, the term *top-down processing* has been used to describe the ability to map observations onto background knowledge in order to *fill in the gaps* of this knowledge. This phenomenon has been widely studied within the fields of biological/human vision (Cavanagh, 1999; Gregory, 1997) and machine vision (Aloimonos et al., 1988; Diamant, 2007). Meanwhile, in the early 1990's, another unusual model of perception emerged from the field of mathematics. At this time, James Gibson's ideas on perception were gaining influence; in particular, the idea that physical stimuli carry *information* about the world, and our sensory organs have evolved to intercept, extract, and decode this information (Gibson, 1966). From this general idea, Peter Norwich was able to extrapolate the conjecture that the stimulus information intercepted by our senses could be measured and understood using Claude Shannon's Information Theory (Shannon, 1948). This is the basis for the Entropy Theory of Perception which uses mathematical models of entropy and information-gain to determine the informational value of different observations (Norwich, 1991).

Inspired by such theories, this work provides an explicit formalization of perception that may be understood and systematically executed by machines. The formalization is based on three core ideas:

1. Perception is an active, cyclical process of exploration and interpretation (Nessier).
2. The perception cycle is driven by background knowledge in order to generate and test hypotheses (Gregory).
3. In order to effectively test hypotheses, some observations are more informative than others (Norwich).³

We synthesize these ideas and view perception as an efficient, cyclical process of actively seeking and detecting those qualities that carry information most useful for testing and evaluating hypothesis, as discussed below.

3. Ontology of Perception

Over the years, cognitive theories of perception have been proposed, evaluated, revised, and evolved within an impressive body of research. This research presents a valuable stepping-stone towards the goal of machine perception, to embody this unique human ability within a computational system. In this section, we aim to explicitly define the information processes involved in perception that will serve as an ontological account of knowledge production. The ontology of perception, or *IntellegO*, attempts to formally model perception in a way that is independent of any particular implementation technology and of suitable generality to encompass both machine perception and human perception.⁴ A formal semantics⁵ of perception can be defined by providing high-level interpretation of low-level *observational* data, which may be derived through computational means.

³ To our knowledge, Norwich never actually made this connection between his ideas of measuring the informational value of observations and Gregory's ideas of testing hypotheses. This connection stems from our own imagination, and plays a critical role within the ontology of perception.

⁴ We are NOT claiming to represent the full spectrum of human perception, but have tried to include ideas from cognitive theory of perception that may be useful for machine perception.

⁵ Formal semantics is a rigorous, systematic, and unambiguous description of the meaning of some conceptualization, described in purely symbolic terms; and often embodied within a set of logical axioms and entailments rules.

To communicate the semantics of perception, we must define an appropriate terminology. In the text below, we will describe several concepts, relations, and processes. The concepts and relations include *entity*, *quality*, *quality-type*, *percept*, *observer*, *perceiver*, *focus*, *perceptual-theory*, *inheres-in*, and *has-type*. These concepts and relations are described in Section 3.1. The processes include *observation-process*, *perception-process*, and *perception-cycle*. These processes, along with several sub-processes, are formally defined in Section 3.2. Table 1 provides a quick reference guide to the terminology of IntellegO.

Table 1. Quick reference guide to the terminology of IntellegO.

Term	Description	Example
entity	An object or event in the world	apple
quality	An inherent property of an entity	red
quality-type	A category (or class) of qualities	color
percept	A quality that has been detected	<i>red</i>
observer	An agent that executes the observation-process	sensor
perceiver	An agent that executes the perception-process	computer
focus	A quality-type whose detection may reduce the perceptual-theory	color
perceptual-theory	A set of entities that each explains a set of percepts	{apple, nose}
inheres-in	A relation between a quality and an entity	red inheres-in apple
has-type	A relation between a quality and a quality-type	red has-type color
observation-process	An act of detecting a quality and generating a percept	observation-process(red) → <i>red</i>
perception-process	An act of inferring a perceptual-theory from a set of percepts	perception-process(red) → {apple, nose}
perception-cycle	An act of minimizing a perceptual-theory by focusing attention	perception-cycle(...) → {apple}

3.1. Semantics of Perception: Concepts and Relations

Imagine again that you are looking at the red apple depicted in Figure 1. The apple is an *entity* and red is a *quality*. The red quality is an inherent property of the apple entity. An *entity* is an object or event in the world; a *quality* is an inherent property of an entity; and *inheres-in* is a relation between a quality and an entity. Figure 3 illustrates an abstract set of *inheres-in* relations. A *quality-type* is a named category, or class, of qualities; such as color. A quality must have one-and-only-one quality-type;

has-type is a relation between a quality and a quality-type. Different qualities associated with the same quality-type (through the *has-type* relation) are mutually exclusive. For example, red and green are mutually exclusive for the quality-type color. The background knowledge needed for perception, termed *perceptual-BK*, is composed of a set of *has-type* relations between qualities and quality-types, and a set of *inheres-in* relations between qualities and entities.

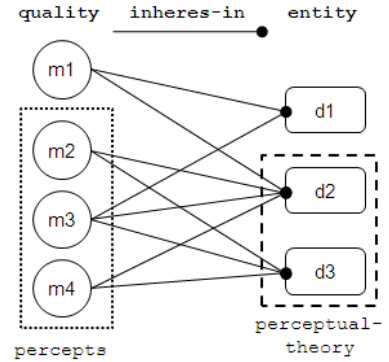


Figure 3. Set of inheres-in relations.

An *observer* is an agent that detects a quality. In this case, the observer is a human eye detecting the color red; however, the role of observer can be played by mechanical agents (e.g., sensors), biological agents (e.g., human eyes), or social agents (e.g., micro-blogs (Sheth, 2009)). Upon detecting the color red, the mind brings forth an experience of redness embodied within an experience of an apple (of *appleness*). These mental experiences are referred to as *qualia*. While there is a clear distinction between a quality, or entity, and its associated *qualia*, we will make no such distinction in IntellegO.⁶ A *percept* is a quality that has been detected by an observer. A *perceiver* is an agent that generates explanations for a set of *percepts*; for example, an apple may explain the red percept. In this case the perceiver is a human mind; however, the role of perceiver can be played by mechanical agents (e.g., computers) or biological agents (e.g., human minds) (Goldstine, 1964). An explanation of a set of *percepts* is a set of entities, termed a *perceptual-theory*. Specifically, each entity in the *perceptual-theory* accounts for (explains) all the *percepts*. In order to refine or minimize a *perceptual-theory*, a perceiver may provide instructions to an observer to detect a particular quality-type. The phrase “detect a particular quality-type” should be interpreted as the detection of a member quality. When this occurs, the quality-type is termed *focus*. This ability to refine a *perceptual-theory* by employing *focus* is the key to efficient perception and will be further discussed in Section 3.2. Figure 4 provides an example of how qualities, quality-types, entities, *percepts*, and *perceptual-theories* are related.

⁶ While such a distinction may more accurately reflect, or approximate, human perception, in the authors’ opinion it would also complicate IntellegO without adding sufficient utility.

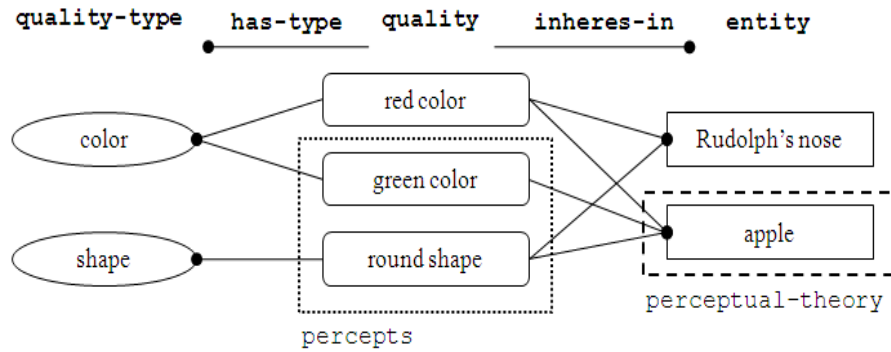


Figure 4. Example of how qualities, quality-types, entities, percepts, and perceptual-theories are related. The green color and round shape qualities have been detected and can be explained by the apple entity. Rudolph's ("the red-nosed reindeer") nose is not a member of the perceptual-theory since it cannot explain a set of percepts containing the green color quality.

3.2. Semantics of Perception: Processes

The three primary processes of IntellegO are: *observation-process*, *perception-process*, and *perception-cycle*. These processes are formally specified in set-theoretic notation. We have chosen to formalize the semantics of IntellegO in this manner because set-theory provides a notation that is unambiguous, well-established, and suitably expressive. Before these processes are defined, Table 2 provides a few required preliminary definitions.

Table 2. Required definitions for formalizing processes in IntellegO.

Term	Description
perceptual-BK = $\langle Q, E, I, QT, T \rangle$	Background knowledge about qualities, entities, and their relationships
Q	Set of all qualities
E	Set of all entities
$I \subseteq (Q \times E)$	Set of all inheres-in relations between qualities and entities
QT	Set of all quality-types
$T \subseteq (Q \times QT)$	Set of all has-type relations between qualities and quality-types
$P \subseteq Q$	Set of all percepts

3.2.1. Observation Process

While looking at the image in Figure 1, your eye detects the color red and generates an abstract representation in

your mind. This is an example of an *observation-process*. Given a *quality-type* as input, *observation-process* returns a detected quality, termed a *percept*. The *observation-process* represents an interface between an agent and the outside world. While we can define the input and output parameters of this process, the method in which a *quality* is detected is highly application dependent. For example, in many application scenarios the *observation-process* would activate a transducer which interacts with some physical stimuli; and this interaction is then interpreted as the detection of some *quality* in the world (Kuhn, 2009). For the application presented in this paper, the sensor data has already been collected, encoded in RDF, and made accessible on the Web. Therefore, in order to determine which *qualities* are detected, the *observation-process* generates and executes queries against the sensor data on the Web (Section 5.1.2). For this reason, only the input and output parameters of *observation-process* are fully specified.

Definition: *observation-process* ($QT \rightarrow Q$)

observation-process(qt) = p, where $(p \in Q) \wedge (qt \in QT) \wedge ((p, qt) \in T) \wedge$
 "p is detected"

A set of *qualities* is considered *valid* if the set contains at most one *quality* associated with each *quality-type*. This validity check reflects real-world constraints on a set of *percepts* and the nature of the background knowledge.

Definition: *valid* ($Q \rightarrow \text{Boolean}$)

valid(ps) $\Rightarrow (ps \subseteq Q) \wedge (\forall p1, p2 \in ps : (p1 \neq p2) \Rightarrow (\exists qt1, qt2 \in QT :$
 $((p1, qt1) \in T) \wedge ((p2, qt2) \in T) \wedge (qt1 \neq qt2))$

3.2.2. Perception Process

Again, while looking at the image in Figure 1, after detecting the color red, your mind attempts to explain the red color and generates an abstract representation of this explanation in your mind. An *entity explains* a set of *qualities* if the set of *qualities* are *valid* and each *quality* is an *inherent property* of the *entity*.

