Exploiting Structural Knowledge using Network Description Language and Causal Models for Fault Diagnosis in Wireless Sensor Networks

Álvaro Carrera, Carlos A. Iglesias
Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Madrid, Spain
a.carrera@dit.upm.es, cif@dit.upm.es

ABSTRACT: *Fault Diagnosis is an essential management task for any telecommunication network and it is even more crucial for Wireless Sensor Networks due to their dynamic nature. Based on Agent Technology, this paper presents an architecture that combines different network and diagnosis models to carry out a Fault Diagnosis process: a Causal Model to relate fault root causes with their symptoms and a Structural Model to define the network and its properties. The behaviour of the proposed agent is driven by an Expertise Model which handles the different phases, tasks and problems which can be found in any diagnosis process. The proposed approach has been evaluated in a simulated environment with emulated MICAz devices for a motion detection application.*

Keywords: Wireless Sensor Networks, Fault Diagnosis, Agent Technology

Received: 17 July 2015, Revised 24 August 2015, Accepted 29 August 2015

1. Introduction

Wireless Sensor Networks (WSNs) have emerged as a paradigm that makes the network highly dependent on the environment where it is deployed [1]. Due to the features of this type of network, such as low-cost [5] or multifunctional sensor capacity [12], WSNs have been applied in many different fields increasing the demand of efficient and flexible solutions for different management tasks [9], such as energy management [24] or fault management [26].

As in any telecommunication network, Fault Diagnosis is a key task in a WSN. Actually, the features of such networks make this task even more crucial than in other types of networks [20]. For instance, that are error-prone means they have unsatisfactory reliability for some applications [13]. Thus, a correct real-time fault management would improve its reliability and further spread its use in the industry.

Generally, two different approaches are used for Fault Diagnosis in WSN: proactive and passive. On one hand, the proactive approach uses extra computational resources in low capacity devices, but it offers high-detailed information about the whole network, such as energy status [27] or neighbour lists [19]. On the other hand, the passive approach collects those data with less intrusive methods which are decisive for low-energy consumption systems, such as packet marking strategy [10] or packet sniffer approach [4].

This paper proposes a flexible agent architecture that blends both approaches, proactive and passive, which enables to increase or to decrease the work load in different critical nodes of the dynamic topology of WSNs using agent technology. To evaluate the proposed approach, a simulation environment is presented. In the simulation, a WSN for motion detection is running and Fault Diagnosis tasks are carried out in real time by a Fault Diagnosis Agent that is running in the sink node of the network.

The reminder sections of this paper are structured as follows. Firstly, Section 2 describes the different models used by the proposed architecture to carry out the diagnosis tasks in the WSN. Secondly, Section 3 presents the proposed Fault Diagnosis Agent Architecture which combines different reasoning techniques to manage the uncertainty inherent in the diagnosis process of a dynamic network. Thirdly, Section 4 shows the most important features of the simulated WSN scenario and Section 5 exposes the preliminary results of some experiments. Finally, Section 6 presents some concluding remarks and proposes some future works.

2. Diagnostic Models

Fault Diagnosis is a task that has been carried out traditionally by human operators following different strategies depending on the context. These strategies are presented in Section 2.1 following an Expertise Model. This model only provides different ways to solve the diagnosis task, not the domain knowledge required to carry out the diagnosis process for a telecommunication network, such as a WSN.

In the scope of this work, other models are required to represent that domain knowledge, such as the network structure, the services offered in that network or the possible faults and the symptoms of those faults. Section 2.2 exposes the network diagnostic models used in the proposed architecture.

2.1 Expertise Model

This section presents a generic expertise model following the MAS-CommonKADS methodology [6] based on the analysis exposed by Benjamins in [2]. The main elements of an expertise model are tasks, problem-solving methods and primitive inferences.

A task is a specification of a goal that is needed to be achieved. It can be decomposed into subtasks by a Problem- Solving Method (PSM). A PSM is the definition of the way to achieve the goal specified in a task. The achievement of a goal specified in a task can be realised by several methods, each of which consists of subtasks that can be reusable in different PSMs. In the following diagrams, PSM are represented by rectangles and tasks by ellipses.

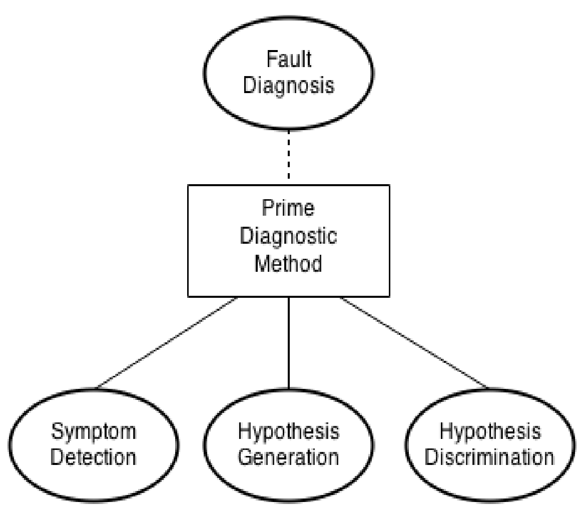

Figure 1. Prime Diagnostic Method

Following this nomenclature, the Fault Diagnosis task is realised by the Prime Diagnostic Method (see Figure 1), which is decomposed into three subtasks: (i) Symptom Detection, finding out whether complaints are indeed symptoms, (ii) Hypothesis Generation, generate possible causes based on the symptoms, and (iii) Hypothesis Discrimination, discriminating between the hypotheses based on additional observations.

