Adaptive Sensor Fusion Architecture Through Ontology Modeling and Automatic Reasoning

Enrique Martí, Jesús García and José M. Molina¹

Abstract—This paper presents a solution for implementing context-based self-adaptive sensor fusion systems. The adaptation process works over an ontology-based description of the problem space that includes sensors and other information sources, a repository of algorithms, and data types managed by the fusion system. An automatic reasoning module integrates this description with contextual information of the system, and determines how to combine available solution elements, to produce a fused output that best satisfies the goals of the system.

Our proposal keeps the system working in the best conditions under events that include (a) intermittent sensor availability, (b) changing fusion requirements and (c) uneven information quality. Compared with existing proposals, our solution provides a generic mechanism to integrate arbitrary external factors in the adaptation process, such as context-related events, constraints and specific knowledge about the algorithms.

We present an example on ground vehicle navigation, which combines on-board sensors with those available in a smartphone.

I. INTRODUCTION

Adaptability is a key concept in the path towards intelligent systems that can deliver optimal response in an open and always changing world. Most sensor fusion applications have a closed design that combines a fixed sensor set configuration with a processing logic that is thought to deliver best results, but this type of solutions can show degraded performance when the working conditions change.

The context of a system has a direct influence on its performance, understanding context as "every factor that constraints or affects the process of solving a problem, without being part of the problem or the solution itself". It has been proved that exploiting our knowledge about this context can help in the detection of unfavorable conditions and subsequent application of corrective actions. Some works as [1][2] show how to infer and apply context information directly to fusion algorithms for increasing its accuracy and robustness, but the potential of context information extends to other applications as algorithm and sensor selection. The question of what is context and what is just part of the problem can be rather complex and subtle in some border cases. We provide a brief discussion on this subject later on.

This work presents a system that can solve generic fusion problems using an arbitrary, dynamic set of sensors and a repository of processing algorithms, incorporating context

This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485).

¹Group of Applied Artificial Intelligence, Department of Computer Science, University Carlos III de Madrid, Colmenarejo CP28270, Spain. Corresponding author: Enrique Martí emarti at inf.uc3m.es information to enhance the adaptability and quality of the final solution. This information, which receives the name of Problem Space Characterization, is expressed in terms of an ontology that can be exploited by a Fusion Adaptation Module. This module decides how to modify the solution for getting optimal results, based on the desired fusion products, the features of the involved components, and contextual information either acquired or inferred.

In order to make this possible, we had to make our way through the following steps:

- Define a formal system for expressing Problem Space Characterizations: domain, constraints, data/information types, sensors and algorithms.
- Define a formal system for expressing the relevant context information and how it affects the problem.
- Design a procedure that, given a problem specification, determines how to solve it using some available tools. The solution has to be the best possible one, according to some criteria expressed in the Problem Space Characterization.
- Combine the aforementioned elements in a generic framework for developing adaptive sensor fusion problems

The developed solution is applied to a previously explored scenario [3] of GNSS/INS fusion for car navigation in mixed urban and open road environments. The car is equipped with mid-end inertial device (accelerometer, gyroscope, magnetometer) and GPS receiver with differential capabilities. In order to make the problem more interesting and increase the robustness of the final solution, the system can use the same sensors (inertial and GPS) in a smartphone that is placed inside the vehicle.

We show that the proposed system can determine and configure a suitable fusion solution under dynamic changes in sensor availability, desired products and system context, trying to optimize quality indicators as energy consumption or accuracy of the solution. The context of the system includes the battery status of the smartphone, if it is being used in that moment, and the type of environment in which the vehicle is moving.

II. PREVIOUS WORK

This section reviews the existing literature for works related with our proposal. Since this proposal comprehends different areas, we have split the review in subsections.