Definition: explains ($E \times \text{Powerset}(Q) \rightarrow \text{Boolean}$)

$\text{explains}(e, ps) \Leftrightarrow (e \in E) \wedge (ps \subseteq Q) \wedge \text{valid}(ps) \wedge (\forall p \in ps : (p, e) \in I)$

The process of generating explanations for a set of qualities is called the *perception-process*. The perception-process takes a set of qualities as input and yields a set of entities capable of explanation; that is, each entity in the set explains the set of qualities. The set of entities generated by a perception-process is called the *perceptual-theory*. In practice, the perception-process should attempt to explain only a set of percepts.

Definition: perception-process ($\text{Powerset}(Q) \rightarrow \text{Powerset}(E)$)

$\text{perception-process}(ps) = \{ e \in E \mid \text{explains}(e, ps) \}$

3.2.3. Perception Cycle

The perception-process generates a perceptual-theory containing entities that explain a set of percepts. Notice that the perceptual-theory can contain multiple entities. This is not an ideal situation. When you look at Figure 1, you should perceive an apple, not an apple and/or Rudolph's nose. There are examples of human perception, however, where this type of ambiguity prevails. Consider the image in Figure 5: does this image depict a cup, or two human faces? While such ambiguity may not always be ideal, the ability of the



Figure 5. Is this a cup, or two human faces?

perceptual-theory to contain multiple explanatory entities is also what enables handling of incomplete or missing information (i.e., enables graceful degradation). For example, now (with incomplete information) we say that the image in Figure 5 depicts either a cup or two human faces; our perceptual-theory is {cup, two human faces}. However, if we were to subsequently detect eyes and/or ears, then (with additional disambiguating information) our perceptual-theory is refined to {two human faces}. The study of perceptual illusions has provided significant insights into the nature of perception (Gregory, 1968, 1997). Notwithstanding such cases, in general, we would like the resulting perceptual-theory to contain as few entities as possible; ideally, exactly one. This is because, in general, specificity improves action-ability. For example, the perceptual-

theory resulting from Figure 1 is {apple} rather than the ambiguous {apple, Rudolph's nose}. The size of a perceptual-theory can often be reduced (i.e., concepts can be removed/filtered out) through the detection of new qualities, which must subsequently be explained. This cyclical process of observation and perception is called the *perception-cycle*. The perception-cycle is a process that relates the observation-process and perception-process; or, rather, relates observers and perceivers. An observer communicates percepts to a perceiver, representing qualities that have been detected, and the perceiver communicates focus to the observer, representing quality-types that should be detected. Figure 6 provides a graphical representation of the perception-cycle.

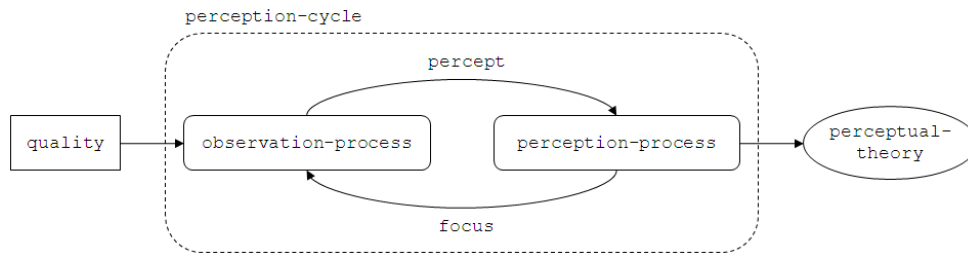


Figure 6. Architecture of the perception-cycle.

Within the perception-cycle, as new qualities are detected and the set of percepts grows, the size of the perceptual-theory shrinks. Thus, perception is an anti-monotonic process (that is, if f is a function that maps a set of percepts to a set of explanatory entities (i.e., perception-process), and x, y are sets of percepts, then $((x \subseteq y) \Rightarrow (f(x) \supseteq f(y)))$ (Gries, 1999). The anti-monotonic nature of the perception-cycle does not permit a straightforward formalization in first-order logic using standard deductive inference.

In order to optimize the perception-cycle, the observation-process should detect only those qualities that are capable of reducing the perceptual-theory. For example, if a perceptual-theory contains the entities apple and Rudolph's nose, detecting shape is probably of little use – i.e., cannot help discriminate between an apple and Rudolph's nose – since you can probably expect both to be round(ish). In order to clarify which qualities enable the reduction of the perceptual-theory, we will define four types of qualities: *expected*, *unknown*, *extraneous*, and *discriminating*. A quality is *expected* with respect to a set of entities if it is an inherent property of every entity in the set. Thus, if it were detected, it

would be explained by every entity in the set. By definition, all percepts are expected. For example, given the perceptual-theory $\{apple, Rudolph's\ nose\}$, then the quality round-shape would be expected since both an *apple* and *Rudolph's nose* are round.

Definition: expected $(Q \times \text{Powerset}(E) \rightarrow \text{Boolean})$

$\text{expected}(q, es) \Leftrightarrow (q \in Q) \wedge (es \subseteq E) \wedge \neg \text{empty}(es) \wedge (\forall e \in es : (q, e) \in I)$

A quality is *unknown* with respect to a set of entities if it is not an inherent property of any entity in the set. Thus, if it were detected, it would not be explained by any entity in the set. By definition, no percepts are unknown. For example, given the perceptual-theory $\{apple, Rudolph's\ nose\}$, then the quality square-shape would be unknown since neither an *apple* nor *Rudolph's nose* are square.

Definition: unknown $(Q \times \text{Powerset}(E) \rightarrow \text{Boolean})$

$\text{unknown}(q, es) \Leftrightarrow (q \in Q) \wedge (es \subseteq E) \wedge \neg \text{empty}(es) \wedge (\forall e \in es : (q, e) \notin I)$

A quality is *extraneous* with respect to a set of entities if it is either an inherent property of every entity in the set (expected), or is not an inherent property of any entity in the set (unknown). Thus, if it were detected, it would either be explained by all entities in the set or explained by no entities in the set. In either case, detection would not help to reduce the perceptual-theory. For example, given the perceptual-theory $\{apple, Rudolph's\ nose\}$, then the quality round-shape and the quality square-shape would both be extraneous (for different reasons).

Definition: extraneous $(Q \times \text{Powerset}(E) \rightarrow \text{Boolean})$

$\text{extraneous}(q, es) \Leftrightarrow \text{expected}(q, es) \vee \text{unknown}(q, es)$

A quality is *discriminating* with respect to a set of entities if its detection could potentially be used to reduce the size of the perceptual-theory. The set of discriminating qualities and the set of extraneous qualities are disjoint. Note that the set of discriminating qualities is not required to

be a valid set. For example, given the perceptual-theory $\{apple, Rudolph's\ nose\}$, then the quality green-color would be discriminating since an *apple* can be green while *Rudolph's nose* cannot.

Definition: discriminating ($Q \times \text{Powerset}(E) \rightarrow \text{Boolean}$)

$\text{discriminating}(q, es) \Leftrightarrow \neg \text{extraneous}(q, es)$

A perceptual-theory is *minimum* if it cannot be reduced through further observation. In other words, there are no qualities whose detection may discriminate between entities in the perceptual-theory.

Definition: minimum ($\text{Powerset}(E) \rightarrow \text{Boolean}$)

$\text{minimum}(es) \Leftrightarrow (\forall q \in Q : \text{extraneous}(q, es))$

With this terminology, we can now define a more specific goal of the perception-cycle: to generate a minimum perceptual-theory for a set of percepts. In order to achieve this goal efficiently, only those qualities capable of discriminating between entities in the perceptual-theory should be detected. To ensure only discriminating qualities are detected, a perceiver may provide instructions to (task) an observer to detect a particular quality-type, termed *focus*. This ability to refine a perceptual-theory by employing *focus* is the key to efficient perception. *Focus* is sent to an observation-process capable of detecting the represented quality-type. Given a set of entities (i.e., perceptual-theory), the *focus-candidates* process returns a set of quality-types which, when detected, can lead to reducing the perceptual-theory.

Definition: focus-candidates ($\text{Powerset}(E) \rightarrow \text{Powerset}(QT)$)

$\text{focus-candidates}(es) = \{ qt \in QT \mid (\exists q \in Q : (q \notin P) \wedge \text{discriminating}(q, es) \wedge ((q, qt) \in T)) \}$

If there are several quality-types capable of reducing the perceptual-theory – i.e., several focus candidates – only one (at-a-time) may be designated as *focus*. The *choose* process takes a set of quality-types as input and returns a quality-type to observe. The method in which a single quality-type may be chosen from a set of quality-types is highly application dependent. For this reason, only the input and output

parameters of the `choose` process are fully specified. In Section 5.1, we evaluate several different implementations of the `choose` process.

Definition: `choose` ($\text{Powerset}(\mathcal{QT}) \rightarrow \mathcal{QT}$)

`choose(qts) = qt`, where $(qts \subseteq \mathcal{QT}) \wedge (qt \in qts) \wedge$ "one qt is chosen"

The *perception-cycle* generates the minimum perceptual-theory. The process begins with the set of all known *entities* and an empty set of *percepts*, and repeatedly seeks suitable observations to better assess the situation. The resultant set of *entities* is progressively refined, to eventually obtain the minimum perceptual-theory. This perceptual-theory represents the best possible explanation(s) admissible by the given background knowledge (*perceptual-BK*). The *perception-cycle* algorithm proceeds as follows: (1) begin with a perceptual-theory and a set of percepts; (2) if the perceptual-theory is minimum then return; otherwise (3) generate and send focus to an observation-process and add the detected quality to the set of percepts; and finally, (4) update the perceptual-theory by removing those entities which cannot explain the updated set of percepts, and recursively continue the *perception-cycle*.