As mentioned above, each subtask can be realised by several methods. Hence, the subtasks that compose the Prime Diagnostic Method can be performed in different ways depending of the diagnosis scenario. The Symptom Detection Task (see Figure 2) can be realised by methods that can be automated by a diagnosis system such as Compare Symptom Detection or Classify Symptom Detection, or performed externally by third-part systems or human agents, such as User Symptom Detection.

The Compare Symptom Detection method is divided in two subtasks to generate an expected value of a variable and to compare the value obtained from the environment. After that comparison, if the detected behaviour is not the expected one, we say that a symptom is detected. In contrast, the Classify Symptom Detected method is based on a classification task. The information observed from the environment are filtered by a classification engine that decides if the observations are a symptom or not. Finally, the other method is the manual report from a user that is supposed to be truth. Then, no check or classification process is required.

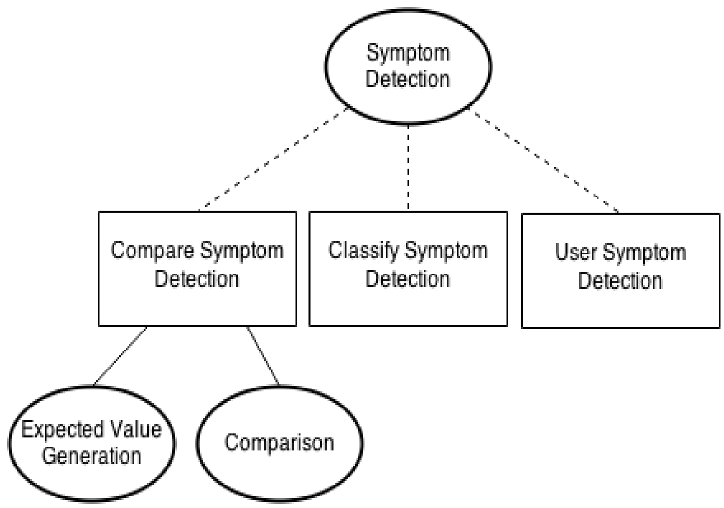

Figure 2. Symptom Detection Task

After a symptom is detected, the Prime Diagnostic Method continues with the next task: Hypothesis Generation (see Figure 3). The two principles options to achieve this goal are the use of a Compiled Method or a Model-based Method. On one hand, the compiled method is composed by three subtasks that uses associations between symptoms and causes and can be followed by a probability filter. This method starts with an abstraction process that translates between raw observations to qualitative and generalised observations. Then, the method continues with the association of that abstracted observations with possible causes of fault. Finally, an optional task is the addition of a probability filter to discard non-probable hypotheses. On the other hand, the model-based method follows other approach based in non-abstracted observations. Firstly, the finding of a set of model entities (such as devices, components or states) that could contribute to an abnormality observation is required. Then, that set is transformed to a hypothesis set in which every element is a possible explanation for the observed symptoms. Finally, a prediction-based filtering can be used to discard inconsistent hypotheses.

Once the hypothesis set is generated, the Hypothesis Discrimination Task (see Figure 4) is performed to find the final conclusion of the Fault Diagnosis Task. The method described by Benjamins in [2] is divided in four subtasks that starts with a decision about the next action to accept or discard one of the possible hypotheses. There are several strategies (or methods) to make that decision driven by different criteria, such as by time restrictions or by computational cost.

Once the strategy is decided, data about possible observations are collected and analysed to find new symptoms or evidences. Finally, the hypothesis set is updated based on the observations collected in the previous task.

As we are focusing the diagnosis task using Agent Technology [25], it is interesting to consider a Multi-Agent System (MAS) in a distributed environment. Thus, we propose the use of two different methods: Individual Reasoning, that is the classical approach exposed in [2], and Social Reasoning, that is a distributed way to reason about the possible hypotheses of fault causes discussing and/or sharing partial information during a diagnosis task.

If a reliable conclusion is found in the hypotheses Update task, the Fault Diagnosis task can be considered as finished. If not, the discrimination process continues repeating the previous tasks (Hypothesis Selection, Data Collection and Symptom Detection) until a reliable conclusions is achieved.