A. Problem-space and Context description

We have identified ontologies as the right tool for describing problem space and relevant context information. Ontologies have a good number of benefits when used for knowledge representation [4]: they allow to describe concepts, domain assumptions and constraints, and it is possible to apply automatic logical inference processes over them. An important application for ontologies is to define a common vocabulary that ensures some degree of interoperability between automated systems. Another advantage is that ontologies can be easily interpreted by both humans and computer.

The paper [5] describes how the System Entity Structure (SES) ontology framework can be used to improve how information is exchanged. The idea is to enable centralized Data Fusion processes that acquire information from networked environments. It is used to transform raw data to different (high-level) representations that satisfy the needs of the various layers in a Data Fusion system.

Dockhorn-Costa describes in her thesis [6] the concepts and fundamental structures for supporting the development of Context-aware applications. It contains a very detailed and solid work on modeling, representing and using context information. It defends the use of foundational ontologies for "representing conceptualizations that are truthful to reality". These and other works as [7] contain nice and clear examples on using ontologies to describe context information and components of computational systems. Our proposal takes some of their ideas and principles.

In the article [8], a context-aware fusion system for harbor surveillance is presented. Authors propose using OWL ontologies to describe the elements of the problem (vessels and their characteristics/features, navigation channels and others), and defines a set of rules for detecting anomalous behaviors using SWRL (Semantic Web Rule Language). Pellet inference engine [9] applies those rules to the entities in the ontology to detect vessels breaking navigation rules. The use of ontologies presents some drawbacks. Knowledge modeling and exploitation requires a higher effort compared with other tools as rule-based systems that would suffice for a simple system as the case of study we presented here. A rulebased approach provides additional advantages as clarifying the reasoning process in charge of adapting the system.

An important remark about this work, is that it uses ontologies for representation purposes but not for inference or reasoning. The reasoning/inference processes read information from the ontology, but use a custom process that combines a rule-based system with a search algorithm. In spite of the higher design and development cost, we encourage the use of ontologies for representing knowledge when it is important to reuse knowledge, or to enhance interoperability between collaborating systems. We find this last case in the proposed scenario, where the smartphone contributes with its sensors to the fusion process but it is not guaranteed to be part of the system.

B. Context-based adaptation

We found in [10] a proposal resembling ours, but more generically defined and not restricted to sensor fusion. It defines a generic Information Fusion Framework strongly focused in the creation of adaptive fusion processes. These adaptive processes represent the 4th level of the JDL model and can affect every part of the fusion application, including algorithms internal parameterization, the interaction between them, and sensor/resource management. The components in charge of doing the adaptation are spread over the whole architecture, and receive the name of "adaptive logic" blocks. They combine information about current system performance and relevant contextual data. Our work is restricted to sensor fusion, which includes only levels 0 and 1 of the JDL model, and is expected to deal with information having a lower abstraction level, both for fusion and context.

In [11], authors discuss the requirements of a middleware for context-aware applications with ubiquity features. We are not interested in context-aware applications and do not pursue ubiquity, but the paper presents some thoughts of interest:

- Support for context evolution: the middleware must support the inclusion of new context types/concepts without affecting the execution of consumer processes.
- Extensible abstractions for accessing and using knowledge: the middleware should allow access to context information through mechanisms that are adequate for the level of abstraction of the target applications. It should allow the specification of new abstractions in top of the existing base of knowledge.
- Architectural independence: related with permitting access to context information from different platforms (hardware or software).
- Decoupling between context management and inference mechanisms: authors argue that the mechanisms for context inference must be decoupled from context management infrastructures, because it results in a good trade-off of expressiveness, consistency, computational efficiency and reusability.

They also identify ontologies as the appropriate tool for modeling context, and the middleware approach as the more suitable mechanism for decoupling context management and usage.

C. Automatic sensor and algorithm selection

Some authors claim that most sensor fusion works need to use all the available sensors all the time. The reason is that we can expect superior performance from fusion systems tailored for a specific problem, especially when they exploit domain knowledge through the subtle relations between sensors or problem variables. This results, however, in highly coupled solutions that are less robust against (a) sensor failure/outage, (b) sensors showing unexpected low performance (c) external conditions invalidating prior knowledge of the domain.