Definition: `perception-cycle` ($\text{Powerset}(E) \times \text{Powerset}(P) \rightarrow \text{Powerset}(E)$)

Algorithm

```
[1] input:  $es \subseteq E, ps \subseteq P$ 
[2] if minimum( $es$ ) then return  $es$ 
[3] else let  $aux = ps \cup \{\text{observation-process}(\text{choose}(\text{focus-candidates}(es)))\}$ 
[4]     in perception-cycle(perception-process(aux), aux)
```

Function Call

```
[1] initialize:  $es = E, ps = \{\}$  //  $\{\}$  is the empty set
[2] perception-cycle(es, ps)
```

4. Integration with Sensor Data on the Web

The ability to effectively analyze sensor data through the process of perception is becoming a critical task; and increasingly, much of this sensor data is being made accessible through the Web (Aberer, Hauswirth, & Salehi, 2006; Botts, Percivall, Reed, & Davidson, 2008; Sheth, Henson, & Sahoo, 2008). To enable the representation and access to sensors and sensor data on the Web, the Open Geospatial Consortium (OGC) has developed the Sensor Web Enablement (SWE) framework, providing XML-based languages and Web-service specifications to manage sensor resources (Botts et al., 2008). Increasingly, however, data on the Web is being encoded in the Resource Description Framework (RDF) language (Manola & Miller, 2004). RDF is a graph-based machine readable format that allows data to be interlinked across the Web. The publishing of data in this form is called Linked Data, and the result is a large graph called the Linked Open Data (LOD) Cloud (Bizer, Heath, & Berners-Lee, 2009). The SWE framework has been further extended in a Semantic Sensor Web (Sheth, Henson, & Sahoo, 2008), in which services provide and utilize semantically annotated sensor data that reference appropriate spatial, temporal, thematic (domain), and sensor ontologies (Corcho & García-Castro, 2010; Henson, Pschorr, Sheth, & Thirunarayan, 2009; Janowicz et al., 2010). Sensor data is also being published as Linked Data. A recently published sensor dataset called *LinkedSensorData* includes descriptions of over 20,000 active sensors and over 160 million sensor observations; resulting in over 1.7 billion⁷ facts (statements) related to major weather events in the United States (Patni, Henson, & Sheth, 2010). In order to better share and unambiguously interpret such sensor data on the Web, the World Wide Web Consortium developed a standard ontology for describing sensors and sensor observations called the Sensor and Sensor Network (SSN) Ontology (Lefort et al., 2011). The SSN ontology is formalized in the Web Ontology Language (OWL) (Hitzler et al., 2009), and the RDF sensor data on LOD conforms to this ontology.

We would like to be able to use IntellegO to generate *perceptual-theories* that explain this sensor data on the Web. To enable this task, we must first understand the relationships between the terminology defined in IntellegO and the terminology used to annotate the sensor data encoded in RDF. These relationships can be expressed through a mapping of terms between IntellegO and the SSN ontology. This task is generally called ontology alignment, or ontology matching (Euzenat & Shvaiko, 2007). In order to express such a mapping, we encode the terminology of IntellegO in OWL and provide appropriate mapping relations to terms in the SSN

⁷ As of the time of this writing: May, 2011

ontology. Note that since OWL is grounded in Description Logic (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003) – a monotonic, tractable subset of first-order Logic – and given the anti-monotonic nature of IntellegO, a straightforward formalization of the `perception-cycle` is beyond the expressivity of OWL. The mapping of terms from IntellegO-OWL to the SSN ontology is shown in Table 3.

Table 3. Mapping of concepts between IntellegO-OWL and the SSN ontology. Shaded rows represent concepts in IntellegO-OWL that are not associated with any concept in the SSN ontology.

IntellegO-OWL	mapping relation	SSN Ontology
<i>io:entity</i>	<i>owl:equivalentClass</i>	<i>ssn:Entity</i>
<i>io:quality</i>	<i>rdfs:subClassOf</i>	<i>ssn:Quality</i>
<i>io:quality-type</i>	<i>owl:equivalentClass</i>	<i>ssn:Quality</i>
<i>io:percept</i>	<i>rdfs:subClassOf</i>	<i>ssn:Quality</i>
<i>io:observer</i>	<i>owl:equivalentClass</i>	<i>ssn:Sensor</i>
<i>io:perceiver</i>		
<i>io:focus</i>	<i>rdfs:subClassOf</i>	<i>ssn:Quality</i>
<i>io:perceptual-theory</i>		
<i>io:inheres-in</i>	<i>owl:equivalentProperty</i>	<i>ssn:isQualityOf</i>
<i>io:has-type</i>		
<i>io:observation-process</i>	<i>owl:equivalentClass</i>	<i>ssn:Observation</i>
<i>io:perception-process</i>		
<i>io:perception-cycle</i>		

The descriptions of sensors, observations, etc., found in `LinkedSensorData` conforms to the SSN ontology (e.g., *ssn:observedProperty*). `LinkedSensorData` also contains terminology from relevant domain ontologies – such as an ontology describing weather,⁸ in this case. The weather ontology extends the SSN ontology. The `qualities` and `quality-types` found in `LinkedSensorData` conform to the weather ontology (e.g., *weather:temperature*). In order to appropriately interpret statements within `LinkedSensorData`, we utilize the mapping of terms from IntellegO-OWL to the SSN ontology. For example, the *owl:equivalentClass* mapping from *io:quality-type* to *ssn:Quality* may be used to interpret the term *weather:temperature* – found in an RDF statement from `LinkedSensorData` and defined in the weather ontology – as a `quality-type` in IntellegO. In the example below, a classification statement *weather:temperature rdfs:subClassOf io:quality-type* is inferred from a mapping statement *io:quality-type owl:equivalentClass ssn:Quality* and a classification statement *weather:temperature rdfs:subClassOf*

⁸ Weather Ontology, <http://sonicbanana.cs.wright.edu/ssw/ont/weather.owl>

ssn:Quality. To better understand why this inference occurs, (Hitzler et al., 2009) provides a description of the formal semantics of OWL.

$$\frac{\begin{array}{l} io:quality\text{-type owl:equivalentClass ssn:Quality} \\ weather:temperature rdfs:subClassOf ssn:Quality \end{array}}{\therefore weather:temperature rdfs:subClassOf io:quality\text{-type}}$$

Now, since *weather:temperature* is known to be a *io:quality-type*, it may be used as a valid input value of the *observation-process*. As a result of these mappings, an implementation of IntellegO can be used to interpret (i.e., generate *perceptual-theories* for) sensor data on the Web.

From the mapping in Table 3, we may notice that the distinction between a *quality* and a *quality-type*, as described in IntellegO, is not represented in the SSN ontology faithfully. As a result, both *io:quality* and *io:quality-type* (along with *io:percept* and *io:focus*) map to *ssn:Quality*. This omission results in a limited ability to unambiguously interpret quality data annotated with terms from the SSN ontology. In its defense, however, SSN is an upper-level ontology that is designed to be extended with additional domain specific terminology. In addition, SSN re-uses several concepts from the DOLCE foundational ontology (Borgo & Masolo, 2009), including the concept of quality (*ssn:Quality*). As such, to resolve the issue, the weather ontology described above re-uses concepts from IntellegO-OWL to appropriately distinguish between *qualities* and *quality-types* in the domain of weather. The connection between the ontologies and datasets described above is illustrated in Figure 7.

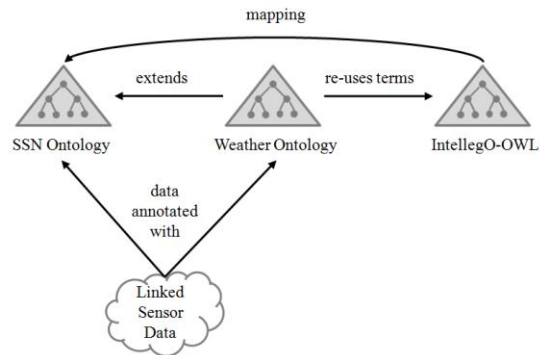


Figure 7. Connections between ontologies and datasets used by IntellegO.

5. Evaluation

In the following section, we provide three evaluations of IntellegO. In Section 5.1, we evaluate the sensing resources required for generating *perceptual-theories*, and show how *focus*, determined by the *perception-*

cycle, can lead to improved efficiency. In Section 5.2, we evaluate the expressivity of IntellegO along two dimensions: (1) the ability to degrade gracefully with incomplete information, and (2) the ability to minimize explanations based on new information. IntellegO’s capacity to embody these abilities is compared with current approaches, such as SWRL and first-order logic. In Section 5.3, we evaluate the resources required for storing sensor observations and perceptual-theories, and show that, for some applications, the generation and storing of perceptual-theories instead of raw observations can lead to significant – an order of magnitude – storage savings.

5.1. Focus Evaluation

The ability to *focus* attention enables a *perceiver* to more efficiently make sense of their environment. To better automate this process, IntellegO formalizes this ability. Specifically, we have implemented a prototype of IntellegO to run three experiments that demonstrate a realization of the *perception-cycle* and test the influence of *focus* on the interpretation of sensor data. Our metric of evaluation, used to compare the results of these experiments, includes the number of times the *observation-process* was executed in order to generate a minimum *perceptual-theory*. The number of times the *observation-process* is executed can also be represented by the size of the set of *percepts* to be explained (since each execution of the *observation-process* generates one *percept*). In the first experiment, we disable the *focus* ability of IntellegO and use background knowledge as discussed in Section 5.1.1. This is analogous to executing a brute force approach and serves as a base-line for comparison against subsequent experiments. This approach, however, is unfortunately the common *modus operandi* for collecting and interpreting data from sensor networks. In the second experiment, we enable the ability to *focus* and use the same background knowledge as the first experiment. In the third experiment, we enable the ability to *focus* and attempt to select an optimal *focus* using an enhanced background knowledge represented as a decision tree.

Claim – *The use of focus within the perception-cycle generates a perceptual-theory more efficiently (i.e., generates a smaller set of percepts) than the naïve brute force approach.*

Proof Sketch – Focus-candidates by definition discriminate between entities that can potentially serve as a viable explanation. Thus, observing a quality-type designated as focus is guaranteed to reduce viable explanations. On the other hand, observing quality-types other than the focus-candidates is of no consequence because either those values are already known (expected) or they are not relevant (unknown) given the current set of viable explanations. Thus, observing quality-types that are not focus-candidates requires wasteful computation and delays the determination of the minimum set of explanations.

Table 4 provides definitions used for the following experiments. The domain of interest is weather.

Table 4. Required definitions for evaluating IntellegO.

Term	Description
perceptual-BK = $\langle Q, E, I, QT, T \rangle$	Background knowledge about qualities, entities, and their relationships (see Figure 8)
Q	Set of all qualities = { <i>freezing-temperature, not-freezing-temperature, snow-precipitation, rain-precipitation, no-precipitation, high-wind-speed, low-wind-speed</i> }
E	Set of all entities = { <i>blizzard, flurry, rain-storm, rain-shower, clear</i> }
$I \subseteq (Q \times E)$	Set of all inheres-in relations (see Figure 8)
QT	Set of all quality-types = { <i>temperature, precipitation, wind-speed</i> }
$T \subseteq (Q \times QT)$	Set of all has-type relations (see Figure 8)
$P \subseteq Q$	Set of all percepts (generated during execution of the perception-cycle)

5.1.1. Background Knowledge

The background knowledge, or perceptual-BK, utilized by these experiments is encoded as a graph representing the relationships between qualities and their types, and between qualities and the entities in which they inheres-in. The domain of interest is weather; therefore, the background knowledge contains weather related entities, such as *blizzard, flurry, rain-storm, rain-shower, and clear*. Weather related inherent qualities of these entities include *freezing-temperature, not-freezing-temperature, snow-precipitation, rain-precipitation, no-precipitation, high-wind-speed, and low-wind-speed*. The quality-types include *temperature, precipitation, and wind-speed*. Figure 8 illustrates the background knowledge related to weather. The concepts originated from the National Oceanic and Atmospheric Association (NOAA)⁹ and are encoded in an ontology of weather.