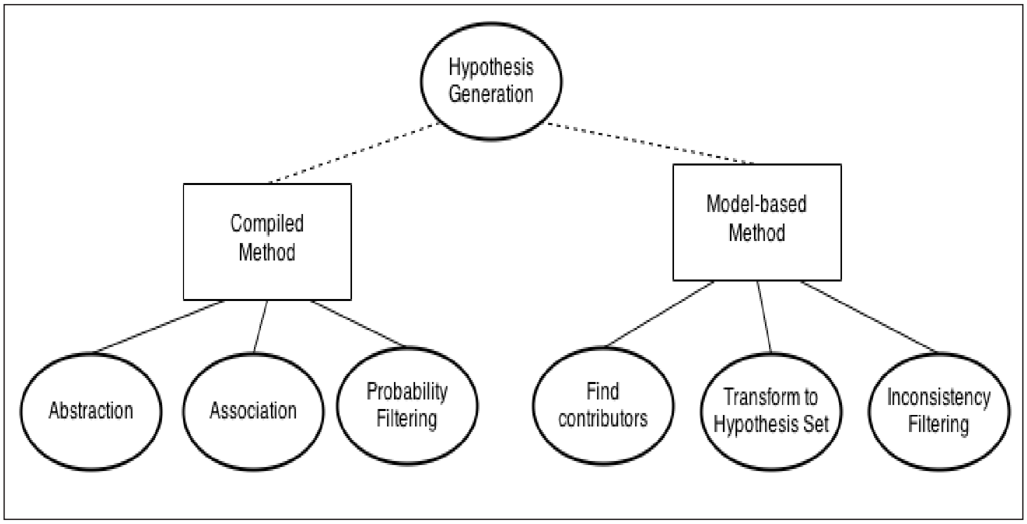

Figure 3. Hypothesis Generation Task

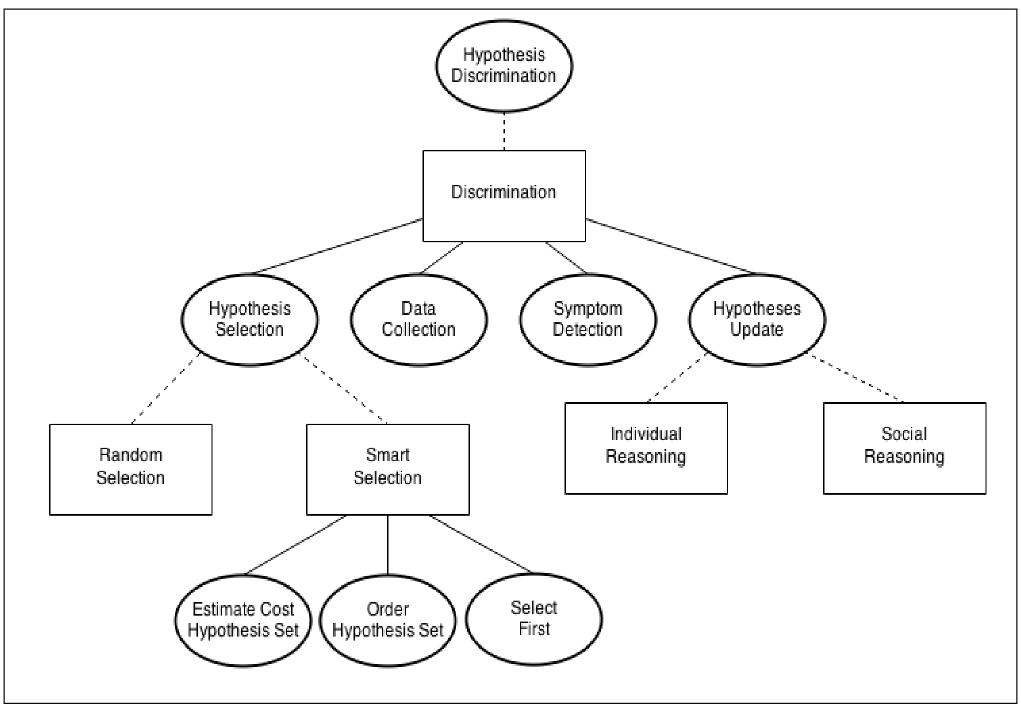

Figure 4. Hypothesis Discrimination Task

Summarising, the expertise model presented above describes a generic Fault Diagnosis process that can be carried out by an agent or a set of agents, both human, software agents or a combination of them. This general expertise model is particularised for the reasoning cycle of proposed Fault Diagnosis Agent Model in Section 3.

2.2 Network Diagnostic Models

The aim of a model is to represent knowledge to be used for a purpose. Thus, we define a Network Diagnostic Model as a representation of some aspects of a network that is used to diagnose fault root causes. For our purpose, we use conceptual models to know and understand key aspects of a telecommunication networks for the fault diagnosis process.

Hence, different aspects of a telecommunication network must be expressed in models to be able to automate fault diagnosis tasks. In this work, two types of models are used: causal models and structural models.

The relation between symptoms and fault root causes is required. A causal model that relates those concepts is used for this purpose. The definition of that causal model is shown in Section 2.2.1. This is an essential model for any diagnosis process, but other views of the network are required to carry out successfully a complete diagnosis task.

The structural model defines the network itself, i.e. its nodes, its links, the services running in the nodes, etc. So, this model not only defines the structure of the network, but also offers information about that nodes and links and the processes that are executing in them. The definition of the structural model is exposed in Section 2.2.2.