The article [12] presents a rule-based framework that selects the most reliable sensors and most suitable algorithm

for fusing sensor data in a mobile robot platform. The framework does not require any preliminary knowledge about the sensors involved, although the presented solution is limited to sensors whose measures can be translated to grid occupancy maps, such as the cameras and ultrasonic sensors used in the experiment. That simplification provides a homogeneous view of the sensory information, making possible to calculate comparable quality metrics.

III. PROPOSAL

This sections contains a detailed description of our proposal. It begins with an overview of the whole architecture (see Fig.1), which is a complete rework over the fundamental concepts explored in the previous works [13], [14] and then details the adaptation-related components.

A. Architectural components

The proposed architecture is based on two important concepts: virtual sensors and widgets.

Information sources are abstracted as "virtual sensors" [15], that provide data through an homogeneous interface, hiding the specific mechanisms through which it is produced. We find an example in the Android OS API, that defines sensors for counting steps or provide the gravity direction vector. These virtual sensors can be backed by hardware implementation (manufacturers are encouraged to implement step counting on hardware to save battery) or be derived/composed by other information, but this fact is transparent to the developer.

The components of the fusion system are abstracted as Widgets [16]. A widget can be seen as a reusable building block that encapsulates a functionality. It exposes a well defined interface that can be used to control it, feed the required input information and extract the produced outputs. Widget-based architectures rely on a centralized control component that acts as repository of available widgets and existing links between components. In our case, the central component is the Fusion Adaptation Module, described later in this section.

These abstractions are fundamental in our proposal for two reasons: they make possible to provide a formal and normalized description of the fusion system components, and simplify the automatic creation of valid sensor fusion schemes. Both aspects are explored in the next subsections.

B. Problem-space description

The proposed ontology describes and relates the different pieces of problem, this is, data elements (inputs, outputs, context information), sensors and algorithms. It is defined using OWL (Ontology Web Language) [17], which is constructed in top of RDFS (Resource Description Framework Schema, a more basic language for describing ontologies). Both RDFS and OWL can be serialized using the XML representation of RDF language [18]. We have chosen XML-RDF as vehicular language because it is accepted by the majority of ontology edition and visualization tools, and is easy to process with Jena library, written in Java language and part of the Apache tools.

OWL and RDFS are defined as "a language for describing vocabularies". We consider a vocabulary as a set of descriptions of classes and properties. The ontology is populated with individuals, that are elements defined and interrelated according to the rules of this vocabulary. A RDF ontology starts describing a hierarchy of classes. A class is a category for individuals, quite similar to OOP (Object Oriented Programming) classes stripped from its behavior (only data). Individuals can belong to several classes at the same time.

The other element of a vocabulary are properties. We are using two types of properties in this work: object properties and data properties. An object property relates two individuals. For example: our ontology defines the "produces" property, whose domain is DataNode and ranges over DataProduct. It can be used to express that a gps sensor produces a latitude-longitude fix. Data properties relate individuals with literal values, and can be seen as the attributes/fields of classes in OOP languages. An example is "energyConsumption" property: its domain are DataNode individuals, and its value is the enumeration "High", "Medium", "Low", "None" (it could be represented as a real number, for example the consumption in mW).



Fig. 2. Hierarchy of classes in the problem-space description ontology

Classes: Fig.2 shows the hierarchy of classes used for this work. The top elements are the most important concepts, while the subcategories are defined for further inference processes. Let us describe the top classes:

• Data Nodes are the basic building blocks of a fusion system. They have a direct correspondence with the Widgets that produce and/or consume the information. The presented figure show several subcategories: Sensors –both virtual and real–, Data Conversion functions, Sensor Corrective actions and Fusion Algorithms. These subcategories can help defining constraints or inference processes, as "the roots of a fusion process have to be Sensors" or "a Fusion Process graph cannot contain



Fig. 1. Architecture of the context-based adaptive sensor fusion system

loops, except when the loop involves a sensor corrective action".