⁹ NOAA, <http://www.noaa.gov/>

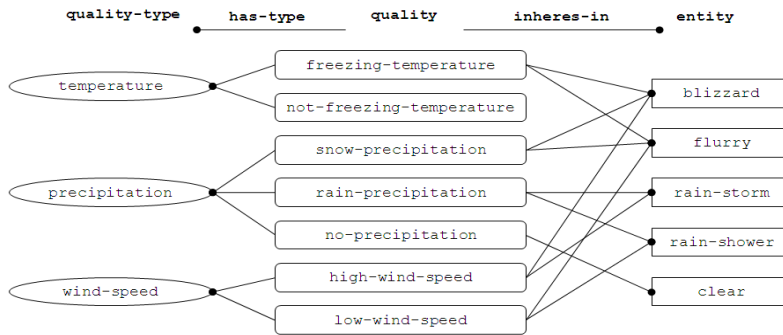


Figure 8. Background knowledge in the domain of weather. The graph shows how qualities and quality-types are related through the `has-type` relationship, and how qualities and entities are related through the `inheres-in` relationship.

The development of background knowledge is often a difficult challenge. In many domains, the relationships between qualities and entities are unclear, resulting in representations that may be imprecise and/or incomplete. While we acknowledge the difficulty, the development of such domain specific knowledge is out of the scope of this paper. For examples of recent work in this area, see (Punuru, 2007; Suchanek, 2009; Thomas, Mehra, Brooks, & Sheth, 2008).

5.1.2. Implementation

Before describing the experiments, we describe our implementation of IntellegO. The implementation is written in Java and conforms to the specification formalized in Section 3. Figure 9 shows an architecture diagram.

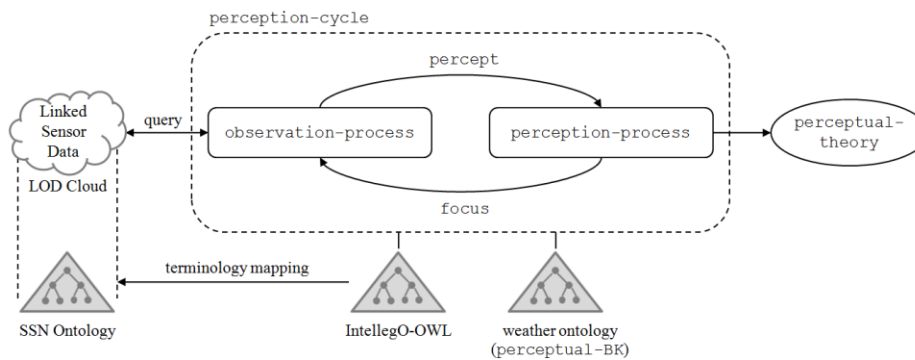


Figure 9. Architecture of an implementation of IntellegO.

The implementation of IntellegO respects the specification given in Section 3. However, two processes – `observation-process` and `choose` – were only partially defined; only the inputs and outputs were defined. The `choose` process is implemented differently for each experiment below, so the details will be described separately for each experiment. Note that the `focus-candidates` process returns a set of `quality-types` as an ordered sequence (the rationale for this decision is discussed in the experiment sections).

The implementation of the `observation-process` is highly dependent on the way in which sensor data is accessed. For example, the `observation-process` could be designed to task sensors in an environment and measure qualities in the world. For the evaluation presented in this paper, however, the sensor data has already been collected, encoded in RDF, and made accessible on the Web. Therefore, the `observation-process`, as implemented here, generates and executes a SPARQL (Prud’hommeaux & Seaborne, 2008) query against the sensor data on LOD. SPARQL (SPARQL Protocol and RDF Query Language) is a W3C recommended language for querying RDF data. A set of Java libraries for managing Semantic Web data, Jena/ARQ (Carroll et al., 2004), is used to build and execute the query.

The `observation-process` receives `focus` as input. In order to utilize this `focus` to generate an appropriate SPARQL query, in this implementation we also annotate `focus` with additional metadata, such as a time-interval and an observer (i.e., weather station). For example, given `focus` of `quality-type temperature`, time-interval `2003-04-01T02:00:00 to 2003-04-01T03:59:59`, and observer `System_SB1`, the `observation-process` will generate the following SPARQL query to detect the quality `freezing-temperature`.

```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
prefix lsd: <http://knoesis.wright.edu/ssw#>
prefix weather: <http://knoesis.wright.edu/ssw/ont/weather.owl#>
prefix time: <http://www.w3.org/2006/time#>

ASK {
  ?observation ssn:observedProperty ?qualitytype .
  ?qualitytype rdf:type weather:temperature .
  ?observation ssn:observedBy lsd:System_SB1 .
  ?observation ssn:observationSamplingTime ?time .
  ?time time:inXSDDateTime ?datetime .
  ?observation ssn:observationResult ?result .
  ?result ssn:hasValue ?value .
  ?result weather:uom weather:Fahrenheit .
  FILTER(?value <= 32.0)
  FILTER(?datetime >= "2003-04-01T02:00:00"^^xsd:datetime)
```

```
FILTER(?datetime <= "2003-04-01T03:59:59"^^xsd:datetime)
}
```

If the above ASK query returns *true* then the percept *freezing-temperature* is returned, otherwise the percept *not-freezing-temperature* is returned. The percept returned is an URI for the detected quality. SPARQL queries for the remaining quality-types are generated and executed in a similar manner. The above query is executed against the `LinkedSensorData` dataset described in Section 4.

5.1.3. Experiment Setup

Between April 1st and April 6th of 2003, a major blizzard hit the state of Nevada. Environmental data within the surrounding area was collected by weather-stations, encoded as RDF, and made accessible on the Web as Linked Data. For every two hour interval and for each observer within a 400 mile radius of the blizzard, we execute the `perception-cycle` and generate a `perceptual-theory`. For each execution of the `perception-cycle`, the observer is a weather-station and the resulting `perceptual-theory` contains member entities representing the weather event occurring at that time and location (of the weather station). After each execution, the resultant `perceptual-theory` is checked for correctness and the total number of percepts, in the set of `percepts`, is counted.

During the execution of the `perception-cycle`, two variables affect the size of the set of percepts: (1) the order in which quality-types are detected, and (2) the weather conditions surrounding the weather-station at the time of observation. The first issue is addressed by the way a quality-type is chosen to be observed by the *choose* process. In order to address the second issue, we evaluate a variety of weather conditions by executing the `perception-cycle` with observers at various distances from the blizzard. More specifically, we execute the `perception-cycle` with observers within a distance of 25 miles (17 observers), 50 miles (70 observers), 100 miles (170 observers), 200 miles (373 observers), and 400 miles (516 observers).

Given a particular time-interval and observer, in addition to the weather background knowledge described in Section 5.1.1, the minimum `perceptual-theory` generated by the `perception-cycle` should always be the same, despite the order in which quality-types are detected. The order only affects how efficiently the `perceptual-theory` is generated. Thus, precision and recall statistics do not make sense in this evaluation and

will not be shown. In these experiments the minimum perceptual-theory always contains either zero or one entity [note that this is a product of how the specific qualities and entities are related within the weather background knowledge and not a general rule]. As such, the resulting perceptual-theory will be labeled with a single term representing the single member entity that explains the set of percepts (e.g., *blizzard*). If the perceptual-theory contains no member entities (representing the empty set) then the perceptual-theory is labeled as *unclassified*.

To allow validation, repeatability, and further experimentation, we have stored the data generated by these experiments as RDF, accessible at: <http://wiki.knoesis.org/index.php/Intellego>.

5.1.4. Experiment 1: No Focus (brute force approach)

An experiment to test the naïve brute force approach was conducted by executing the perception-cycle as described in Section 5.1.3, with one major retraction: the ability to check for discriminating qualities and generate focus was disabled. The set of quality-types to choose from is given as an ordered sequence: *{temperature, precipitation, wind-speed}*. The choose process simply returns the first quality-type in this ordered sequence. Since, in this scenario, all quality-types must be detected, the order of detection is irrelevant and the number of percepts generated for each execution of the perception-cycle remains constant. For this experiment, we executed the perception-cycle 37,152 times; once for each combination of time interval (72) and observer (516). Each execution generated a set of percepts. Table 5 shows the total number of percepts generated for each set of observers.

Table 5. The total number of percepts generated during all the executions of the perception-cycle during experiment 1 (for different sets of observers).

25 miles (17 observers)	50 miles (70 observers)	100 miles (170 observers)	200 miles (373 observers)	400 miles (516 observers)
3672	15,120	36,720	80,568	111,456

Table 6 shows the types of perceptual-theories that resulted from the execution of the perception-cycle for the different sets of observers. From this table, we can see that the *clear* condition is by far the most common while *blizzard* and *unclassified* rarely occur. There is a minor trend towards a decreasing percentage of

blizzard, *flurry*, and *unclassified* theories, and an increasing percentage of *clear* theories, as the distance from the blizzard increases. This trend seems reasonable, since as you move farther away from the blizzard, the weather is more likely to be clear.

Table 6. Shows the percentage of different perceptual-theories generated during the execution of the perception-cycle (for different sets of observers).

distance (# observers)	<i>blizzard</i>	<i>flurry</i>	<i>rain-storm</i>	<i>rain-shower</i>	<i>clear</i>	<i>unclassified</i>
25 miles (17)	1.77%	22.21%	0%	17.65%	58.36%	3.5%
50 miles (70)	0.5%	17.19%	0.44%	14.04%	67.8%	1.7%
100 miles (170)	0.25%	14.08%	0.95%	14.83%	69.86%	1.2%
200 miles (373)	0.11%	9.64%	2.25%	20.5%	67.48%	0.9%
400 miles (516)	0.09%	9.11%	2.41%	21.58%	66.81%	0.85%

5.1.5. Experiment 2: With Focus

The choose process for this experiment is the same as in the first experiment. This time, however, the ability to generate focus is enabled and thus the focus-candidates process returns an ordered sequence of only those quality-types that may discriminate between entities in the perceptual-theory. Under these conditions, the ordering in the ordered sequence of quality-types affects the number of percepts generated.