2.2.1 Causal Model

A Causal Model (CM) is defined as an abstract model that describes the causal mechanisms of a system. Concretely, Judea Pearl [17] defines a CM as an ordered triple $< U, V, E >$, where U is a set of exogenous variables whose values are determined by factors outside the model; V is a set of endogenous variables whose values are determined by factors within the model; and E is a set of structural equations that express the value of each endogenous variable as a function of the values of the other variables in U and/or V.

Following Pearl's definition, a causal model can be represented as a network with nodes (variables $U \cup V$) and links between those nodes (structural equations E), that is called causal network. A Bayesian Network (BN) [16] can be considered as a causal network if the relations between its variables are causal relations. Formally, a BN is a probabilistic graphic model that represent a set of random variables $\mathcal{V}$ and their conditional dependencies $\mathcal{R}$ via a Directed Acyclic Graph (DAG). Hence, the equivalence between a CM and a BN is given if the variables of the network are both exogenous and endogenous variables ($\mathcal{V} = U \cup V$) and the conditional relations are only causal relations ($\mathcal{R} = E$).

Thus, the causal model used in the proposed Fault Diagnosis Agent Model to determine the relation between symptoms and fault root causes in a telecommunication network is a Bayesian network. Following the Pearl's nomenclature, the fault root causes and the symptoms are the exogenous and endogenous variables respectively. In other words, the symptoms are caused by the faults or the faults are the cause of the symptoms.

Focusing on the structure, a BN is composed by layers of variables that are related with variables in other layers. The simplest models, such as QMR [15] or QMR-DT [21], are composed only by two layers: symptoms and fault root causes, as shown in Figure 5.

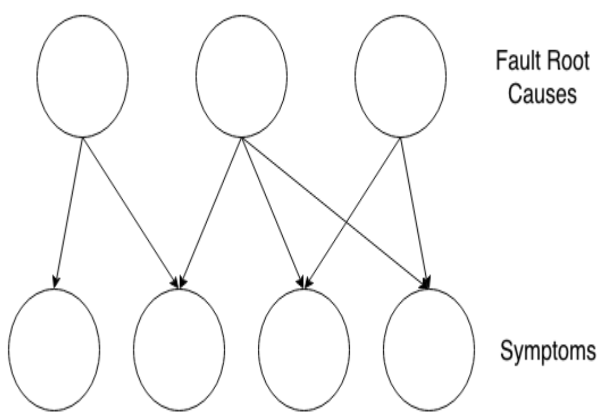

Figure 5. Two layers model of Bayesian network

In contrast, other models, such as the BN3M Model [7], adds context information to the variables set. Thus, a third layer is added to the model, as shown in Figure 6. This three layer structure is simple and its performance is reasonably similar to the two layer models [14]. And the usage of only three types of variables keeps it easy for a human to identify which variables belong to each type. Adding more layer, i.e. more types of variables, will increase the complexity of the model and will not increase the diagnostic performance [18].

Summarising, the models presented above can be used to build causal models that use evidences observed from the environment (in our case, from the telecommunication network) to discriminate between the possible fault root causes. While two layer models use only symptoms and fault root causes (and the relations among them) to form the structure of the network, three layer models adds context variables to be able to use the same model in different contexts. For our purpose of fault diagnosis in telecommunication networks, we can use both types of model, depending of the information that can be collected from the environment.

2.2.2 Structural Model

A Structural Model (SM) of a telecommunication network is defined as a representation of the components of the network, both devices, connections and services, and their properties. Formally, a SM is a duple $< E, P >$, where E is a set of elements that compose the telecommunication network and P is a set of properties that define the set of elements contained in E and the relation between them. To express this knowledge in an standard language, we purpose to use the Network Description Language (NDL) [23] which its first version is being standardised[1] . NDL provides a way to describe networks in a meaningful way in Resource Description Framework (RDF) format. A set of Ontology Web Language (OWL) ontologies have been defined using NDL in the ORCA Project[2] and they are published under the Eclipse Public License[3] . These ontologies are available under the name of NDL-OWL and they cover a great variety of networks. However, they are only a example of use of the Network Description Language that shows the validity of the language.

Some basics classes of NDL are briefly explained below and shown in Figure 7. The NetworkObject class represents the elements of the network, denominated above as E, with complementary concepts, such as Link, Node or Service. These concepts are extended with others that specifies more information about them in type, such as BidirectionalLink or AdaptationService, or in structure, such as NodeComponent. In contrast, other classes represents element properties, denominated above as P, both endogenous, such as VirtualNode, or exogenous, such as Location.

Summarising, the Network Description Language (NDL) is suitable for our purpose of building a Structural Model (SM) for the telecommunication network. With the expressiveness of NDL, the network can be successfully described to be used during the diagnosis tasks in different aspects. Furthermore, the use of standard formats, such as RDF, allows an easy deployment and update of the model in real-time. This is an important feature to allow the use of the proposed architecture in wireless networks where the devices connect and disconnect depending of their locations.

