IMPROVING ENVIRONMENTAL SENSOR DATA QUALITY USING A CATEGORIZATION OF DATA PROPERTIES

Irbis Gallegos
The University of Texas at El Paso
irbisg@miners.utep.edu

Ann Gates
The University of Texas at El Paso
agates@utep.edu

Craig Tweedie
The University of Texas at El Paso
ctweedie@utep.edu

Abstract: To conduct research on the causes of global environmental changes, environmental scientists have been using advanced technologies, such as wireless sensor networks and robotic trams equipped with sensors, to collect spectral readings, ground temperature, ground moisture, wind velocity, light spectrum, and other data. Indeed, the amount of data being collected is rapidly increasing, and the ability to evaluate promptly the accuracy of the data and the correct operation of the instrumentation being used to collect the data is critical in order to not lose valuable time and information. To address these issues, an approach based on software-engineering techniques is being developed to support the scientist's ability to specify data properties, through guidance using property classifications, which can then be used for near real-time monitoring of data streams. This paper presents a data property categorization scheme associated with sensors used for monitoring the environment, and it describes how the categorization facilitates data property specification and supports improved data quality.

Key Words: Data assurance, data property specification, data quality, information quality, data verification, scientific data.

1. INTRODUCTION

Wireless sensor networks [1] [3] are large-scale ad hoc networks of mostly homogeneous, compact, immobile sensor nodes that are randomly deployed in areas of interest. The measurements taken by sensor nodes are discrete samples of physical phenomenon that are subject to review of their accuracy (dependent on location) [2]. Sensor networks are often used in habitat and environmental monitoring, military surveillance, health care monitoring and other applications [4]. There has been an increase in the use of this advanced field-based technology to study the causes of global environmental changes. In addition, scientists are beginning to use autonomous data collection systems that are capable of acquiring and recording environmental data at regular intervals, allowing them to study remote sites, e.g., a robotic tram equipped with multiple sensors [23]. As a result of these advances, the amount of data acquired in real time has increased, including data such as spectral readings, ground temperature, ground moisture, wind velocity, light spectrum, and temperature under the shade.

The ability to evaluate promptly the accuracy of sensor data and the correct operation of the instrumentation is critical now more than ever, especially since errors can lead to loss of valuable time and information. Causes of errors include noise from external sources (e.g., hardware), inaccuracies and impressions in sampling methods and derived data, faulty equipment, human error, and various environmental effects (i.e., adverse weather conditions) [5]. Common weather factors that can affect sensor data quality include abrupt temperature changes, wetness in the form of mist, precipitation, light conditions, cloud base height, wind speed, and gustiness.

Often, scientists examine the gathered data and use their technical experience and knowledge to determine if the data being gathered correspond to their expectations given the equipment and prevailing weather conditions. The following examples illustrate the difficulty of making decisions in the field concerning the quality of the data being collected:

- The data may not be readily available for analysis and interpretation from the electronic device, e.g., data logger.
- Problems with the equipment, such as battery voltage, extreme differences between the temperature of the instrument and the external temperature, and dark current drifts, might be difficult to identify from the data themselves unless the problem has previously occurred and the situation has been properly documented.
- The scientist or technician in the field may not have the depth of knowledge or experience to identify potential problems.
- As the complexity of the equipment increases, so does the difficulty to identify the origin of the error.

This work focuses on understanding the causes of error in data in the environmental science domain, in particular those obtained through sensors. The importance of data to environmental studies emphasizes the need to develop procedures and mechanisms to verify the integrity of the data. For example, corrupted sensor data can cause miscalculations that might have a major impact in environmental policies. For years, the team studying the data obtained from the Ozone Mapping Spectrometer (TOMS) on board of the Nimbus-7 satellite failed to detect the Earth's stratospheric ozone depletion in some areas [6]. The team failure occurred because the TOMS data analysis software had been programmed to flag and set aside data points that deviated greatly from expected measurements and so the initial measurements, which should have set off alarms, were simply overlooked. In another example, the U.S National Snow and Ice Data Center project underestimated for weeks the extent of Arctic sea ice by five hundred thousand square kilometers due to an undetected sensor drift [7]. The sensor drift went undetected because full checks for real-time data had not been done.

The process to ensure the quality of data can be divided into two stages, a property specification stage and a verification stage. In the property specification stage, a practitioner specifies a set of properties that can be used to check the quality of the data. In the verification stage, a mechanism or system checks that the data adheres to the specified properties. Of course, the quality of data verification is as good as the quality of the properties specified.

Although the research addresses both stages, this paper centers on property specification. This paper introduces a data property categorization that forms the foundation for a specification and pattern system that can assist scientists as they specify properties for checking the correctness of sensor data collection. The categorization was derived from a literature survey of the practices of a representative sample of environmental science related projects with published data verification criteria.

Section 2 reviews data properties obtained the results from the literature survey of documented data quality processes and presents a data property categorization. Section 3 relates the preliminary results of

the categorization by applying it to representative properties. Section 4 discusses the results and how the categorization supports specification and improved data quality. Section 5 discusses the related work, and Section 6 presents a summary and describes the future direction of the work.

2. DATA PROPERTY CATEGORIZATION

Checking the quality of sensor data is an essential step in data processing and requires identifying and analyzing data anomalies. A literature review was conducted to review and analyze current efforts in evaluating data quality documented by a total of 15 projects [9-23] focused on environmental sensor data collection. The projects illustrate how data quality is incorporated into sensor data collection systems and processes at field sites, data centers, or both.

The reviewed projects were in one or more of the following fields: atmospheric studies (6), oceanography (9), meteorology (6), hydrology (1) and land productivity (1). The data collected through the projects include CO_2, carbon balance, energy balance, spectral data, bathythermography, water salinity, tide gauge, vessels data, and temperature and wind profiles.