- Data Types categorize the type of information managed by the system. The figure show a further decomposition in Numeric, Image, Map, Symbolic and Text classes. Applications demanding additional detail can use existing ontologies as NASA QUDT (Quantities, Units, Dimensions and Types) ontology (for more details check [19]).
- Data Product are auxiliary entities for representing attributed data production or consumption. RDF ontologies can express through an Object Property that a Data Node "produces" information of a certain Data Type. However, this relation does not allow features as "Sensor Y produces data type Z at 5Hz with high accuracy". Data Product is an instrumental entity that can be placed between both classes and be attributed with data properties that express the desired features. Thus, the above statement can be reflected as the composition of "Sensor Y produces Data Product Z", "Data Product Z has Data Type Y", "Data Product Z update rate is 5Hz" and "Data Product Z quality is high".

Object Properties in Fig.3 have been classified according to their domain class: properties of data nodes and properties of data products. The first category includes the relations "produces" and "consumes". They allow to express which Data Products are the inputs and outputs of Data Nodes. The

property "preconditionedTo" is intended to describe arbitrary requirements for a Data Node being applicable, such as a certain Data Node being active.

The second category, properties of data products, contains the property "hasType" that relates a data product with a data type. It also the inverse relations "producedBy" and "consumedBy", in this work, by

 topObjectProperty data-node-property consumes
energyConsumption
preconditioned i o
produces
Television and the second s
consumedBy
hasType
isBiasCorrected
isFiltered
producedBy
quality
updateFrequencyInHz

Fig. 3. Object properties in the problem-space description ontology

Data Properties, in Fig. 4, allow to add literal values to individuals. Following the example of Object Properties, We have defined three categories depending on the domain class.

data-type-property contains two data properties: "cardinality", describing the size (number of elements) of a data type individual, and "basic_type", that can be used to indicate the type of its underlying data elements.



Fig. 4. Data properties in the problem-space description ontology

This ontology can be further developed to include new features for sensors and algorithms, and support more complex reasoning processes.

C. Context management module

Context management module is in charge of keeping an updated record of the context relevant to the fusion system. It is also in charge of satisfying the needs of contextual information of the different components of the fusion system, with the required level of abstraction and through adequate abstract channels.

Developers are encouraged to describe the relevant context knowledge through an ontology, to foster reusability and improve information exchange between systems. However, it is very difficult to reach a unique representation of contextual information that is suitable for all the problems. Thus, we do not propose a set of classes and properties, as we did with the problem-space description. A concrete example is included in the IV section.

We have defined two channels for accessing the context information maintained by this module. The first one is through virtual sensors linked to the context variable of interest. These virtual sensors can be integrated into the fusion solution by the fusion adaptation module, just as any other data node. The second one is a subscription service that notifies updates in context variables to the interested entities. In our case, the Fusion Adaptation Module can integrate these updates into its inference processes to improve the selected fusion solution.

D. Fusion adaptation module

For this work, we implemented a simple Fusion Adaptation module that loads the specification of information sources and data processing nodes, and generates a valid fusion solution that provides the required outputs.

Back to Fig. 1, the Fusion Adaptation module contains three elements: the problem space description, the repository of solution elements and the inference process. The repository of elements is a software library (or set of libraries) that provides fusion algorithms and other tools following the aforementioned widget style. The proposed solution does not impose restrictions on the real implementation of these components, which can be in-place software or just a wrapper over remote services. Each library has a companion ontology detailing its contents, from the mandatory basic aspects as inputs/outputs to extended features as quality, constraints, or requirements. Problem space description is composed by the union of the sensor set description and these companion ontologies. Describing the problem space through ontologies has an additional advantage: sensors and algorithms can be incorporated online to the adaption process.