Consider the following illustrative examples. Suppose we have a perceptual-theory $\{\textit{blizzard}, \textit{flurry}, \textit{rain-shower}\}$. First, consider the ordered sequence of quality-types $\{\textit{precipitation}, \textit{wind-speed}\}$. The choose process will pick *precipitation* as focus. Now suppose *snow-precipitation* is detected, resulting in an updated perceptual-theory $\{\textit{blizzard}, \textit{flurry}\}$. This perceptual-theory is not minimum, so the perception-cycle will continue. The next ordered sequence of quality-types is now $\{\textit{wind-speed}\}$, so the choose process will pick *wind-speed* as focus. If we suppose that *high-wind-speed* is detected, the resulting perceptual-theory is $\{\textit{blizzard}\}$. This is a minimum perceptual-theory, so the perception-cycle terminates. In total, two percepts were generated, *snow-precipitation* and *high-wind-speed*. Now consider the case where the ordered sequence of quality-types has been changed to $\{\textit{wind-speed}, \textit{precipitation}\}$. The choose process will pick *wind-speed* as focus. Again suppose *high-wind-speed* is detected,

resulting in an updated perceptual-theory {*blizzard*}. This perceptual-theory is minimum, so the perception-cycle terminates. In this case, only one percept was generated, *high-wind-speed*.

Since we do not know a-priori which ordered sequence of quality-types will produce the optimal result, we test each possible sequential order. Given the three quality-types in the background knowledge – *temperature* (t), *precipitation* (p), and *wind-speed* (w) – there are six possible orderings. Thus, for this experiment, we executed the perception-cycle 222,912 times; once for each time interval (72), observer (516), and permutation of quality-types (6). As expected, the perceptual-theories generated during the execution of the perception-cycle, as shown in Table 6, were found to be identical in this experiment. Figure 10 shows the results of executing the perception-cycle for the 516 weather-stations within a radius of 400 miles of the blizzard, for each time interval, and for each ordering of quality-types.

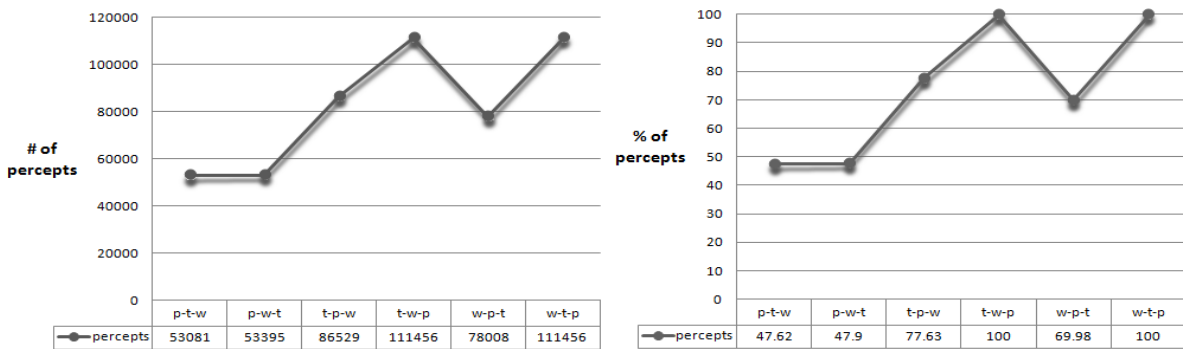


Figure 10. Percepts generated by observers within 400 miles of a known blizzard. The horizontal axis represents the different orderings of observable qualities; *p* represents *precipitation*, *w* represents *wind-speed*, and *t* represents *temperature*.

For each ordering of quality-types, 37,152 perceptual-theories were generated from 111,456 potential quality-type detections. However, in many cases the number of percepts is far less than the number of quality-types that could potentially be detected. For example, with two of the orderings (*p-t-w* and *p-w-t*) the ratio is less than 48%; this accounts for a significant reduction in the number of percepts needed to generate the minimum perceptual-theory. On the other hand, with two of the orderings (*w-t-p* and *t-w-p*) the ratio is 100%; all of the quality-types were detected resulting in the maximum number of percepts. Looking at Figure 10, we see that the two orderings with the best results (*p-w-t* and *p-t-w*) both begin with *precipitation*; and the two orderings with the worst results (*w-t-p* and *t-w-p*) both place *precipitation* last in the

order. This may inform us that a *precipitation* percept is proficient in discriminating between entities in the perceptual-theory. From the weather background knowledge in Figure 8, this can be more clearly seen by noticing that the *no-precipitation* quality only inheres-in the entity *clear*. This means that a *no-precipitation* percept is only explained by the entity *clear*; if this quality is detected then the minimum perceptual-theory is found. Therefore, detecting the *precipitation* quality-type early is an efficient approach. This experiment clearly shows that the order in which quality-types are evaluated and detected dramatically affects the efficiency of the perception-cycle.

The statistics of the remaining executions of the perceptual-cycle – for each set of observers within a radius of 25, 50, 100, 200, and 400 miles – are shown in Table 7. You may notice that the percentage of percepts remains fairly stable across the different sets of observers. There is a minor trend towards a decreasing percentage of percepts as the distance from the blizzard increases. This trend can be explained by noticing that complex weather conditions (e.g., blizzard) may require more percepts to explain than more simple weather conditions (e.g., clear).

Table 7. Shows the percentage of percepts generated during the execution of the perception-cycle (for different sets of observers and orderings of quality-types).

distance (# observers)	p-t-w	p-w-t	t-p-w	t-w-p	w-p-t	w-t-p
25 miles (17)	55.58%	56.67%	80.09%	100%	75.49%	100%
50 miles (70)	50.25%	50.82%	77.21%	100%	73.03%	100%
100 miles (170)	48.38%	48.80%	76.58%	100%	71.80%	100%
200 miles (373)	47.60%	47.92%	77.40%	100%	70.20%	100%
400 miles (516)	47.62%	47.90%	77.63%	100%	69.98%	100%

5.1.6. Experiment 3: With Optimized Focus

The previous experiment showed that while *focus* is useful for efficient perception, these gains in efficiency are dependent on knowing the optimal sequential ordering of quality-types to *focus* attention. Since the number of possible sequential orderings grows exponentially with the total number of quality-types ($O(n!)$, where $n = \#$ of quality-types), such an approach to arriving at the optimal order may not be suitable for Web-scale data. One possible solution to the scalability issue would be to learn the optimal ordering of quality-types by analyzing a *representative* training dataset. For this approach, we have leveraged a decision-tree algorithm (C4.5

(Mitchell, 1997)). Starting with a representative training dataset that has been annotated with correct classifications, a decision-tree representation capable of efficient classification is generated. This algorithm is able to compute the optimal ordering of quality-types in polynomial-time with respect to the total number of quality-types ($O(n^3 * \max(v)^n$, where $n = \#$ of quality-types and $v = \#$ of discrete qualities for a quality-type). The polynomial-time complexity of this algorithm is a drastic improvement over the exponential-time complexity of the technique employed in the second experiment (Section 5.1.5).

In the third experiment, we executed the C4.5 decision-tree algorithm over a training dataset which includes a representative sample of quality detections within 400 miles of the blizzard in Nevada. An excerpt of the training dataset is shown in Table 8, and the resulting decision-tree is shown in Figure 11. The choose process for this experiment is more complex than in the first or second experiment. Instead of simply returning the first quality-type in the sequence, the choose process returns the quality-type represented by the current node in the decision-tree, which has the highest informational value (i.e., it will discriminate between entities the most)¹⁰. The quality-type represented by this node, therefore, is the optimal choice for focusing attention. As noted in Section 2, Peter Norwich first used Information Theory to quantify the informational value of observations; he called this the Entropy Theory of Perception (Norwich, 1991). This experiment goes (slightly) further and orders the observable quality-types based on their informational value and encodes this order in a decision tree data-structure.

Table 8. Excerpt from the representative training dataset.

wind speed	temperature	precipitation	classification
high	freezing	snow	blizzard
low	freezing	snow	flurry
low	freezing	snow	flurry
high	not-freezing	rain	rain-storm
low	not-freezing	rain	rain-shower
low	not-freezing	none	clear
low	not-freezing	none	clear
high	not-freezing	none	clear

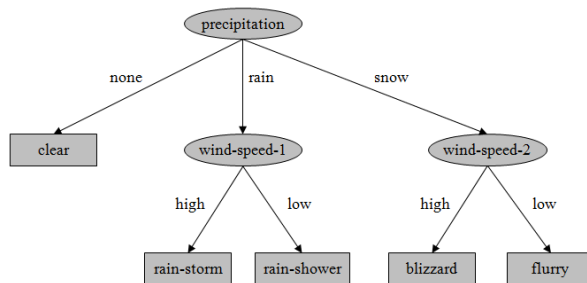


Figure 11. Decision tree representing the optimal sequential ordering of quality-types.

¹⁰ The C4.5 decision-tree algorithm is based on Claude Shannon's Information Theory (Shannon, 1948), and the current node in the tree corresponds to the attribute from the training dataset with the highest *information-gain* (Mitchell, 1997).

The optimal ordering of quality-types, shown in Figure 11, begins with *precipitation* – at the root of the tree – followed by *wind-speed*. This is consistent with the previous optimal orderings found in Section 5.1.5 (i.e., *p-w-t*). Notice, however, that *temperature* is not represented within the decision-tree. This omission is the result of a discovery by the decision-tree algorithm that *temperature* is always extraneous.

For this experiment, we executed the *perception-cycle* 37,152 times; once for each combination of time interval (72) and observer (516). Table 9 shows the number and percentage of *percepts* generated for each set of observers. The results show an average 50% reduction in the number of *percepts* needed to generate a minimum *perceptual-theory*; these results mirror those found in Section 5.1.5 for the sequential ordering of quality-types $\{precipitation, wind-speed, temperature\}$.

Table 9. Shows the number and percentage of *percepts* generated during the execution of the *perception-cycle* (for different sets of observers and orderings of quality-types).

distance (# observers)	# of percepts	% of percepts
25 miles (17)	2081	56.67%
50 miles (70)	7684	50.82%
100 miles (170)	17921	48.80%
200 miles (373)	38613	47.92%
400 miles (516)	53395	47.90%

5.2. Expressivity Evaluation

In our design and representation of perception, we emphasize two important capabilities: (1) the ability to degrade gracefully with incomplete information, and (2) the ability to minimize explanations based on new information. Current solutions to the perception problem often encode the background knowledge within first-order logic (FOL). In particular, Riquebourg (Riquebourg et al., 2007), Keßler (Keßler, Raubal, & Wosniok, 2009), and Sheth (Sheth, Henson, & Sahoo, 2008) and have all used the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) to encode such background knowledge and infer explanations. SWRL is a restricted fragment of FOL (see (Horrocks et al., 2004) for additional details). Calder (Calder, Morris, & Peri, 2010) and Henson (Henson, Pschorr, Sheth, & Thirunarayan, 2009) have also used the Jena Rule Engine for this task. In the following section, we compare IntellegO with SWRL and first-order logic; and summarize our results in Table 10. While IntellegO is expressive enough to achieve both capabilities, SWRL achieves neither, and FOL can degrade gracefully with incomplete information, but cannot minimize explanations. The ability to minimize explanations is an anti-monotonic process;

and therefore, it is not surprising that the monotonic FOL and the more restrictive SWRL cannot express such a process.