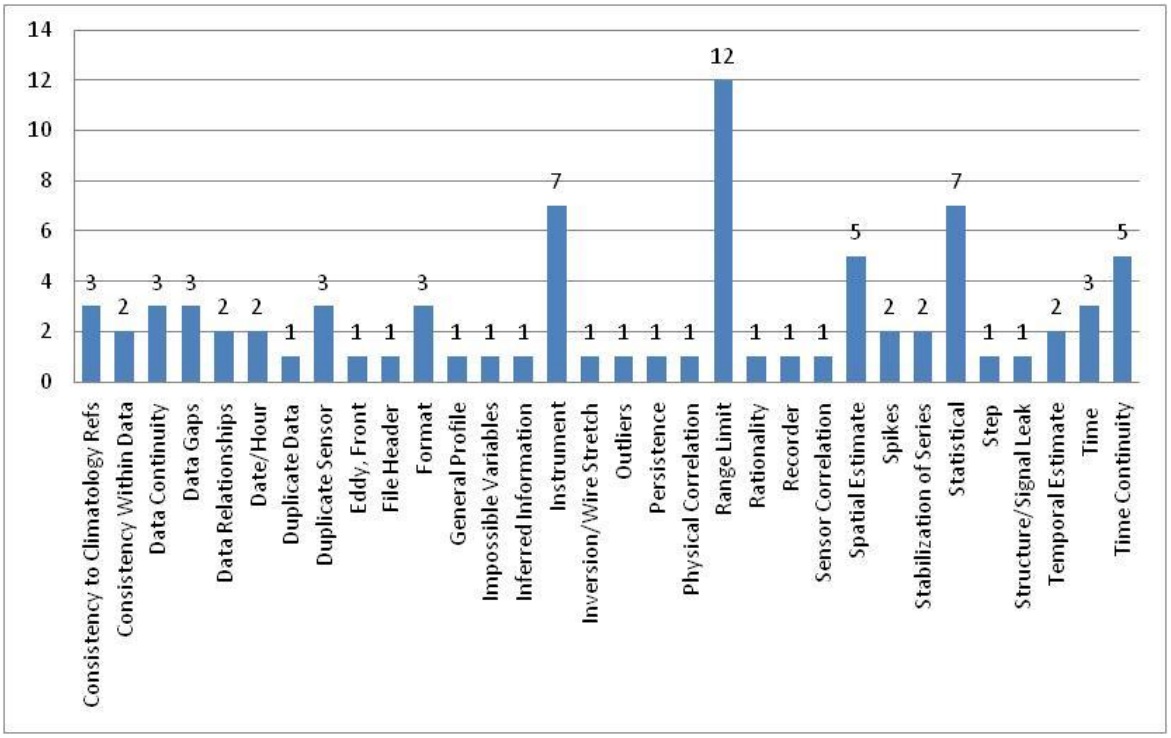

Figure 1. Groupings of data checks and analysis using the terminology of the projects.

The groupings of data checks and analysis gleaned from the projects are summarized in Fig. 1. The number of projects in which they occurred is given with each bar. As shown, the most frequently specified check is *range limit* (i.e., those that capture sensor readings thresholds that are ecologically sound according to the scientists expertise), followed by checks associated with *instrument behavior* (i.e., those related to conditions associated with the instrument during the data collection processes) and those that use *statistical analysis* (i.e., those checks performed after the data is processed and analyzed). *Time continuity* checks, which denote those that have a time-dependent relationship among sensor data readings also play an important role. *Spatial estimates* refer to checks that identify expected data values for

environmental and physical conditions that influence experiments.

Of the projects studied, an observation was that different projects use different terminology to describe similar checks or properties of the data. For example, *range checking* can apply to checks that identify *outliers* and *spikes*. Those classified as *data continuity* include checks that could be referred to as *data gaps*, *data relationships*, or *persistence* in various projects. *Time* and *date/hour* are considered the same. *Physical correlation* could be interchangeable with d*uplicate sensor* and *sensor correlation*.

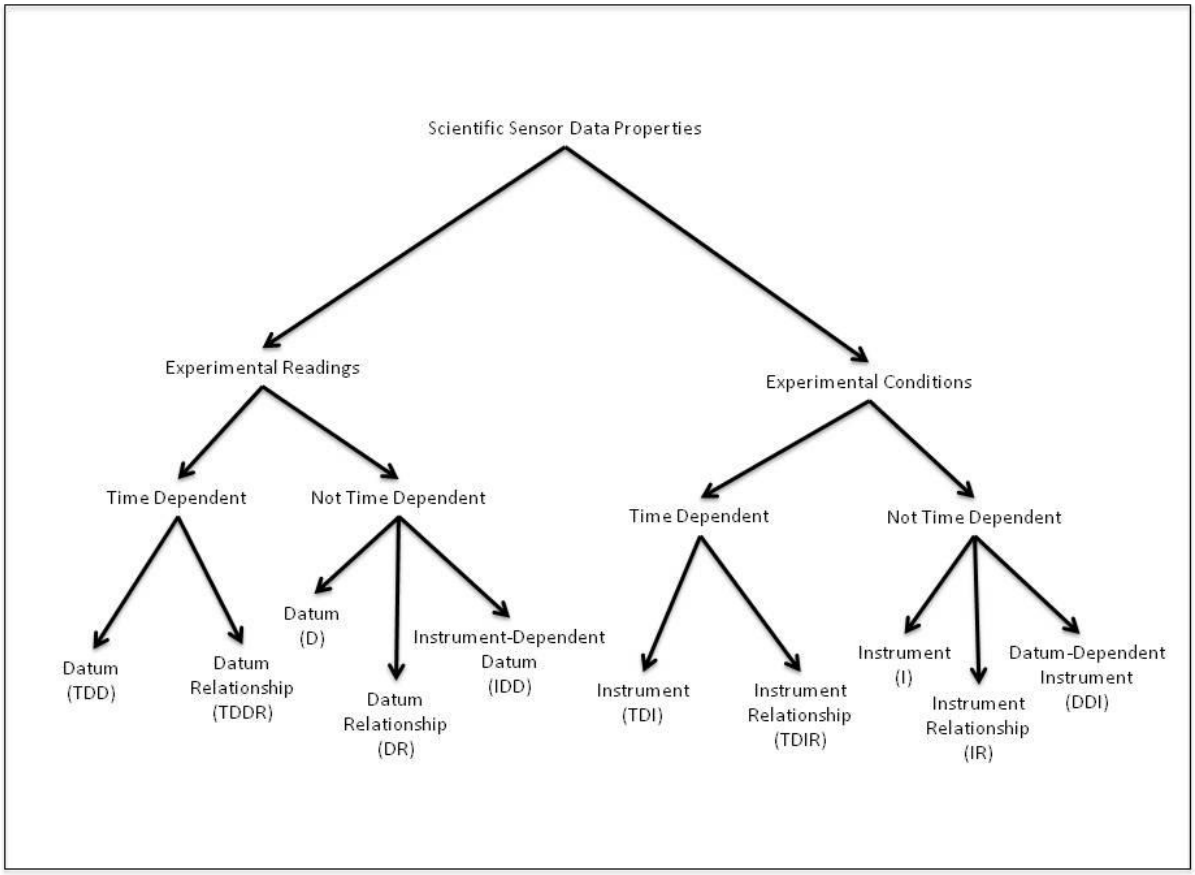

Figure 2. Scientific sensor data properties categorization.