The last element in the figure is the Inference process, in charge of combining the available solution elements to create the sensor fusion system expected to deliver the best performance in the present conditions. It selects which sensors and algorithms are used, and how they are connected. The optimal implementation depends on the domain of application and the considered factors, e.g. contextual information and solution quality indicators. For this work, we have chosen an event-triggered search process, where the terms of the search are affected by contextual factors. The process is as following:

a) Event processing: Fusion adaptation module receives an event. We have defined the following events: (a) Some sensor is no longer available (b) A new sensor is available (c) Some context variable has changed (d) The list of desired fusion products has changed.

b) Determine reach of the event: The inference process determines how the received event affects the elements of the solution. The effect can be direct, e.g. if a sensor is down, the equivalent DataNode has to be marked as not available so that it is not used in a solution, or indirect, e.g. in the selected case of use, some fusion algorithms cannot be used if the vehicle is moving underground. For more details, see table II.

This information is ideally described as constraints or rules in the different ontologies used to describe the problem. An inference process can be used for generic constraint reasoning. Previously in this paper, we referenced the work [8], which defines rules using SWRL to reason directly over the domain ontology. Ontology-based reasoning is, however, a computationally expensive choice. We overcome this problem using the Drools rule-based system [20]: this library, written in Java, provides a fast inference engine (implements the RETE algorithm). Rules can be defined in text files that can be loaded dynamically, and the inference engine can use Java objects of the target application as facts for the knowledge base. The implemented system is equivalent to using SWRL and Pellet for ontology reasoning. For more details, check section IV.

c) Compose sensor fusion solution: Once the elements of the solution have been modified according to the events, a new sensor fusion solution is composed. For this work, we chose to restrict valid solutions to a tree: a directed graph with no loops. The list of desired fusion products is fed into the system as a data node that consumes data without producing any output. It is the root of the tree.

Solutions are composed through back-chaining, guided by a search algorithm that follows a depth-first strategy. The leaves of the tree will be sensors (pure information sources), that produce data without requiring any input. In many cases, there are several valid solutions for a given set of conditions. Our implementation determines which is the best solution using a set of rules which determine the suitability of each solution under different contexts. It comprises basic checks such as avoiding self-connections and loops, as well as more advanced criteria as maximization/minimization of numeric indicators –energy consumption, accuracy score. For more details, check section IV

IV. CASE OF USE

We have selected a mixed open and urban ground navigation scenario. The fusion system is initially composed by an on-board MEMS IMU (gyroscope and accelerometer) and a differential GPS receiver. At some point in time, it is augmented with a smartphone that also includes IMU and GPS, and some other sensors and sources of information. The result is a redundant, heterogeneous sensor set. This scenario presents several difficulties:

- Fusion requirements change over time.
- Sensor availability change over time.
- Sensors performance is uneven. For example, GPS signal is subject to degradation and outages in urban navigation, and the smartphone is not guaranteed to keep its orientation with respect to car body –this makes harder the interpretation of IMU readings.

Part of this information is explicit (fusion requirements, sensor availability), but some other has to be inferred, such as when the vehicle is in urban environment.

A. Problem-space model

The problem space is described creating individuals in the ontology. These individuals have to represent the sensor set, the involved data types and the features/requirements of the consumed/produced data.

We created a total of 8 sensors (Data Nodes), 5 Data Types and 8 data products to represent the two available sets of sensors. Table I summarizes the data introduced in the ontology.