Table 10. Qualitative comparison of logic frameworks to express the desired capabilities of perception; including the ability to degrade gracefully with incomplete information and the ability to minimize explanations with additional information.

	Graceful Degradation	Minimize Explanations
IntellegO	Yes	Yes
SWRL	-	-
FOL	Yes	-

5.2.1. Example Background Knowledge

To compare the expressivity of existing approaches, we will provide an example scenario based on the background knowledge represented in Figure 12 and show how each approach behaves.

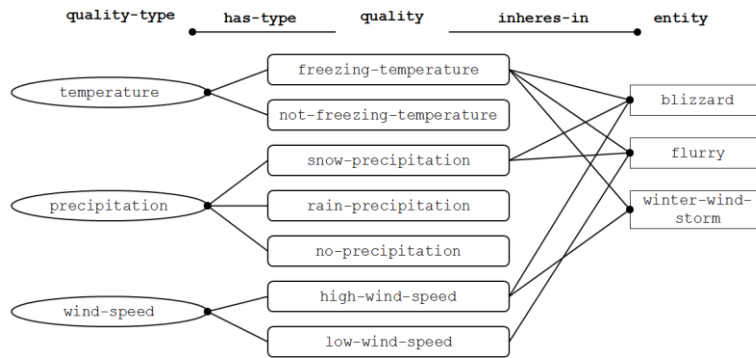


Figure 12. Example background knowledge used within the expressivity evaluation of IntellegO.

5.2.1.1. Encoding of Rules in SWRL

A straightforward encoding of Figure 12 in SWRL, based on the SSN ontology, is shown below. In this encoding, each `entity` is defined as a rule, as demonstrated by (Keßler et al., 2009; Riquebourg et al., 2007; Sheth, Henson, & Sahoo, 2008). For each rule, we assume that a generic `entity` individual has been created and added to the knowledge base for each spatial-temporal context. For simplicity, all spatial-temporal and value constraints are removed, since they complicate the rules without differentiating the approach (note that in a real-world application, such constraints must be added).

Blizzard Rule

```
ssn:isQualityOf(high-wind-speed, ?e)  $\wedge$   
ssn:isQualityOf(freezing-temperature, ?e)  $\wedge$   
ssn:isQualityOf(snow-precipitation, ?e)  
 $\rightarrow$  blizzard(?e)
```

Flurry Rule

```
ssn:isQualityOf(low-wind-speed, ?e)  $\wedge$   
ssn:isQualityOf(freezing-temperature, ?e)  $\wedge$   
ssn:isQualityOf(snow-precipitation, ?e)  
 $\rightarrow$  flurry(?e)
```

Winter Wind Storm Rule

```
ssn:isQualityOf(high-wind-speed, ?e)  $\wedge$   
ssn:isQualityOf(freezing-temperature, ?e)  $\wedge$   
 $\rightarrow$  winter-wind-storm(?e)
```

In order to satisfy the entity rules defined above, each of the *ssn:isQualityOf* predicates in the antecedent must be satisfied. This is achieved if a corresponding observation is found in the knowledge base, as defined in the following rule:

Observation Rule

```
ssn:observedProperty(?o, ?q)  $\wedge$  ssn:featureOfInterest(?o, ?e)  
 $\rightarrow$  ssn:isQualityOf(?q, ?e)
```

5.2.1.2. Additional Rules in FOL

First-order logic (FOL), in general, is more expressive than SWRL. In particular, FOL provides the disjunction (\vee) and negation (\neg) operator – which are not permissible in SWRL [Mei-2004] – that may be utilized to infer more complex explanations. In addition to the SWRL rules defined above, the following FOL rules can be added in order to guarantee that a set of observed qualities (i.e., percepts) are mutually exclusive and collectively exhaustive for each quality-type. These rules also require the introduction of the *has-type* relation from IntellegO, which explicitly relates qualities to quality-types. The *mutually exclusive criterion* says that at most one quality, per quality-type, may be detected for each entity.

Mutually Exclusive Rule

```
ssn:isQualityOf(?q, ?e)  $\wedge$  io:has-type(?q, ?t)
```

$$\rightarrow [\forall ?q' \in Q : (?q' \neq ?q) \wedge \text{io:has-type} (?q', ?t) \rightarrow \neg \text{ssn:isQualityOf} (?q', ?e)]$$

The *collectively exhaustive criterion* says that at least one quality, per quality-type, must be detected for each entity. This criterion may not be generally true and is not defined in IntellegO; but is necessary to infer suitable explanations with FOL.

Collectively Exhaustive Rule

$$\forall ?e \in E, \forall ?t \in QT : [\exists ?q \in Q : \text{ssn:isQualityOf} (?q, ?e) \wedge \text{io:has-type} (?q, ?t)]$$

With the addition of these two rules, and given a set of observations, FOL is able to generate an explanation as a disjunction of entities. As seen in the comparisons below, this enables satisfactory results in relation to the ability to degrade gracefully with incomplete information.

5.2.2. Expressivity Comparison

Given the background knowledge described above, the following scenarios will provide additional facts to the knowledge base. Each will then show the inference results from IntellegO, a SWRL inference engine, and a FOL inference engine.

5.2.2.1. Comparison 1: The ability to degrade gracefully with incomplete information

Given observations for *freezing-temperature* and *snow-precipitation*, in addition to the background knowledge above, the expected perceptual-theory would be $\{blizzard, flurry\}$. That is, *blizzard* is an explanation, and *flurry* is an explanation. Notice that *freezing-temperature* and *snow-precipitation* do not completely describe either *blizzard* or *flurry*, but they both provide evidence for *blizzard*, and they both provide evidence for *flurry*.

Facts in knowledge base

`ssn:Entity(e)`
`ssn:Observation(o1)`
`ssn:observedProperty(o1, snow-precipitation)`
`ssn:featureOfInterest(o1, e)`
`ssn:Observation(o2)`

```

ssn:observedProperty(o2, freezing-temperature)
ssn:featureOfInterest(o2, e)

```

Given the above facts in the knowledge base resulting from observations, in addition to the rules in Section 5.2.1.1 and Section 5.2.1.2, the following results:

Resulting explanations (i.e., perceptual-theory)

```

IntellegO:  {blizzard, flurry}
SWRL:
FOL:       blizzard(e)  $\vee$  flurry(e)

```

From this example, IntellegO provides *blizzard* as an explanation, and *flurry* as an explanation, for the set of observations. SWRL, on the other hand, does not provide any explanation. FOL provides *blizzard* \vee *flurry* as the explanation. This result occurs by virtue of the fact that *high-wind-speed* and *low-wind-speed* are the only two qualities of quality-type *wind-speed*. This knowledge thus allows FOL to derive meaningful partial information. We can see that both IntellegO and FOL allow explanations to degrade gracefully with incomplete information, while SWRL does not.

5.2.2.2. Comparison 2: The ability to minimize explanations based on new information

In the initial situation, we are given observations for *high-wind-speed* and *freezing-temperature*.

Initial situation

Facts in knowledge base

```

ssn:Entity(e)
ssn:Observation(o1)
ssn:observedProperty(o1, high-wind-speed)
ssn:featureOfInterest(o1, e)
ssn:Observation(o2)
ssn:observedProperty(o2, freezing-temperature)
ssn:featureOfInterest(o2, e)

```

Given the above facts in the knowledge base resulting from observations, in addition to the rules in Section 5.2.1.1 and Section 5.2.1.2, the following results:

Resulting explanations (i.e., perceptual-theory)	
IntellegO:	{winter-wind-storm, blizzard}
SWRL:	winter-wind-storm(e)
FOL:	winter-wind-storm(e)

Now suppose that the *precipitation* quality-type is detected and an additional *snow-precipitation* observation is added to the knowledge base. Given observations for *high-wind-speed*, *freezing-temperature*, and *snow-precipitation*, in addition to the background knowledge above, the expected perceptual-theory would be {*blizzard*}.

Updated situation with new observation

Facts in knowledge base	
<i>ssn:Entity</i> (e)	
<i>ssn:Observation</i> (o1)	
<i>ssn:observedProperty</i> (o1, <i>high-wind-speed</i>)	
<i>ssn:featureOfInterest</i> (o1, e)	
<i>ssn:Observation</i> (o2)	
<i>ssn:observedProperty</i> (o2, <i>freezing-temperature</i>)	
<i>ssn:featureOfInterest</i> (o2, e)	
<i>ssn:Observation</i> (o3)	
<i>ssn:observedProperty</i> (o3, <i>snow-precipitation</i>)	
<i>ssn:featureOfInterest</i> (o3, e)	

Given the above facts in the knowledge base resulting from observations, in addition to the rules in Section 5.2.1.1 and Section 5.2.1.2, the following results:

Resulting explanations (i.e., perceptual-theory)	
IntellegO:	{blizzard}
SWRL:	winter-wind-storm(e), blizzard(e)
FOL:	winter-wind-storm(e) \wedge blizzard(e)

From this example, IntellegO provides *blizzard* as the only explanation for the set of observations, while SWRL and FOL provide *winter-wind-storm* and *blizzard* as explanations. With IntellegO, as more information (i.e., observations) is provided, the set of explanations is reduced to improve specificity. With SWRL and FOL, on the other hand, as more information is provided, the set of explanations can grow only larger. We can see that IntellegO has the ability to minimize explanations based on new information in an intuitively satisfactory way, while both SWRL and FOL are unable to do so due to their monotonicity.

5.3. Storage Requirements and Scalability Evaluation

In the digital age, information is being generated at an extraordinary pace. It has even been suggested that *more data has been created in the last three years than in the past 40,000*. Around 2008, this rate of expansion surpassed the generation rate of storage capacity, leading to a future where we can no longer store all the sensor data being generated (Higginbotham, 2010). With this future ahead, the efficient representation and interpretation of sensor data at scale has become an important area of research.

A single flight from New York to Los Angeles on a twin-engine Boeing 737 generates around 240 terabytes of data; and, on any given day, there are around 28,537 such commercial flights in the United States (Higginbotham, 2010). But how much of this sensor data is actually useful for decision-making and needs to be stored for later retrieval? The pilot may need to understand the general condition of the plane and the external environment during flight; the ground crew may need to know the speed, location and trajectory of the aircraft; and the mechanic may need to know of any anomalous behavior detected during the flight. Much of this knowledge must be derived from the low-level observational data, but only the high-level inferences may be required for actual decision-making.

As another example, consider a weather alert service which represents, stores, and alerts people of the type of current weather conditions. In this case, all the low-level observations can be discarded while only the *perceptual-theories* need to be stored. If the weather alert service only generates alerts for severe weather conditions, then *perceptual-theories* representing “irrelevant” weather conditions, such as *clear*, can also be discarded; such information is unnecessary for the task of alerting people of severe weather.