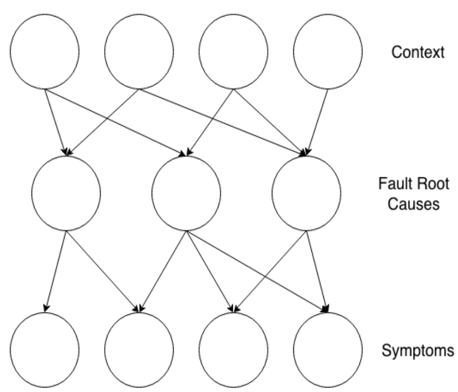

Figure 6. Three layers model of Bayesian network

[1]NML Standardisation Work Group: http://redmine.ogf.org/projects/nml-wg

[2]ORCA Project Web Page: https://geni-orca.renci.org/trac/

[3]Eclipse Public License 1.0: http://www.eclipse.org/legal/epl-v10.html

3. Fault Diagnosis Agent

This section presents an Agent Architecture for Fault Diagnosis in Telecommunication Networks based on the Belief- Desire-Intention (BDI) model [25]. This architecture is based on the classical BDI approach adding the models shown in Section 2. The Reasoning Cycle of the proposed Fault Diagnosis Agent, shown in Figure 8, follows the Expertise Model exposed in Section 2.1. The following subsections shows how the subtasks which compose the Expertise Model are implemented and combined with the Network Diagnostic Models exposed in Section 2.2.

3.1 Symptom Detection

This subtasks is realised analysing the incoming data that agent receives from the telecommunication network. The information obtained from those data are stored in the SM. The updated information is used to reason with the SPARQL Inference Notation (SPIN) rules[4]. These rules can be used to detect anomalies in the expected behaviour of the network and/or to updated the dynamic network features, such as topology, network load or services status. A simple example of SPIN rules used in the case study[5] to discover the path to the sink node is shown in Figure 9.

Figure 7. Main classes of the Network Description Language

3.2 Hypothesis Generation

When any symptom is detected, the next subtasks of the Fault Diagnosis process starts. Based on the symptom and on the information available in the SM, the fault is classified by its typology. Once the fault type is identified, it is included in the Diagnosis Model which handles the diagnosis itself which starts when a symptom is detected and finishes when a conclusion is reached.

This Diagnosis Model manages the information referred to a specific diagnosis case, such as detected symptoms, possible fault root causes, performed tests or hypotheses set. Based on the context (i.e. the information about the network available in the SM), on the fault typology and on the detected symptoms, an hypothesis set is generated using that information as input for the CM.

---

[4]SPIN Web Site: http://spinrdf.org/

[5]Other examples of SPIN rules can be found in our Github public repository: https://github.com/gsi-upm/shanks-wsnmodule/tree/master/src/main/resources/rules