TABLE I Available sensors

On-board	Product	Quality	Freq.	Bias corrected
Gyroscope	angular rate	medium	100 Hz	No
Accelerometer	acceleration	medium	100 Hz	No
CPS	Lat-Lon	medium	5 Hz	
015	Altitude	medium	5 Hz	
Diff CPS	Lat-Lon	high	5Hz	
Dill-GF5	Altitude	high	$5H_7$	
	7 minude	mgn	JIIL	
Smartphone	Product	Quality	Freq.	Bias corrected
Smartphone Gyroscope	Product angular rate	Quality low	Freq.	Bias corrected
Smartphone Gyroscope Accelerometer	Product angular rate acceleration	Quality low low	Freq.	Bias corrected
Smartphone Gyroscope Accelerometer	Product angular rate acceleration Lat-Lon	Quality low low low	Freq.	Bias corrected
Smartphone Gyroscope Accelerometer GPS	Product angular rate acceleration Lat-Lon Altitude	Quality low low low low	Freq. 1 Hz 1 Hz	Bias corrected

The repository of algorithms include:

• 1 Data Conversion algorithm: transforms GPS fixes (latitude-longitude using the WGS84 geoid model) to local cartesian coordinates referred to a given origin.

- 3 Fusion Algorithms:
 - a Kalman Filter that uses position measures to estimate position and speed of the vehicle.
 - an Unscented Kalman Filter that calculates the attitude (orientation) from the position and the angular rates of the vehicle.
 - a Unscented Kalman Filter using the position, angular rate and acceleration of the vehicle to estimate its full kinematics.
- 1 Sensor Corrective Action: estimates and compensates the bias of the gyroscope. It requires knowing when the vehicle is stopped, and the raw (biased) angular rates.
- 2 Virtual Sensors for context data: a stop detector and a turn detector. The availability of these sensors is determined automatically from the Context description ontology (see next section). There is a potential virtual sensor for each context variable (i.e. smartphone placement and energy policy), but they have not been integrated in the solution search process because they are not used as input to any fusion algorithm.

B. Context model

We chose a simple context for this application, consisting on four different pieces of information.

- Vehicle environment (inferred): open road, urban or underground.
- Vehicle motion conditions (inferred): vehicle stopped, vehicle turning.
- Smartphone energy policy (explicit): critical, low, normal or plugged
- Smartphone placement (inferred): resting or on user hand

These elements are defined in a OWL ontology, as shown in Fig.5.



Fig. 5. Context description ontology for the ground vehicle navigation experiment

Regarding the acquisition of context information, vehicle environment is automatically inferred using the availability and quality of GPS measures and the speed of the vehicle and the motion conditions of the vehicle are calculated from the inertial readings. In future experiments motion conditions could be explicitly acquired from the ODB-II port of the vehicle, but this will not affect components of the fusion system consuming this context information thanks to the Context management module and the virtual sensor abstraction.

Smartphone placement is determined using accelerometer readings, that change direction of the gravity vector and are significantly noisier when a user takes the device. The context management module acquires context information from sensors and from Data Nodes designed to calculate or infer these features. In our case, context acquisition processes are fixed (not adaptive), so we simply add the required inputs as desired fusion outputs, and let the Fusion Adaptation module calculate a valid generation scheme.

Part of the domain logic is referred to how context affects the solution of the fusion problem. In our case, we have identified some contextual constraints on sensor usability and the applicability of some fusion algorithms. More specifically, we will include the following constraints in the reasoning processes of the Fusion Adaptation module:

- If smartphone energy policy is critical, its sensors must not be used.
- If the smartphone is not resting in a surface, the accelerometer and gyroscope readings cannot be used to determine vehicle motion.
- If environment is urban, solutions not using GPS are preferred. Filtering solutions based on GPS alone are discarded.

Since the Fusion adaptation process replicates relevant ontology information in a Java application, we can use Drools to mimic the capabilities of an ontology-based reasoning system directly over the inference process of the implemented Fusion Adaptation module. The rules used in this case, shown in II, have direct translation between SWRL and Drools languages.