From analysis of the checks described in the previous section, the property categorization shown in Fig. 2 resulted. The classification divided the properties into two major types: *experimental readings and experimental conditions*. *Experimental readings* properties specify expected values and relationships related to field data readings and can be used to identify anomalies in a dataset, as well as random data errors, i.e., those errors that can be detected, estimated, and minimized by examining the convergence of calculations with increasing size of data sets [24, 25]. *Experimental conditions* properties specify expected instrument behavior and relationships by defining examining attributes (e.g., voltage) and instrument functions based on readings. This type of properties can identify systematic errors, i.e., persistent offsets or multipliers that can affect the whole or a portion of the dataset [25]. The values being checked may be sensor readings, derived values based on one or more sensor readings, pre-defined values, and historical values.

Properties labeled *experimental readings* are divided into the following five subcategories:

- *Datum:* A datum (D) property specifies the expected value of a single sensor reading. A sensor reading is compared against a pre-defined or historical value. Example: *The relative humidity percentage should always be greater than or equal to 0 and less than or equal to 103* [16].
- *Time-Dependent Datum:* A time-dependent datum (TDD) property specifies the expected value(s) of a single type of sensor, where the readings are filtered by date and time. The selected sensor readings are compared against a predefined value or a historic value. Example: *During daylight on May 12th, the dry bulb temperature should be less than or equal to 1* [8].
- *Datum Relationship:* A datum-relationship (DR) property specifies the relationship between two or more types of sensor readings. A DR property can be used to compare sensor readings against readings from other types of sensors, against a predefined constant value, or against an historic value. Example: *Temperature < Wet-Bulb-Temperature < Dew-Point-Temperature* [12].
- *Time-Dependent Datum Relationship:* A time-dependent datum relationship (TDDR) property specifies the relationship between two or more related sensor readings that are filtered based on time. The selected readings may be compared against each other, against a predefined value, or an historic value. TDDR properties capture relationships within time series data and datasets behaviors dependent on time. Example: *No two measurements of the consensus subset can differ by more than 1/8 of the maximum measurable velocity, where the consensus subset is created each hour by applying the consensus algorithm from the ten 6-minute radial velocity measurements on each antenna beam* [15].
- *Instrument-Dependant Datum:* An instrument-dependant datum (IDD) property is one that specifies a property about an instrument that influences behavior of the sensor readings. Example: *If the profile lies close to land and the depth is less than 50 meters, the observed value should lie within 5 standard deviations from the mean value* [19].

Experimental conditions properties are divided into the following five subcategories:

- *Instrument:* An instrument (I) property specifies the expected behavior of an instrument by describing an attribute of the instrument. The attribute is compared against either a predefined value or an historic value. Example: *The real-time sensor voltage should fall inside the expected range* [21].
- *Time-Dependent Instrument:* A time-dependent instrument (TDI) property captures the expected behavior of a single instrument that is dependent on time. The instrument reading is compared against a predefined constant value, a historic value, or a time entity in a given time constraint. Example: *Based on the time since last scanned, each radar must scan a 360-degree sector at the lowest two elevations every 2.5 minutes* [13].
- *Instrument Relationship:* An instrument relationship (IR) property captures the relationship between one or more related instruments. An IR property can be used to compare the behavior of the instrument. Example: *If a current meter is used, at least one of the HCSP/HCDT or NSCT/EWCT sensor couples must be present* [14].
- *Time-Dependent Instrument Relationship:* A time-dependent instrument relationship (TDIR) property captures the relationship between two or more related instruments and expected behavior based on time. A TDIR property can be used to compare instrument behavior dependent on a time. Example: *Based on the time since last scanned, perform sector scans of storms with 2 or more radars every 1-minute* [13].
- *Data-Dependant Instrument:* A datum-dependant instrument (DDI) property captures a known datum or datum relationship whose value influences instrument behavior, or causes an instrument's action. DDI properties capture continuity problems. Example: *If there is no change in current direction data, the system must generate an error alert* [11].

3. ANALYSIS OF THE DATA PROPERTY CATEGORIZATION

Using the data property categorization given in Fig. 2, a total of 532 properties from the aforementioned projects were analyzed and classified. The process took three iterations and resulted in refinement of the categorization. These iterations are labeled as "initial categorization," "revised categorization," and "tool categorization."

The initial categorization had eight categories: *datum, time-dependent datum, datum relationship, time-dependent datum relationships, instrument, time-dependent instrument, instrument relationship*, and *time-dependent datum relationship*. As shown in Fig. 3, the initial categorization classified 386 properties as *experimental readings*, 82 properties were classified as *experimental conditions*, and 53 properties were not classifiable. Figs. 4 and 5 compare the number of properties classified for each subcategory of type *experimental readings* and *experimental conditions*.

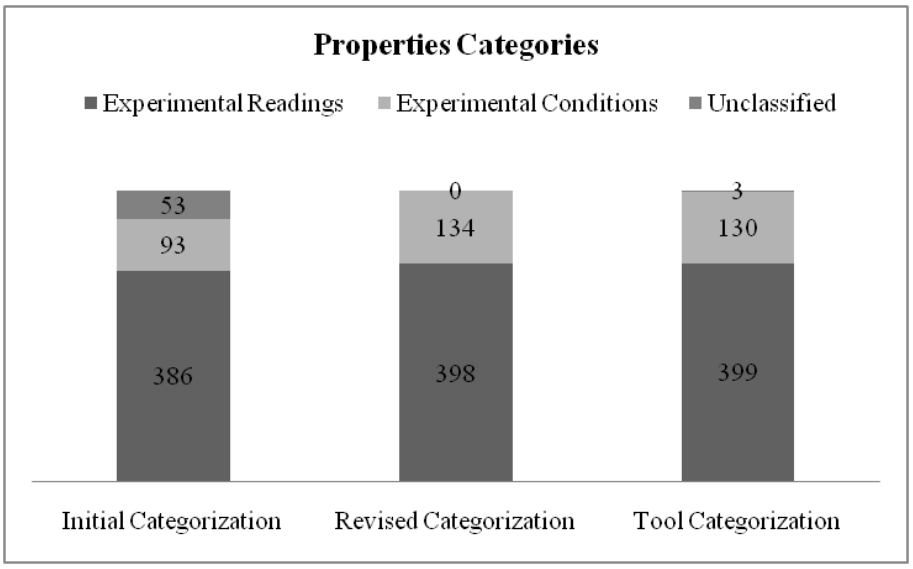

Figure 3. Property categorization results.