We evaluate the resources required for storing sensor observations and *perceptual-theories*, and show that for some applications the generation of *perceptual-theories* can lead to significant storage savings. For this evaluation, we will count the number of records that are generated and stored for different dataset storage configurations. A record could represent an observable quality, percept, or *perceptual-theory*. The statistics used in the following five data storage configurations are summarized in Figure 13.

1. All observable *qualities* are stored,
2. All *perceptual-theories* are stored,

3. All observable qualities and all perceptual-theories are stored,
4. Only *relevant* perceptual-theories are stored, and
5. Only *relevant* perceptual-theories and *relevant* percepts are stored.

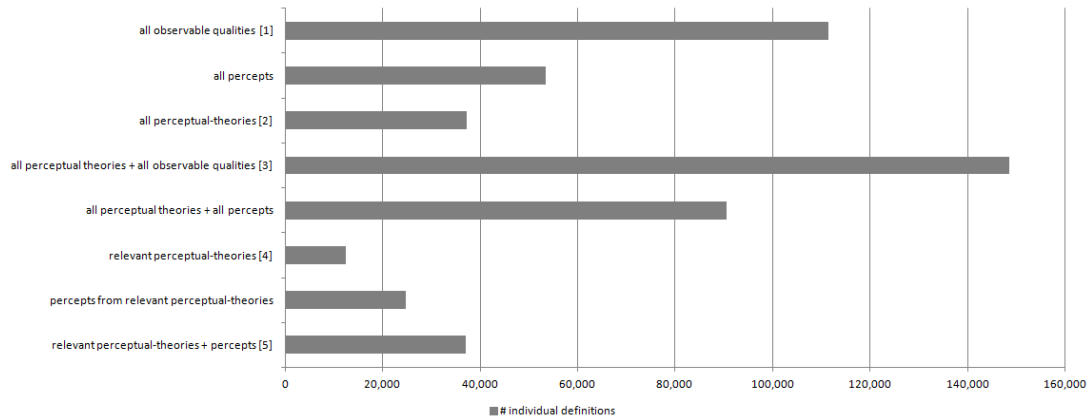


Figure 13. Storage requirements for several common dataset configurations; based on statistics gathered from the experiment discussed in Section 5.1.

1. **All observable qualities are stored** – Within this dataset configuration, `percepts` for all possible observable qualities are generated and stored. From the experiment in Section 5.1, this configuration generates 111,456 records.
2. **All perceptual-theories are stored** – This dataset configuration generates and stores `perceptual-theories` for all weather conditions (within some spatial-temporal context). From the experiment in Section 5.1, this configuration generates 37,152 records. The ratio of records generated for (1) **all observable qualities** versus (2) **all perceptual-theories** is around 3:1. Thus, if only `perceptual-theories` are required for an application then the storage requirements result in 66% savings.
3. **All observable qualities and all perceptual-theories are stored** – This configuration represents the situation from our experiment in Section 5.1.4 that executed a brute force approach to observe all qualities and generate all `percepts` and `perceptual-theories`. During this experiment, 111,456 `percept` records and 37,152 `perceptual-theory` records are generated and stored, totaling 148,608 records.
4. **Only *relevant* perceptual-theories are stored** – Within this configuration, `perceptual-theories` for only *relevant* (*severe*) weather conditions (within some spatial-temporal context) are defined and stored.

From the experiment in Section 5.1, we can define a *relevant* perceptual-theory as any perceptual-theory representing a *blizzard*, *flurry*, *rain-storm*, or *rain-shower* condition. 12,331 records are generated with this configuration. The ratio of records generated for **(1) all observable qualities versus (4) only relevant perceptual-theories** is around 9:1. The ratio of records generated for **(3) all observable qualities and all perceptual-theories versus (4) only relevant perceptual-theories** is around 12:1. These represent over an order of magnitude storage savings.

- 5. Only relevant perceptual-theories and relevant percepts are stored** – It may also be important to store the percepts associated with the *relevant* perceptual-theories. Such information may be useful for validation, additional analysis, or simply allowing for further investigation. For example, suppose the *severe* weather alert service described above also allowed the user to access the observation records that were used to determine the *severe* weather condition (e.g., to see just how fast the wind is blowing). From the experiment in Section 5.1, this configuration generates 36,993 records. The ratio of records generated for **(3) all observable qualities and all perceptual-theories versus (5) only relevant perceptual-theories and relevant percepts** is around 4:1.

This evaluation illustrates the benefits that come from the ability to abstract away from the details of low-level sensory input and generate high-level explanations. While the application scenarios may vary, and the definition of *relevance* is highly dependent on the domain of interest and application, such examples clearly show the benefits, from generating perceptual-theories with IntellegO.

6. Related Work

In addition to the SSN ontology, there have been several other attempts to develop an ontology for sensors and sensor observations (Compton, Henson, Lefort, Neuhaus, & Sheth, 2009). Kuhn (Kuhn, 2009) and Stasch (Stasch, Janowicz, Bröring, Reis, & Kuhn, 2009) have developed an ontology for describing sensors and sensor observations in Haskell. Similar to IntellegO, this ontology attempts to represent the process of observation in a way that is independent of any particular implementation technology (e.g., machine sensor, human eye). The Perception,

Cognition and Communication (PCC) Ontology (Anandavala, 2010) defines a set of classes and relations that provide terminology to describe "what we believe exists, what we experience, what we think and what we communicate." PCC shares many concepts with the ontology of perception described in this paper. However, we have been unable to find any use-case, application, or evaluation for this ontology. Even the Gene Ontology (GO) contains representation for sensory perception ("Sensory Perception," 2010). However, perception in GO is defined as a neurophysiological process and is similar to what we describe as the *observation-process*. Devaraju et al. (Devaraju & Kuhn, 2010) have developed an approach for representing the relationship between observed qualities and the geo-processes that influence those observations. This approach is aligned with the DOLCE foundational ontology (Borgo & Masolo, 2009) and has been used to represent relations between qualities and entities in the domain of hydrology. This ontology has been used primarily for the integration of sensor data, and there is no attempt to show how to perform inference over observed qualities. Scheider et al. (Scheider, Probst, & Janowicz, 2010) have developed a general theory for grounding entities and qualities to observation processes, based on ideas from language semantics.

In addition to the development of related ontologies, there have been several efforts to reason over observational data, encoded as RDF, in order to infer entities in the world. In 2008, Sheth (Sheth, Henson, & Sahoo, 2008) suggested using the Semantic Web Rule Language (SWRL) to reason over sensor data. Since this time, Keßler (Keßler et al., 2009) has further developed this idea and provides a suitable investigation of the use of SWRL for reasoning over sensor data. Riquebourg (Riquebourg et al., 2007) has utilized SWRL to infer context and determine proper actions (i.e., turn on/off lights) within a smart home environment which contains sensors. Calder (Calder et al., 2010) has used the Jena Semantic Web Toolkit (Carroll et al., 2004) to define rules to detect anomalous events (such as *winter weather* and *algal blooms*). These approaches all define first-order logic (deductive) rules to infer entities from qualities. As discussed in this paper, however, such an approach is limited in its ability to represent the anti-monotonic nature of perception, to handle incomplete information, and to minimize explanations based on new information. In particular, such approaches cannot model perception as a cyclical process that actively seeks out and detects those qualities which carry information most useful for explanation.

Reggia and Peng (Reggia & Peng, 1986) have discussed techniques for abductive reasoning (called Parsimonious Covering Theory) that is similar to the approach taken in this paper; however, they were mainly interested in inferring medical diseases from symptoms. Perhaps the closest related work is by Shanahan (Shanahan,

2005), who formalized an abductive account of perception using Event Calculus. More specifically, he characterized perception as follows: “Given a stream of low-level sensor data, represented by the conjunction Γ of a set of observation sentences, the task of perception is to find one or more explanations of Γ in the form of a consistent logical description Δ of hypothesized objects, such that, $\Sigma \cap \Delta \Rightarrow \Gamma$, where Σ is a background theory describing the environment” (Shanahan, 2005). Thirunarayan (Thirunarayan, Henson & Sheth, 2009) explored how a logic programming-based abductive reasoning framework can benefit the formalization and interpretation of sensor data to garner situation awareness. Although there are several related efforts, the research discussed in this paper is first of its kind to present and evaluate an approach that is grounded in well-established cognitive theories of perception and also applicable for sensor applications on the Web.

7. Conclusion

Sensors are quickly becoming ubiquitous and are now collecting data about our environment at an extraordinary pace. In this paper, we have demonstrated substantial benefits to be gained in processing sensor data by automating an approximation of how people perceive their environment efficiently. Specifically, this ability is afforded by background knowledge and the cyclical nature of observation and perception processes. While one can take different positions on the philosophical (or technical) foundations of perception, it is clear that a careful ontological specification makes these positions explicit and testable. The ontology described in this paper establishes a formal semantics for machine perception; and provides a solution to difficult challenges, such as the ability to effectively model the process of perception, to provide an appropriate interpretation of observational data with incomplete information, and to efficiently interpret and store the growing stream of observational data. This approach has been evaluated on a large, open, real-world dataset of weather observations. Three evaluations were provided to demonstrate (1) how focus can lead to improved efficiency in generating and processing sensor observations, (2) how the expressivity of IntellegO compares to existing solutions, and (3) how perception can lead to significant storage savings. In the future, we hope that such results can be exploited to enable significant savings of energy and computational resources. If current trends in sensor capabilities continue, the predicted data requirements will be in excess of a yottabyte (10^{24} Bytes) by 2015 (Mitre, 2008). Besides the obvious issues of data management, knowing

how to effectively make sense of sensor data – that will largely be flowing through the Web – represents a significant challenge. This research is an early and modest step in understanding and addressing this challenge.

Acknowledgements. This research was supported in part by The Dayton Area Graduate Studies Institute (DAGSI), AFRL/DAGSI Research Topic SN08-8: "Architectures for Secure Semantic Sensor Networks for Multi-Layered Sensing." We would also like to thank Pramod Anantharam for his help with implementation and evaluation.