TABLE II Contextual constraints on sensor set and algorithms

Sensor	Conditional	Context
smartphone (any)	INCOMPATIBLE	EnergyPolicy == LOW
" (any)	INCOMPATIBLE	EnergyPolicy == CRITICAL
" accel.	REQUIRES	Placement == RESTING
" gyro	REQUIRES	Placement == RESTING
Algorithm	Conditional	Context
KFKinematic	INCOMPATIBLE	Environment == UNDERGR.
UKFAttitude	INCOMPATIBLE	Environment == UNDERGR.

C. Fusion adaptation

We have tested the system for a mixed urban and open road trajectory. The Fusion Adaptation module is notified of relevant changes in the system:

- Addition and removal of sensors: smartphone is available only from t=100s to t=300s.
- Changes in relevant context: the trajectory starts in urban environment, switches to open road around t=150s and goes back to urban close to the end. The battery status of the smartphone starts in "plugged", and switches to "critical" at t=200s.
- Changes in the list of required fusion products. We start asking for position and linear speed, change to only position but with high accuracy during the open road fragment. At the end, we add turn detection to the list of desired products.

The adaptation module calculates a new solution right after detecting each change. We include two sample solutions reached by the system. Fig.6 is a solution around t=120s that returns the vehicle position and its linear speed, with smartphone sensors available and moving in a strict urban environment with poor GPS signal. The system chose to take the position from a Kalman Filter, and use the Unscented Kalman filter to extract the speed.



Fig. 6. Sample fusion solution for getting vehicle position and linear speed

The second solution, shown in Fig.7, describes a solution in open road environment where the system is asked to produce a high-accuracy position and stop detection, with smartphone in critical battery status. In this case, the solution includes the simple kinematic Kalman Filter using the available differential GPS readings. The stop detection module, fed by the onboard sensors (smartphone is not available due to battery status), is also connected to the output.



Fig. 7. Sample fusion solution for getting high-accuracy location and stop detection

V. CONCLUSIONS

This paper presents a generic framework for creating multi-sensor fusion applications. The framework is composed by an ontology for describing sensor fusion problems and the elements available for solving them, a generic architecture for context-aware sensor fusion, and an inference module that determines the best solution for a given problem in the considered context.

The capabilities of the proposed solution include:

- inclusion of arbitrary sensors, sources of information and processing algorithms, as long as they are described in terms of the proposed ontology.
- online addition and removal of sensors and processing algorithms (while the fusion process is running).

- maintenance of a context information repository accessible by any component of the fusion process.
- automatic determination of the optimal fusion solution for a given list of required fusion products. The solution takes into account the relevant context of the system and can incorporate arbitrary criteria for determining the suitability of the solutions.

The correct application of the framework has been shown to create fusion systems that are robust against sensor failures and external conditions affecting the performance of a sensor or a particular processing scheme. Context information is a fundamental aspect of this adaptability. The proposed design principles do also improve the reusability of the implemented software, and make easier to augment a existing system with new algorithms, sensors and capabilities.

A real experiment is briefly described, that illustrates the usage of the framework, some of its capabilities and the obtained results.

ACKNOWLEDGMENTS

This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485).

REFERENCES

- J. Wang and Y. Gao, "The aiding of MEMS INS/GPS integration using artificial intelligence for land vehicle navigation," *IAENG International Journal of Computer Science*, Jan. 2007.
- [2] F. Caron, E. Duflos, D. Pomorski, and P. Van-"GPS/IMU data fusion using multisensor Kalman heeghe, introduction of contextual aspects," filtering: Information Fusion, vol. 7, no. 2, pp. 221-230, Jun. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1220959.1221050 http://linkinghub.elsevier.com/retrieve/pii/S156625350400065X
- [3] E. D. Martí, D. Martín, J. García, A. de la Escalera, J. M. Molina, and J. M. Armingol, "Context-aided sensor fusion for enhanced urban navigation," *Sensors (Switzerland)*, vol. 12, no. 12, pp. 16 802– 16 837, Jan. 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/12/16802
- [4] T. Bürger and E. Simperl, "Measuring the Benefits of Ontologies," in On the Move to Meaningful Internet Systems: OTM 2008 Workshops, ser. Lecture Notes in Computer Science, R. Meersman, Z. Tari, and P. Herrero, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 5333, pp. 584–594. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-540-88875-8
- [5] H. Lee and B. P. Zeigler, "System Entity Structure Ontological Data Fusion Process Integrated with C2 Systems," *The Journal* of Defense Modeling and Simulation: Applications, Methodology, Technology, vol. 7, no. 4, pp. 206–225, Apr. 2010. [Online]. Available: http://dms.sagepub.com/content/7/4/206.abstract
- [6] P. D. Costa, "Architectural Support for Context-Aware Applications," Ph.D. dissertation, Telematica Instituut, 2007.