The initial categorization was refined and extended to increase its coverage by adding the instrument-dependent datum and the data-dependent instrument categories. The classification of properties under the revised categorization is shown in Fig. 3. The new categorization resulted in a discrepancy between the number of initial properties placed in a particular category and those placed in a category using the revised categorization properties (other than the unclassified properties). There were properties in the initial categorization that were unclassified. Once the categorization was revised, some of the unclassified properties were placed in one of the new categories.

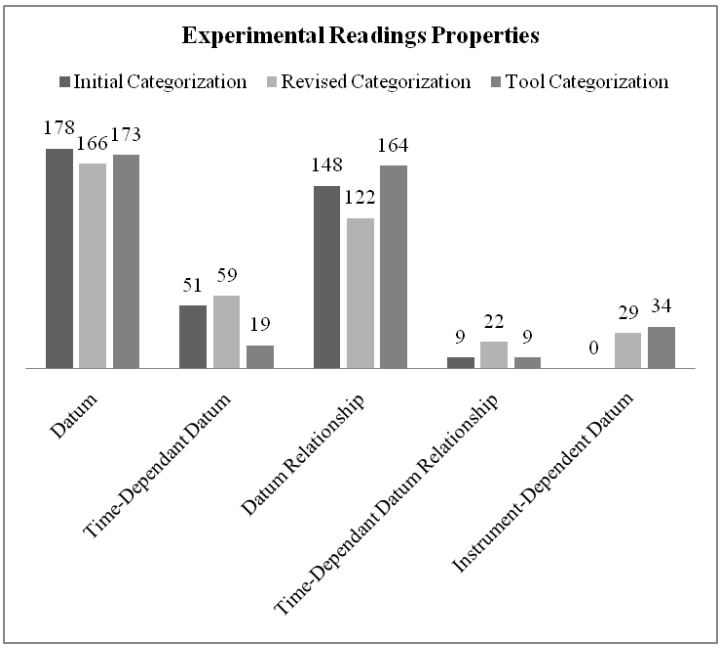

Figure 4. *Experimental readings* properties categories distributions.

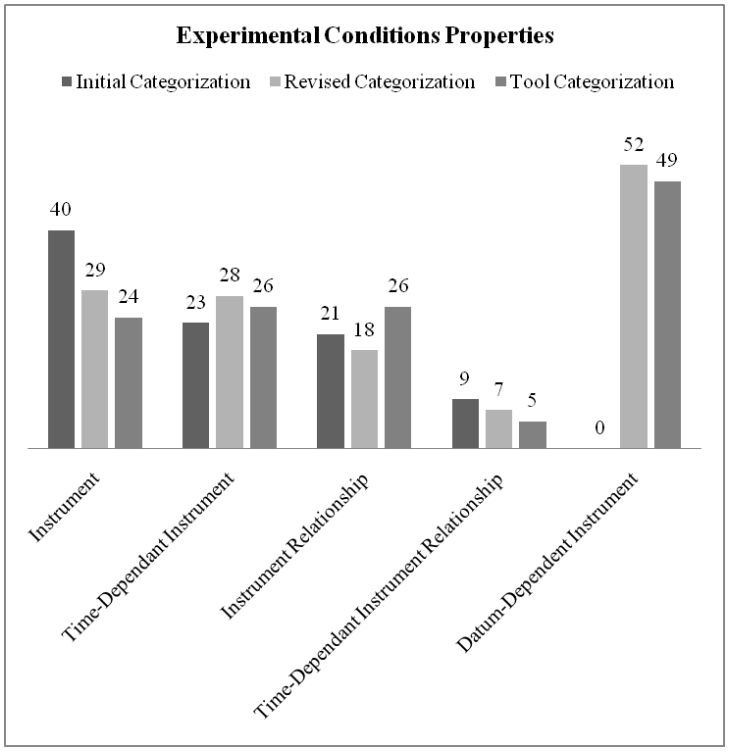

Figure 5. *Experimental conditions* properties categories distributions.

The results obtained by the categorization revealed that the studied environmental projects captured more experimental than systematic properties. A possible explanation is that scientists have concentrated less on instrument malfunctions even though the latter can be a major source of errors in the data. In addition,

with *experimental readings* properties, datum properties and data relationships were specified more frequently than time-related properties. In the systematic errors category, datum-dependent instrument properties ranked higher than the other categories, an indication that scientists use data inspection to determine instrument malfunctions instead of specifying separate instrumentation properties.

Several factors that can limit the effectiveness of the categorization for data property specification were identified during the property categorization process. Some data properties are described at such an abstract level making it difficult to translate such a property into a specification that could be automatically verified. Other data properties were complex, required them to be decomposed into several simpler properties. Due to the inherited ambiguous nature of natural languages, data properties descriptions are sometimes too ambiguous and are difficult to determine the intended property meaning.

A number of specifications are a combination of data verification and data steering properties. Combined property specifications require both verifying that the properties adhere to predefined behaviors, the verification aspect, and guaranteeing that a reaction occurs in response to a data or instrument stimulus, the steering aspect. As a result, combined property specifications must be decomposed into separate data verification properties and data steering properties.

4. PROPERTY SPECIFICATION BASED ON CATEGORIZATION

The data property categorization resulted in development of *Data Property Specification (DaProS)*, a scientist-centered prototype tool that uses the categorization to assist the user in specifying a data property. Through a series of guiding questions and selections, the user identifies the appropriate category and enters required information, and the tool yields the appropriate specification as well as a disciplined natural language representation of the specification for validation purposes. The DaProS prototype tool was used to categorize 399 properties as *experimental readings* properties and 122 properties as *experimental conditions*. Because some properties were too ambiguous to be classified, there was a difference between the number of properties classified by the tool and the number of properties classified manually. This section describes how the categorization has been used to define the DaProS tool.