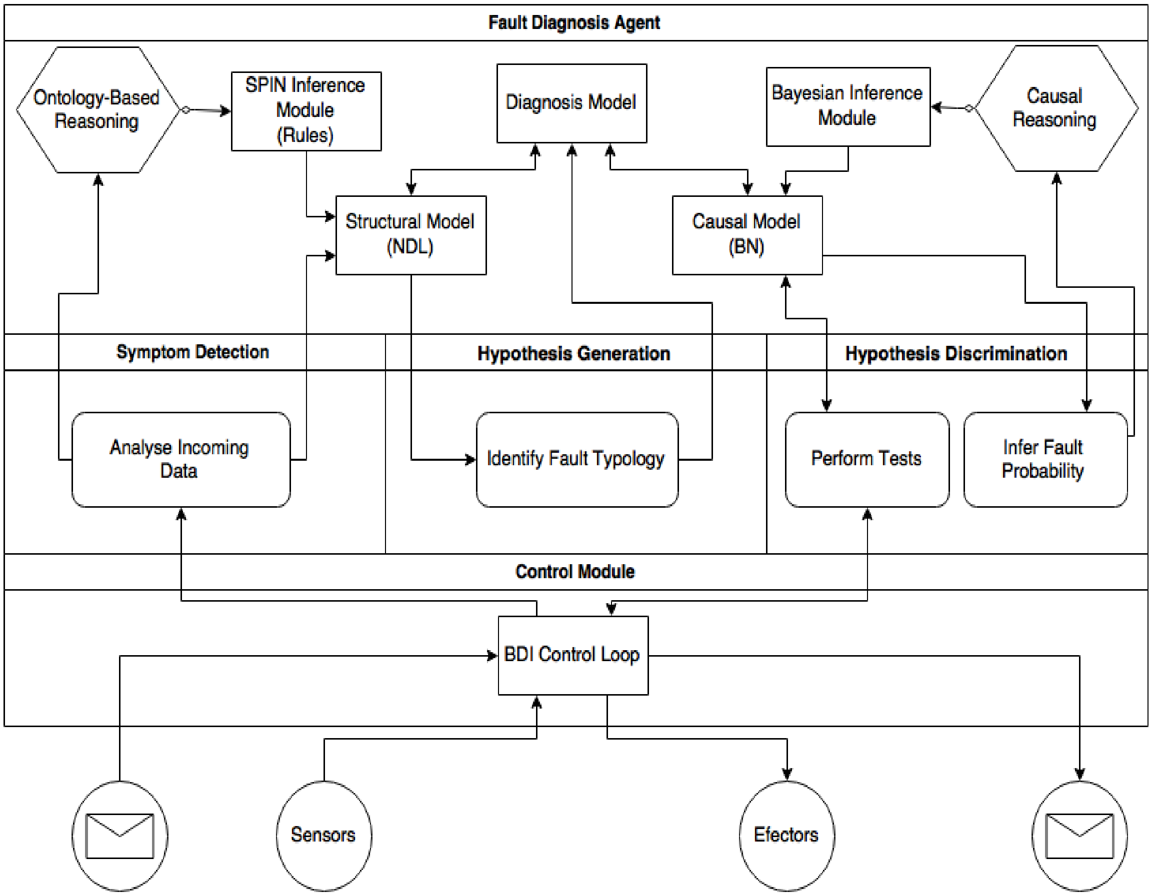

Figure 8. Fault Diagnosis Agent Architecture

```
CONSTRUCT {
        ? p a t h nm l- b a s e : hasNode ? n e x tRo u t e r .
        ? p a t h nml - b a s e : ha sLink ? l i n k .
}
WHERE {
        ? t h i s a wsn - n d l : ZigBe eRout e r .
        ? p a t h a wsn - n d l : P a t hToBa s e S t a t i o n .
        ? p a t h nml - b a s e : hasNode ? t h i s .
        ? t h i s nml - b a s e : i s S o u r c e ? l i n k .
        ? n e x tRo u t e r nml - b a s e : i s S i n k ? l i n k .
}
```

Figure 9. Example of SPIN rule

3.3 Hypothesis Discrimination

After a set of possible fault root causes has been identified in the previous phase, the discrimination phase starts collecting more information about other possible symptoms performing any additional test. That information will be included in the CM and the probabilities of the possible faults are inferred based on the new evidences collected from the network.

This loop of executing tests and updating the hypothesis set continues until no more tests can be executed or an hypothesis reach a confidence threshold. Then, the diagnosis process is finished and the most reliable hypothesis (i.e. the hypothesis with highest probability) is offered as fault root cause of the detected symptoms.

4. Simulation Scenario

This section describes the simulated WSN scenario that has been developed to validate the agent architecture proposed in Section 3. The developed simulation environment is based on MASON simulation framework [11] and it is available in a public Github Repository[6].

The WSN topology is automatically generated[7] with two parameters: number of ZigBee End Devices (ZEDs) and number of ZigBee Routers (ZRs). One ZigBee Coordinator (ZC) is deployed as sink node in the topology and the Fault Diagnosis Agent is executed in this node which we consider without any power or computational restrictions. The ZR nodes route all packages received from the ZEDs, i.e. motion detection alerts, to the sink. All ZED devices connected to the same ZR device conform a cluster, and that ZR is known as cluster head.

Thus, when the simulation starts, the network topology is generated in three steps. Firstly, the ZC node (sink node) is placed in a fixed position and all ZR and ZED nodes are randomly placed. Secondly, ZR and ZED nodes are moved until they have at least one router (or directly the ZC sink node) in range. Finally, Dijkstra's algorithm [22] is used to create routes (i.e. links) between ZR nodes using minimum power consumption criteria based on accurate power consumption values provided by Landsiedel et al. in [8] and ZED nodes are linked to the closest ZR node. An example of the result of this process is illustrated in Figure 10.

The behaviour of the simulated devices have been implemented following the specifications of MICAz device[8] . The emulated MICAz devices are equipped with IR Motion Detection sensors[8] which detect a mobile target. When the target is in range, the detecting nodes generate a message to notify that detection. To reduce the messaging cost of the simulated scenario, no ACK messages are sent to confirm the reception. So, some messages can be lost due to several causes, such as network overflow or weather/noise conditions, as exposed in Section 5. These messages are forwarded including trace information and some data about the nodes in the path. In other words, every node which receives a message adds data about the node itself, such as, node id, message id, cpu load or memory load. Thus, the message received by the sink node contains information about all nodes which have forwarded the original message. That information is used by the Fault Diagnosis Agent to update the structural model and to detect symptoms. The packages between nodes are parsed as plain text and sent using the maximum throughput in ZigBee WSN obtained by Burchfield et al. in [3].

When the WSN is ready to work, the mobile target starts its random movement generating traffic over the network. Then, the Fault Diagnosis Agent that is running in the ZigBee Coordinator node, i.e. the sink node, processes the incoming information about the network status and topology and updates the structural network model in real-time, as shown in Section 3.