- [7] H. Wu, "Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory," Ph.D. dissertation, Carnegie Mellon University, 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.3600
- [8] J. Gómez-Romero, M. A. Serrano, J. García, J. M. Molina, and G. Rogova, "Context-based multi-level information fusion for harbor surveillance," *Information Fusion*, vol. 21, pp. 173–186, Jan. 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2657516.2657961
- [9] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570826807000169
- [10] J. Llinas, "A Survey and Analysis of Frameworks and Framework Issues for Information Fusion Applications," in *Hybrid Artificial Intelligence Systems*, M. Graña Romay, Manuel and Corchado, Emilio and Garcia Sebastian, Ed. Springer Berlin Heidelberg, 2010, pp. 14–23. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13769-3_2
- [11] R. C. A. da Rocha, M. Endler, and T. S. de Siqueira, "Middleware for ubiquitous context-awareness," in *Proceedings* of the 6th international workshop on Middleware for pervasive and ad-hoc computing - MPAC '08. New York, New York, USA: ACM Press, Dec. 2008, pp. 43–48. [Online]. Available: http://dl.acm.org/citation.cfm?id=1462789.1462797
- [12] O. Cohen and Y. Edan, "A sensor fusion framework for online sensor and algorithm selection," *Robotics and Autonomous Systems*, vol. 56, no. 9, pp. 762–776, Sep. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092188900700187X
- [13] E. Martí, J. García, and J. M. Molina, "Context-awareness at the service of sensor fusion systems: Inverting the usual scheme," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6692 LNCS, 2011, pp. 653–660.
- [14] —, "Opportunistic multi-sensor fusion for robust navigation in smart environments," in *Advances in Intelligent and Soft Computing*, vol. 94. Springer, 2011, pp. 59–68.
- [15] J. Indulska and P. Sutton, "Location management in pervasive systems," pp. 143–151, Jan. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=827987.828003
- [16] D. Salber, A. K. Dey, and G. D. Abowd, "The context toolkit: aiding the development of context-enabled applications," in *Proceedings of* the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99. New York, New York, USA: ACM Press, May 1999, pp. 434–441. [Online]. Available: http://dl.acm.org/citation.cfm?id=302979.303126
- [17] L. W. Lacy, Owl: Representing Information Using the Web Ontology Language. Trafford Publishing, 2005. [Online]. Available: http://books.google.com/books?hl=es&lr=&id=71sGYlby2uwC&pgis=1
- [18] D. Brickley and R. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema," 2004. [Online]. Available: http://www.w3.org/TR/rdfschema/.
- [19] H. Rijgersberg, M. van Assem, and J. Top, "Ontology of units of measure and related concepts," *Semantic Web*, vol. 4, no. 1, pp. 3–13, Jan. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2595053.2595055
- [20] E. de Ley and D. Jacobs, "Rules-based analysis with JBoss Drools: adding intelligence to automation," in *Proceedings of ICALEPCS 2011*, 2011. [Online]. Available: http://inis.iaea.org/Search/search.aspx?orig_q=RN:43107173