4.1 Basis for DaProS

The Specification and Pattern System (SPS) [29], a software engineering solution for specifying and refining properties about critical software systems, provides the foundation for the approach used to specify data quality properties using a categorization system. In SPS, a *pattern* describes the essential structure of some aspect of a system's behavior and provides expressions of this behavior in a range of formal specification languages and formalisms. Each pattern is associated with a scope, which is the extent of the program execution over which the pattern must hold. The SPS was adapted to create the *Data Property Specification and Pattern System (DA-SPS)*, which uses scopes, patterns and Boolean statements to specify data properties. *Boolean statements* express data properties, which are defined using mathematical relational operators that are applied to a datum, datum relationships, and Boolean methods that are available to the scientist.

4.2 Data Property Categorization and DA-SPS Patterns

In DA-SPS, a *property scope* delimits the subset of data over which a property holds. The scope is defined by specifying the datum occurrences in a dataset Δ over which a property will hold. Given $L \epsilon \Delta$ and $R \epsilon \Delta$, a practitioner delimits the scope of a property by designating one of the following types:

- *Global:* the property holds for all the data in dataset Δ;
- *Before R*: the property holds over the sequence of data that begins with the first datum in Δ and ends with the datum immediately preceding the first datum in Δ that matches *R*;
- *After L*: the property holds over the sequence of data starting with the first datum in Δ that matches *L* and ending with the last datum in Δ;
- *Between L and R*: the property holds over the sequence of data starting with the first datum in Δ that matches *L* and ending the datum immediately preceding the first datum that matches *R*; and
- *After L until R:* the property holds over the sequence of data starting with the first datum that matches *L* and ends with the datum immediately preceding either the first datum that matches *R*, or the last element in Δ if datum *R* does not occur.

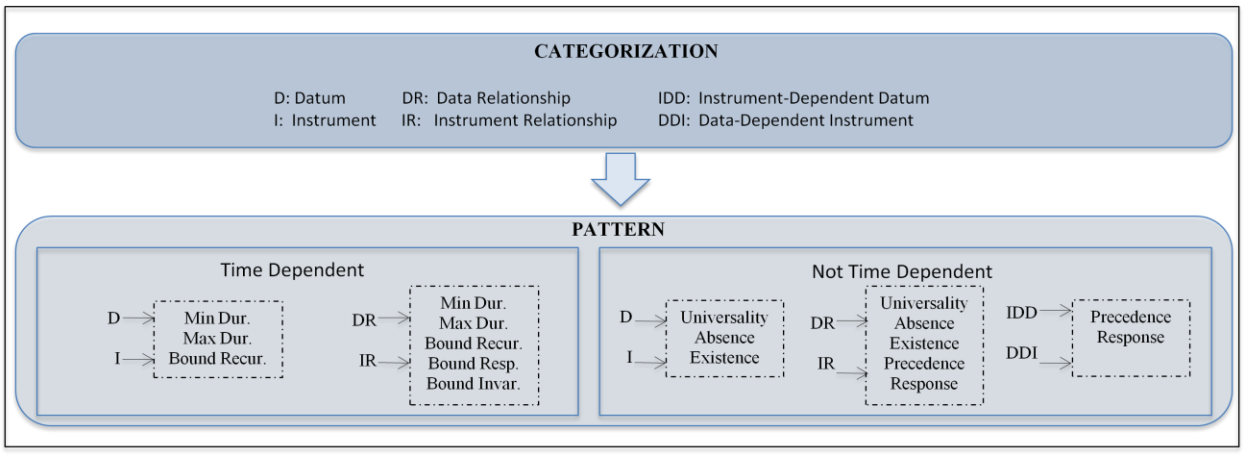

Figure 6: The relationship of the categorization with the pattern choices.

A *property pattern* is a high-level abstraction describing a commonly occurring property about a scientific dataset. Users typically select patterns through a variety of decisions. As described earlier, the patterns are grouped as *experimental readings*, which describe the expected behavior of the data, and *experimental conditions*, which describe external conditions such as those associated with the functioning of the instrument or weather conditions. Fig. 6 summarizes how the categorization is used to drive the pattern selection.

Time dependent patterns are interpreted over a discrete time domain, e.g., over the natural numbers ℕ. Timed patterns assume a system clock, where the clock is treated as a local entity for each dataset value. For timed patterns, it is assumed that the independent value associated with each dataset value is a discrete time t. A time constrained property specifies one of the following:

- *Minimum Duration(P,c):* Boolean function *P* holds for a minimum of *c* units of time;
- *Maximum Duration (P,c)*: Boolean function *P* holds for a maximum of *c* units of time;
- *Bounded Recurrence(P, c)*: Boolean function *P* holds every *c* units of time;
- *Bounded Response(T,P,c)*: Boolean function *T* holds after Boolean function *P* holds at no more than $t + c$ time, where *t* is the time that *P* holds; and
- *Bounded Invariance(T,P,c)*: Boolean function *T* holds for at least $t + c$ time before Boolean function *P* holds, where *t* is the time that *T* holds.

Patterns associated with categories that are not time dependent are specified as follows:

- *Universality(P):*Boolean function *P* always holds over dataset Δ;
- *Absence(P)*: Boolean function *P* never holds over dataset Δ;
- *Existence(P)*: Boolean function *P* holds at least once over the dataset Δ;
- *Precedence (T,P):* Boolean function *T* holds before Boolean function *P* eventually holds; and
- *Response (T,P)*: Boolean function *T* holds immediately after Boolean function *P* holds.

The data property categorization can be used to help practitioners determine which property pattern best suits the data property to be specified. Fig. 6 presents the relationship between the property categories and the DA-SPS patterns. The data categories are related to a property pattern depending on whether the property to be specified is time-dependent or not and by the number of entities required to specify the property.