References

- Aberer, K., Hauswirth, M., & Salehi, A. (2006). A middleware for fast and flexible sensor network deployment. In D. Umeshwar, K.-Y. Whang, D. Lomet, G. Alonso, G. Lohman, M. Kersten, S. K. Cha, et al. (Eds.), *Proceedings of the 32nd International Conference on Very Large Data Bases* (pp. 1199–1202). VLDB Endowment.
- Aloimonos, J., Weiss, I., & Bandyopadhyay, A. (1988). Active Vision. (A. Blake & A. Yuille, Eds.) *International Journal of Computer Vision*, 1(4), 333-356. MIT Press. doi:10.1007/BF00133571
- Anandavala. (2010). Perception, Cognition, and Communication (PCC) Ontology. *Anandavala*. Retrieved from <http://www.anandavala.info/ontech/PCC/pcc-intro.html>
- Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. (F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, & P. F. Patel-Schneider, Eds.) *Description Logic Handbook* (p. 622). Cambridge University Press. doi:10.2277/0521781760
- Bajcsy, R. (1988). Active perception. (Y. Aloimonos, Ed.) *Proceedings of the IEEE*, 76(8), 996-1005. IEEE. doi:10.1109/5.5968
- Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3), 1-22. Elsevier. doi:10.4018/jswis.2009081901
- Borgo, S., & Masolo, C. (2009). Foundational choices in DOLCE. *Applied Ontology*, 2, 361-381. Springer-Verlag.
- Botts, M., Percivall, G., Reed, C., & Davidson, J. (2008). OGC (R) Sensor Web Enablement: Overview and High Level Architecture. *Lecture Notes in Computer Science* (Vol. 4540, pp. 175–190). Springer. doi:10.1007/978-3-540-79996-2
- Calder, M., Morris, R. A., & Peri, F. (2010). Machine reasoning about anomalous sensor data. *Ecological Informatics*, 5(1), 9-18. Elsevier B.V. doi:10.1016/j.ecoinf.2009.08.007
- Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., & Wilkinson, K. (2004). Jena: implementing the semantic web recommendations. *Digital Media*, (HPL-2003-146), 74-83. ACM. doi:10.1145/1013367.1013381
- Cavanagh, P. (1999). Top-Down Processing in Vision. *The MIT Encyclopedia of the Cognitive Sciences*, (Costall 1980), 839-840.
- Compton, M., Henson, C., Lefort, L., Neuhaus, H., & Sheth, A. (2009). A Survey of the Semantic Specification of Sensors. In K. Taylor, A. Ayyagari, & D. D. Roure (Eds.), *International Workshop on Semantic Sensor Networks* (Vol. 522, p. 17). CEUR-WS.org.

- Corcho, O., & García-Castro, R. (2010). Five challenges for the Semantic Sensor Web. *Semantic Web - Interoperability, Usability, Applicability*, 1(1), 121-125. IOS Press. doi:10.3233/SW-2010-0005
- Devaraju, A., & Kuhn, W. (2010). A Process-Centric Ontological Approach for Integrating Geo-Sensor Data. *Proceeding of the International Conference on Formal Ontology in Information Systems* (pp. 199-212). IOS Press.
- Diamant, E. (2007). Modeling human-like intelligent image processing: An information processing perspective and approach. *Signal Processing Image Communication*, 22(6), 583-590. doi:10.1016/j.image.2007.05.007
- Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. *booksgooglecom* (pp. 1 - 333). Springer-Verlag, Berlin Heidelberg. doi:10.1007/978-3-540-49612-0
- Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. (L. Carmichael, Ed.) *Leonardo* (Vol. 1, p. 335). Houghton Mifflin. doi:10.2307/1571911
- Goldstone, H. H. (1964). Computers and Perception. *Proceedings of the American Philosophical Society*, 108(4), pp. 282-290. American Philosophical Society.
- Gregory, R. L. (1968). Perceptual illusions and brain models. *Proceedings of the Royal Society of London*, 171(24), 279-296. JSTOR.
- Gregory, R. L. (1997). Knowledge in perception and illusion. *Philosophical Transactions of the Royal Society of London*, 352(1358), 1121-1127. The Royal Society.
- Gries, D. (1999). Monotonicity in Computational Proofs. *Correct System Design, Recent Insight and Advances* (pp. 79-85). London, UK: Springer-Verlag.
- Gürgen, L., Roncancio, C., Labbé, C., & Olive, V. (2006). Transactional issues in sensor data management. *Proceedings of the 3rd Workshop on Data Management for Sensor Networks in conjunction with VLDB 2006*, 27. ACM Press. doi:10.1145/1315903.1315910
- Henson, C., Pschorr, J. K., Sheth, A. P., & Thirunarayan, K. (2009). SemSOS: Semantic sensor Observation Service. *2009 International Symposium on Collaborative Technologies and Systems*, 0(Cts), 44-53. Ieee. doi:10.1109/CTS.2009.5067461
- Higginbotham, S. (2010, September). Sensor Networks Top Social Networks for Big Data. *Gigaom.com*. Retrieved from <http://gigaom.com/cloud/sensor-networks-top-social-networks-for-big-data-2/>
- Hitzler, P., Parsia, B., Patel-Schneider, P. F., & Rudolph, S. (2009). *OWL 2 Web Ontology Language Primer*. W3C Recommendation.
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., & Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *Syntax*. W3C Recommendation.
- Janowicz, K., Schade, S., Bröring, Arne, Keßler, C., Maué, P., & Stasch, C. (2010). Semantic Enablement for Spatial Data Infrastructures. (J. P. Wilson, A. S. Fotheringham, & D. O'Sullivan, Eds.) *Transactions in GIS*, 14(2), 111-129. Wiley Online Library. doi:10.1111/j.1467-9671.2010.01186.x
- Keßler, C., Raubal, M., & Wosniok, C. (2009). Semantic Rules for Context-Aware Geographical Information Retrieval. (P. Barnaghi, K. Moessner, M. Presser, & S. Meissner, Eds.) *4th European Conference on Smart Sensing and Context EuroSSC 2009*, 5741, 77-92. Springer.
- Kuhn, W. (2009). A Functional Ontology of Observation and Measurement. In J. Janowicz, M Raubal, & S. Levashkin (Eds.), *International Conference on Geospatial Semantics* (Vol. 3, pp. 26-43). Springer.

- Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., et al. (2011). *Semantic Sensor Network XG Final Report, W3C Incubator Group Report*. Retrieved from <http://www.w3.org/2005/Incubator/ssn/XGR-ssn/>
- Locke, J. (1690). *An Essay Concerning Human Understanding*. (L. A. Selby-Bigge, Ed.) *Philosophy and Phenomenological Research* (Vol. 68, p. 496). Orion Publishing Group, Ltd.
- Manola, F., & Miller, E. (2004). *RDF Primer*. (Frank Manola & Eric Miller, Eds.) *W3C Recommendation* (Vol. 10, pp. 1-107). W3C.
- Mitchell, T. M. (1997). *Machine Learning*. (J. F. Traub, B. J. Grosz, B. W. Lampson, & N. J. Nilsson, Eds.) *Annual Review Of Computer Science* (Vol. 4, pp. 417-433). McGraw-Hill. doi:10.1145/242224.242229
- Mitre. (2008). *Data Analysis Challenges*. Retrieved from <http://www.fas.org/irp/agency/dod/jason/data.pdf>
- Neisser, U. (1976). *Cognition and Reality*. *Psychology* (Vol. 218). San Francisco: W.H. Freeman and Company.
- Nevatia, R. (1982). Machine Perception. *Perception*, 10. Prentice-Hall.
- Nokia. (2008, December). Sensing the World with Mobile Devices: The Vision. *Nokia Technology Insight Series, Nokia Research Center (NRC)*. Retrieved from http://research.nokia.com/files/insight/NTI_Sensing_-_Dec_2008.pdf
- Norwich, K. H. (1991). On the fundamental nature of perception. *Acta Biotheoretica*, 39(1), 81-90.
- Patni, H., Henson, C., & Sheth, A. (2010). Linked Sensor Data. *2010 International Symposium on Collaborative Technologies and Systems*, 362-370. Ieee. doi:10.1109/CTS.2010.5478492
- Pawlowski, A., Guzman, J. L., Rodriguez, F., Berenguel, M., Sanchez, J., & Dormido, S. (2008). Event-based control and wireless sensor network for greenhouse diurnal temperature control: A simulated case study. *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 500-507. Ieee. doi:10.1109/ETFA.2008.4638446
- Pawlowski, A., Guzman, J. L., Rodriguez, F., Berenguel, M., Sanchez, J., & Dormido, S. (2009). The influence of event-based sampling techniques on data transmission and control performance. *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on* (pp. 1-8). doi:10.1109/ETFA.2009.5347045
- Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL Query Language for RDF. (E. Prud'hommeaux & A. Seaborne, Eds.) *W3C Recommendation*. W3C.
- Punuru, J. R. (2007). *Knowledge-based Methods for Automatic Extraction of Domain-specific Ontologies*. Louisiana State University.
- Reggia, J. A., & Peng, Y. (1986). Modeling Diagnostic Reasoning: A Summary of Parsimonious Covering Theory. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 25(2), 125-134.
- Ricquebourg, V., Durand, D., Menga, D., Marhic, B., Delahoche, L., Loge, C., & Jolly-Desodt, A. M. (2007). Context Inferring in the Smart Home: An SWRL Approach. *International Conference on Advanced Information Networking and Applications Workshops* (Vol. 2, pp. 290-295).
- Scheider, S., Probst, F., & Janowicz, K. (2010). Constructing Bodies and their Qualities from Observations. *2010 International Conference on Formal Ontology in Information Systems* (pp. 131-144). IOS Press.
- Sensory Perception. (2010). *Gene Ontology*. Retrieved from <http://www.geneontology.org/GO.doc.sensory-perception.shtml>
- Shanahan, M. (2005). Perception as Abduction: Turning Sensor Data into Meaningful Representation. *Cognitive Science*, 29(1), 103-134. Psychology Press. doi:10.1207/s15516709cog2901_5

- Shannon, C. E. (1948). A Mathematical Theory of Communication. *MD computing computers in medical practice*, 27(4), 306-17. ACM. doi:10.1145/584091.584093
- Sheth, A. (2009). Citizen Sensing, Social Signals, and Enriching Human Experience. *IEEE Internet Computing*, 13(4), 87-92. doi:10.1109/MIC.2009.77
- Sheth, A., Henson, C., & Sahoo, S. S. (2008). Semantic Sensor Web. *IEEE Internet Computing*, 12(4), 78-83. IEEE Computer Society. doi:10.1109/MIC.2008.87
- Stasch, C., Janowicz, K., Bröring, A., Reis, I., & Kuhn, W. (2009). A Stimulus-Centric Algebraic Approach to Sensors and Observations. (N. Trigoni, A. Markham, & S. Nawaz, Eds.) *GeoSensor Networks*, 5659, 169-179. Springer-Verlag. doi:10.1007/978-3-642-02903-5_17
- Suchanek, F. M. (2009). *Automated Construction and Growth of a Large Ontology*. Saarland University.
- Thirunarayan, K., Henson, C. A., & Sheth, A. P. (2009). Situation Awareness via Abductive Reasoning for Semantic Sensor Data: A Preliminary Report, *International Symposium on Collaborative Technologies and Systems* (pp. 111-118). doi: 10.1109/CTS.2009.5067470
- Thomas, C., Mehra, P., Brooks, R., & Sheth, A. (2008). Growing Fields of Interest. *2008 IEEE International Conference on Web Intelligence and Intelligent Agent Technology*, 2(1), 496-502. Ieee. doi:10.1109/WIIAT.2008.358
- Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4), 399-458. ACM. doi:10.1145/954339.954342