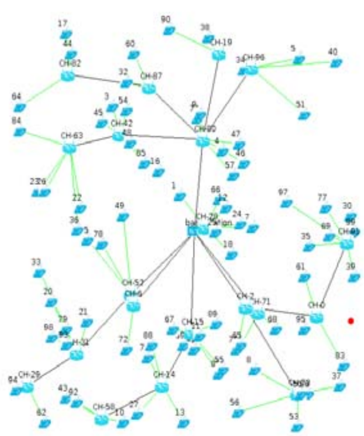

Figure 10. Snapshot of the simulated WSN scenario

[6]WSN Simulation Public Repository: https://github.com/gsi-upm/shanks-wsn-module

[7]We are using the ZigBee Alliance nomenclature: http://www.zigbee.org/

[8]MICAz Datasheet: http://www.memsic.com/wireless-sensor-networks/

[9]IR Motion Detection Sensor Datasheet: https://www3.panasonic.biz/ac/e/control/sensor/human/wl/index.jsp

5. Experimentation

This section exposes the results obtained executing 200 different simulated scenarios with different parameters. We simulate a WSN where nodes are deployed on a two-dimensional space of 100 square meters during 10 minutes[10] with one mobile target with 5 kilometres per hour speed that moves randomly in the simulation space. The perception range of the motion detectors is 5 meters. We consider the simulation is indoor for radio range of MICAz devices. Under these conditions, we design our experiments to quantify the behaviour of the Fault Diagnosis Agent that is running in the sink node. Specifically, we have measured the number of lost messages due to network overflow are detected[11]. In the current version of the simulation environment, we have focused on the symptom detection task (see Section 2.1) using the trace information contained in the messages, as shown in Section 4.

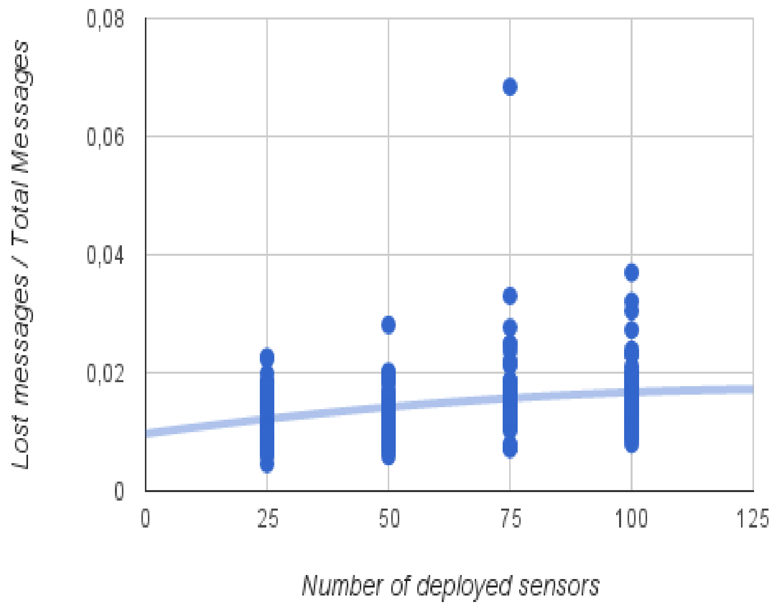

Figure 11. Ratio of lost messages by number of deployed sensors

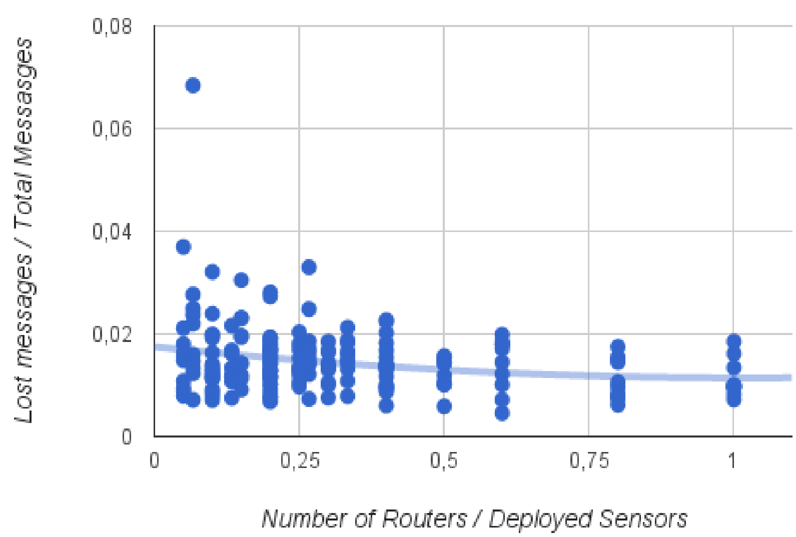

Figure 12. Ratio of lost messages by ratio of number of routers and number of deployed sensors

The results of the simulations are analysed comparing the ratio of detected lost messages and some metrics of the WSN topology. Figure 11 shows the ratio of lost messages by the number of sensors deployed in the simulation. A gradual increase of the ratio can be observed with the number of deployed sensors.

Other interesting relation is the ratio of lost messages compared with the ratio of routers by total sensors, shown in Figure 12. A decrease is detected when the number of routers is similar to the number of deployed sensors, i.e. there are small sensor clusters in the network topology.