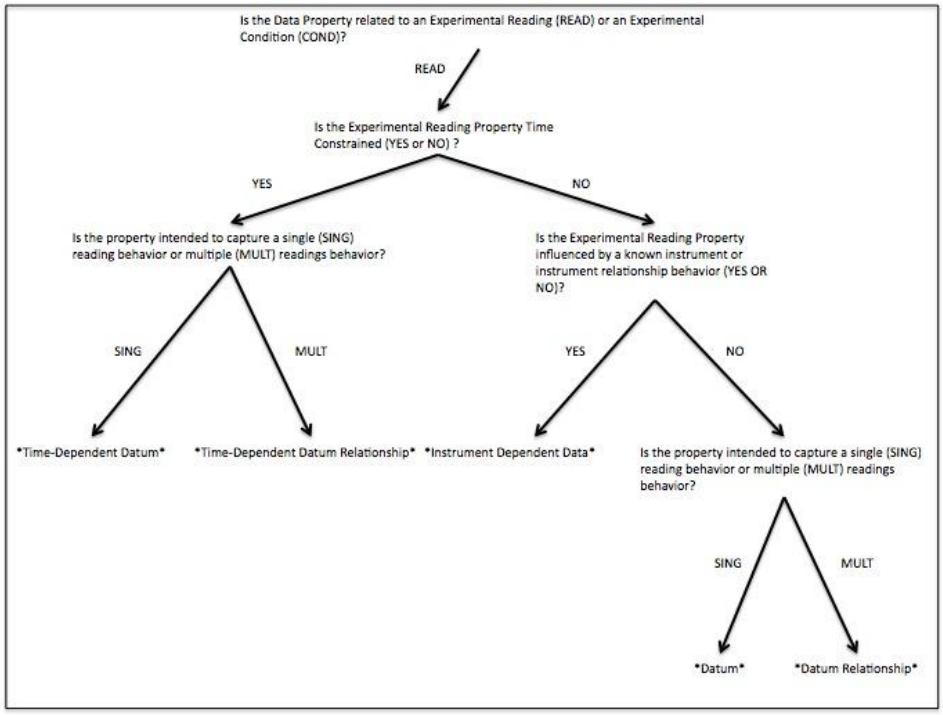

Figure 7. Data property categorization decision tree for experimental readings.

For time-related categories that require a single value representing time (*Time-Dependent Datum, Time-Dependent Instrument*), the patterns are restricted to the following patterns: *Minimum Duration, Maximum Duration, and Bounded Recurrence*; these patterns support verification of a Boolean statement depending on a time-dependency being satisfied. For time-related categories that require two or more values (*Time-Dependent Datum Relationship, Time-Dependent Instrument Relationship*), the supporting patterns are: *Bounded Response* and *Bounded Invariance*; these properties verify two or more Boolean statements, which may be given as a function, depending on a time-dependency being satisfied. Similarly, for non-time constrained single entity properties (*Datum, Instrument*), the data patterns are restricted to *Absence, Universality* and *Existence*; these patterns allow practitioners to specify properties about a single Boolean statement. For non-time constrained properties requiring two or more entities (*Datum Relationship, Instrument Relationship*) with entities of the same type, the patterns are as follows: *Absence, Universality, Existence, Precedence, Response*; These patterns allow two or more Boolean statements or functions to be verified given that both Boolean statements or functions belong to the same

general category (*experimental condition* or *experimental readings)*. For non-time constrained properties requiring two or more entities with entities of different categories (*Instrument-Dependent Datum, Datum-Dependent Instrument*), the patterns are restricted to *Precedence and Response*.

Tool support for the specification process through DaProS allows practitioners to select the correct data pattern for an intended property. The DaProS graphical user interface automatically restricts the pattern to be used depending on the selected data category. If the practitioner is undecided about which data property category to use, DaProS uses decision trees and guided questions to help the practitioner decide on a data property category. Fig. 7 depicts the decision tree for experimental readings properties.

4.3 An Example Scenario

To illustrate the strengths of the DA-SPS and the data property categorization, consider the following scenario. Nailea, who is an environmental scientist conducting research in oceanography, wants to specify the following data property:

"For sea depths less than 25 m, the salinity should be less than 41 psu."

The data property will be verified at near-real time as the data is streamed and collected. Nailea is undecided on what data category captures the intended meaning of her property, so she uses a decision tree to help her select the correct classification for the data property. Nailea decides that her property should capture an *experimental reading* because she is interested on checking the salinity reading value. Then, she decides that the data property is not *time-constrained* based on her knowledge about the environmental system with which she is working. Nailea realizes that the salinity level depends on the sea depth at which the measurement is taken; the salinity reading is influenced by the proper functioning of an outside instrument which, in this case, is the side-scan sonar used to measure the water depth. Nailea decides to use an *Instrument Dependent Data* as her category.

To specify her property, Nailea decides that she interested in looking at all of the data being obtained by her salinity sensor, so she chooses *Global* as her *scope*. Based on her given category choice, Nailea can choose between *Precedence* and *Response* pattern. Because Nailea wants to make sure that while the side-scan sonar is less than 25 m, the salinity is less than 41 psu, she selects *Response* as her pattern. The side-scan sonar does not store the current sea depth in the datalogger, but the value can be accessed by a monitoring control function *getSeaDepth(); getSeaDepth()* returns an integer value representing the current sea depth when the function is called. Because the sea depth is not logged, the value is not considered a reading. In this case, Boolean statement *P* is defined as *getSeaDepth()<25m*, and Boolean statement *T* is defined as *salinity<41 psu*. The complete specification is summarized in Table 1. The specification can be used as input to a near-real time data verification process

Table 1. Data property specification summary after the specification processes is completed.

Property: *"For sea depths less than 25 m, the salinity should be less than 41 psu."*
Category: Instrument Dependent Data **Scope:** Global (for all salinity values in the dataset) **Pattern:** Response (T, P), which is read as T responds to P **Boolean Statements:** T: salinity<41 psu P: *getSeaDepth()<25 m*
Discipline Natural Language Description: For all dataset values, it is always the case that if *getSeaDepth()<25 m* holds, then immediately *salinity<41psu* holds.

.

4.4 Impact on Data Quality

The process to ensure the quality of data can be divided into two stages, a property specification stage and a verification stage. In the property specification stage, a practitioner specifies a set of properties that can be used to check the quality of the data. In the verification stage, a mechanism or system checks that the data adheres to the specified properties. The focus of this work is on property specification. Indeed, the quality of the properties specified can influence the quality of data verification.

The use of a scientific data property categorization to specify properties encourages scientists to further analyze and refine properties for specific ecosystems, increases the scientists' ability to reuse properties and to document expert knowledge, fosters standardization of scientific processes related to data quality, and allows data properties to be interpreted and verified by data verification mechanisms.

With the data property categorization in place, scientists have the ability to further analyze and refine properties for their specific ecosystems. In the environmental sciences, it is difficult for scientists to distinguish true errors from anomalies generated by environmental events. The approach presented in this paper allows scientists to fine tune data properties that can distinguish errors from environmental events. For instance, in Eddy Covariance data, data obtained during strong rainy conditions are considered bad data, yet, it is difficult for scientists to determine when a rain event occur just by looking at the data. For the example scenario in the previous subsection, Nailea could specify properties to capture rain events by specifying properties that identify sudden changes in temperature or atmospheric pressure.

The data quality assurance process can be improved in several ways using the proposed approach. Capturing data properties formally allows the scientist to document knowledge about scientific domains, and this in turn facilitates knowledge sharing and reuse by other scientists. For the example scenario in the previous subsection, Nailea's property shown in Table 1 can used by other scientists wanting to define similar properties by substituting the parameter values *25 m* and *41 psu* with the appropriate values accordingly to the ecosystem being studyied. The changes can be performed at the specification level eliminating the need to make modifications to source code as is often the case in many monitoring systems.

The data property specification process can support standardization of data quality processes for similar scientific communities. A set of data properties can be specified to cover the needs of specific ecosystems and be shared by members of a community. Tool support will allow scientists to discuss and refine existing and new properties. The common data property set will allow scientists to verify the data being collected using the same properties and tools, thus moving toward a unified way of verifying data.

DaProS abilities to generate properties in an exchangeable format and to mitigate ambiguity allow scientists to use the generated properties as input to data verification mechanisms that can interpret and verify such properties over scientific datasets. Toward this effort, a prototype Sensor Data Verification (SDVe) tool has been developed to verify the quality of the data from the specifications generated by DaProS. SDVe takes as input a property specification file generated from DaProS and a sensor data dataset file obtained from a data logger, and verifies that the data in the dataset adhere to the property specified in the property specification file. SDVe raises alarms whenever the data property is not satisfied by the data. SDVe is implementation agnostic and data-type agnostic. For the example in Section 5, Nailea uses DaProS to specify the property and to generate the property specification file. Nailea uses the generated property specification file and the side-scan sonar hourly data file as input to the SDVe. Because the property scope is Global, the SDVe will verify that for all measurements taken during the day below the 25 m limit, the salinity level is less than 41 psu. If alarms are raised by the SDVe, Nailea can immediately analyze the data and determine probable causes for the violation, or can experiment with the

property by adjusting the salinity level threshold. This approach could help Nailea determine at near-real time if the anomalies in the data are true errors or environmental features with scientific implications about the ecosystem.

5. RELATED WORK

There are several efforts that have developed categorizations of data properties mostly based on classes of queries related to sensor networks. These categorizations rely on traditional SQL-like queries and aggregates or probabilistic range queries for moving objects.

Elnahrawy and Nath [5] categorized data properties into four categories: Single Source Queries (SSQ), Sent Non-Aggregate Queries (SNAQ), Summary Aggregate Queries (SAQ), and Exemplary Aggregate Queries (EAQ). SSQ return the value(s) of the attribute(s) of a specific sensor and no aggregation is involved. SNAQ return the set of sensors that satisfy a given user-defined predicate. The predicates are assumed to be simple range queries on one or more attributes and are allowed to include AND and OR operands. SAQ are queries performed using one of the following aggregate functions: SUM, COUNT, and AVG. EAQ are queries performed using one of the following aggregate functions: MIN and MAX. This approach is based only on the sensor data and do not consider other data entities such as instrumentation functioning.

Bonnet et al. [26] suggest classifying the queries as historical, snapshot and long-running queries. Historical queries aggregate queries over historical data obtained from the device network. Snapshot queries concern the device network at a given point in time. Long-running queries concern the device network over a time interval. In this approach, queries are formulated in Structured Query Language (SQL) with minimal additions to the language. This approach is tied to a distributed query-processing model that is not in place for all wireless sensor networks.

Madden et al. [27] classify data aggregates according to their state requirements, tolerance of loss, duplicate sensitivity, and monotonicity. Duplicate sensitivity implies restrictions on network properties and on certain optimizations. Exemplary aggregates return one or more representative values from the set of all values, and summary aggregates compute properties over all values. Monotonic aggregates are used to determine whether some predicates can be applied in the network before the final value of the aggregate is known. Finally, the state requirements refer to the amount of space required to store partial aggregate states. The classification is tailored to match sensor networks properties.

In their work, Cheng et al. [28] present a classification of probabilistic queries. The authors identify two dimensions for classifying database queries, by nature of the answer and by aggregation. Value-based Non-Aggregate queries return an attribute value of an object as the only answer and involve no aggregate operators. Entity-based Non-Aggregate queries return a set of objects, each of which satisfies the condition(s) of the query, independent of other objects. Entity-based aggregate queries return a set of objects that satisfy an aggregate condition. Value-based aggregate queries involve aggregate operators that return a single value. This approach requires a deep understanding of the data and the different probabilistic measurements associated with the data.

6. SUMMARY

The number of sensor networks that collect environmental data at research sites is rapidly increasing, and scientists need to be assured that the collected data sets are correct. In order for data verification mechanisms to assist and be effective, it's essential to understand the type of properties that are of interest to environmental scientists and to be able to specify such properties. The specification of data properties has been limited by many factors such as the lack of reusable properties [8], ambiguity in natural languages when describing properties, complexity in properties when dealing with time and multiple criteria, and the lack of scientists' technical knowledge required to specify properties formally (in the general case). Furthermore, the use of embedded or hard-coded property checking in many existing systems makes it difficult to reuse and refine properties.

To address these issues, the authors developed an approach based on software-engineering techniques to support the scientist's ability to specify data properties formally through guidance based on property categories. The approach supports near real-time monitoring of data streams. This paper presents the results of a study of properties captured by a wide variety of projects that use sensors for monitoring the environment, which resulted in a categorization scheme. Published data quality verification criteria provided a view of properties being used to check the quality of sensor data. Applying the categorization to the projects' documented data properties resulted in deeper understanding of the properties and the specification process. The outcome was a refined categorization scheme and the data property specification and pattern system (DA-SPS) that supports full specification of sensor-data properties.

7. REFERENCES

[1] K. Römer, F. Mattern. "The Design Space of Wireless Sensor Networks," in *IEEE Wireless Communications*, Vol. 11, No. 6. 2004. pp. 54-61.

[2] S. Tilak, N. Abu-Ghazaleh, W. Heizelman. "A Taxonomy of Wireless Micro-Sensor Network Models," in ACM Mobile Computing and Communications Review (MC2R), Vol. 6, No.2. 2002.

[3] M. Tubaishat, S. Madria. "Sensor Networks: An Overview," in *IEEE Potentials*, Vol. 22, No. 2, April/May 2003. pp. 20-23

[4] D. Wagner, R. Wattenhofer. "Algorithms for Sensor and Ad Hoc Networks: Advanced Lectures", in *Lecture Notes in Computer Science*, Springer, 2007

[5] E. Elnahrawy, B. Nath. "Cleaning and Querying Noisy Sensors," in *Proceedings of ACM WSNA'03*, 2003. 78-87

[6] M. King, D. Herring, "Research Satellite for Atmospheric Sciences, 1978-present," in *Encyclopedia of Atmospheric Sciences*, Academic Press, 2002.

[7] A. Morales. (Feb 20,2009). "Arctic Sea Ice Underestimated for Weeks Due to Faulty Sensor," Available: http://www.bloomberg.com/apps/news?pid=20601110&sid=aIe9swvOqwIY

[8] H. Zimmermann, "Check of Meteorological Station Data," in Report of Research Center for Urban Safety and Security Kobe University. 1998. pp. 235-240

[9] National Oceanic and Atmospheric Administration, "Handbook of Automated Data Quality Control Checks and Procedures of the National Data Buoy Center," Stennis Space Center. 2003

[10] Canada Federal Department of Fisheries and Oceans, "Data Quality Assurance (QC) at the Marine Environmental Data Service (MEDS)," DFO ISDM Quality Control. http://www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog_Int/WOCE/WOCE_UOT/qcproces_e.htm. January 21, 2009

[11] Tropical Atmosphere Ocean Project, "Data Quality Control," The TAO project: Data Quality Control. http://www.pmel.noaa.gov/tao/proj_over/qc.html. January 22, 2009

[12] S. Smith, C. Harvey, D. Legler, "Handbook of Quality Control Procedures and Methods for Surface Meteorology Data," WOCE Report No. 141/96. 1996

[13] B.Philips, D. Pepyne, D. Westbrook, E. Bass, J. Brotzge, W. Diaz, K. Kloesel, J. Kurose, D. McLaughlin, H. Rodriguez, M. Zink, "Integrating End User Needs Into System Design and Operation: The Center for Collaborative Adaptive Sensing of the Atmosphere (CASA)," In *16th Conference on Applied Climatology*. 2007
[14] D. Bardet, "System of Control Oriented Oceanographic Parameters SCOOP". User Manual. 2000
[15] W. Lambert, F.J. Merceret, G.E.Taylor, J.G. Ward, "Performance of Five 915-MHz Wind Profilers and an Associated Automated Quality Control Algorithm in an Operational Environment," In *Journal of Atmospheric and Oceanic Technology*. Volume 20. 2003. pp 1488-1495
[16] M.A. Shafer, C.A. Fiebrich, D.S. Arndst, S.E. Fredrickson, T.W. Hughes, "Quality Assurance Procedures in the Oklahoma Mesonetwork," in *Journal of Atmospheric and Oceanic Technology*. Volume 17. 2000. pp 474-494
[17] M.J. Garcia, B. Perez, F. Raicich, L. Rickards, E. Bradshaw, "Quality Control of Sea Level Observations," European Sea Level Service-Research Infrastructure (ESEAS-RI) Work Package 1, Task 1.2. 2005
[18] R. Bailey, A. Gronell, H. Phillips, E. Tanner, G. Meyers, "Quality Control Cookbook for XBT Data," CSIRO Marine Laboratories Report 221. Version 1.1. 1994
[19] Global Temperature-Salinity Pilot Project, "GTSPP Real-Time Quality Control Manual," Manual and Guides #22. January 22, 2009. http://www.meds-sdmm.dfo-mpo.gc.ca/ALPHAPRO/gtspp/qcmans/mg22/guide22.htm
[20] D.J. Doong, S.H. Chen, C.C. Kao, B.C. Lee, S.P. Yeh. "Data Quality Check Procedures of an Operational Coastal Ocean Monitoring Network," Ocean Engineering. Volume 34. 2007. pp 234-246
[21] R. A. Peppler, K. E. Kehoe, K.L. Sonntag, C.P. Bahrmann, S.J. Richardson, S.W. Christensen, R.A. McCord, K.J. Doty, R. Wagener, R.C. Eagan, J.C. Liljegren, B.W. Orr, D.L. Sisterson, T.D. Halter, N.N. Keck, C. N. Long, M.C. Macduff, J.H. Mather, R.C. Perez, J.W. Voyles, M.D. Ivey, S.T. Moore, K.L. Nitschke, B.D. Perkins, D.D. Turner, "Quality Assurance of ARM Program Climate Research Facility Data," U.S Department of Energy, Office of Science, Office of Biological and Environmental Research. 2008
[22] A. Wong, R. Keeley, T. Carval, "Argo Quality Control Manual," Version 2.4. Feb 17, 2009
[23] J. Gamon, Y. Cheng, H. Claudio, L. MacKinney, D.A Sims, "A Mobile Tram System for Systematic Sampling of Ecosystem Optical Properties," Remote Sensing of Environment 103. 2006, 246-254
[24] J.R. Taylor, "An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements," 2nd Ed., University Science Books, 1997. pp 94
[25] J.B. Moncrief, Y.Malhi, R. Leuning, "The Propagation of Errors in Long-Term Measurements of Land-Atmosphere Fluxes of Carbon and Water," in *Global Change Biology*, Vol. 2, Issue 3, 2006. pp 231-240
[26] P. Bonnet, J. Ghrke, P. Seshadri. "Querying the Physical World," in *IEEE Personal Communication*, Vol. 7, Issue 5, 2000. pp 10-15
[27] S. Madden, M. J. Franklin, J. Hellerstein, W. Hong., "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks," in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI'02)*, 2002.
[28] R. Cheng, D. V. Kalashnikov, S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," in *Proceedings of ACM SIGMOD 2003*, 2003.
[29] M.B. Dwyer, G.S. Avrunin, and J.C. Corbett: A System of Specification Patterns. In: proceedings of the 2nd Workshop on Formal Methods in Software Practice (1998)