Finally, we analyse the lost message ratio with the ratio of edge routers, i.e. the first router in a path to the sink node. This is a metric of the complexity of the trunk network if we build the WSN topology as a tree. Figure 13 shows a similar behaviour independently of the ratio of edge routers in the network.

These initial results show the potential of the simulation environment to continue exploring this WSN scenario in further extensions of the developed simulation framework. That extended scenario will be used to evaluate the remaining phases of the proposed Agent Architecture shown in Section 3: Hypothesis Generation and Hypothesis Discrimination.

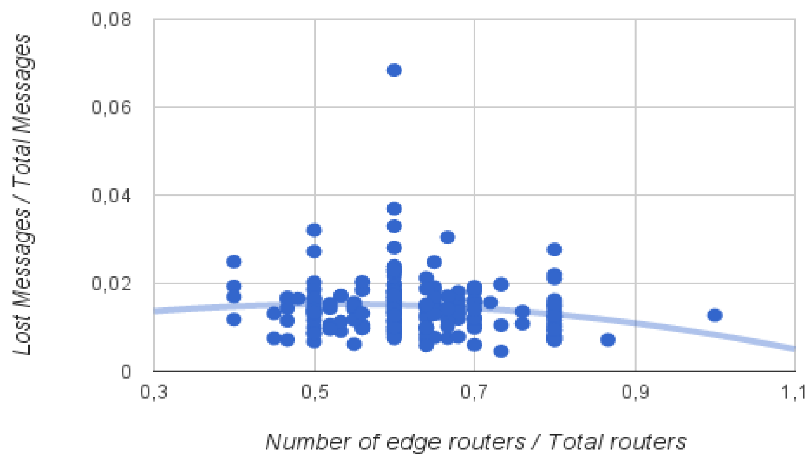

Figure 13. Ratio of lost messages by ratio of number of edge routers and number of total routers

6. Conclusions and Future Work

This paper presents the basis of a Fault Diagnosis Agent Architecture for Wireless Sensor Network (WSN) based on the BDI model. Different models that describe the diagnosis process and the network itself are used together to carry out a complete Fault Diagnosis task: from the Symptom Detection, through the Hypothesis Generation, to the Hypotheses Discrimination. The proposed architecture uses a Causal Model to relate the possible symptoms with their fault root causes and a Structural Model using the NDL language [23] to describe the network topology and the heterogeneous information that composes the global network.

For future work, we plan to extend our simulation environment to evaluate the proposed architecture with a autonomous diagnosis process, as we have focus in the symptom detection task in this work. For scalability and privacy reasons, we plan to explore the application of argumentation techniques among several agents that have a partial view of the global network. The deployment of several agents in a WSN could require some computational cost in the nodes where agents execute, but could get some interesting capabilities for distributed fault diagnosis process.

Acknowledgment

This research work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D project CALISTA (TEC2012-32457); and, by the Autonomous Region of Madrid through the program MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER).

References

[1] Akyildiz,Ian F., Su, Weilian., Sankarasubramaniam, Yogesh., Cayirci, Erdal.(2002). Wireless sensor networks: a survey. *Computer networks*, 38 (4) 393–422.

[2] Benjamins, Richard . (1995). Problem-solving methods for diagnosis and their role.I 8 (2) 93–120.

[3] Burchfield, Ryan, T., Venkatesan, S. (2007). Douglas Weiner. Maximizing throughput in zigbee wireless networks through analysis, simulations and implementations. *In*: Proceedings Int. Workshop Localized Algor. Protocols WSNs, 15–29.

[4] Chen, Bor-Rong., Peterson, Geoffrey., Mainland, Geoff., Welsh, Matt. (2008). Livenet:Using passive monitoring to reconstruct sensor network dynamics. *In*: *Distributed Computing in Sensor Systems*, 79–98. Springer,.

[5] Giorgetti, Gianni., Cidronali, Alessandro., Gupta Sandeep, KS., Manes, Gianfranco. (2007). Exploiting low-cost directional antennas in 2.4 ghz ieee 802.15. 4 wireless sensor networks. *In*: *Wireless Technologies*, 2007 European Conference on, 217–220. IEEE.

[6] Iglesias, Carlos A., Garijo, Mercedes., González, José C., Velasco, Juan R. (1998). Analysis and design of multiagent systems using mas-commonkads. In Intelligent Agents IV Agent Theories, *Architectures, and Languages*, 313–327. Springer.

[7] Kraaijeveld, Pieter., Druzdzel, Marek., Onisko, Agnieszka., Wasyluk, Hanna. (2005). Genierate: An interactive generator of diagnostic bayesian network models. *In*: Proceedings 16th Int. Workshop Principles Diagnosis, 175–180. Citeseer.

[8] Landsiedel, Olaf., Wehrle, Klaus., Götz, Stefan. (2005). Accurate prediction of power consumption in sensor networks. *In*: Proceedings of the Second Workshop on Embedded Networked Sensors. Citeseer.

[9] Lee, Winnie Louis., Datta, Amitava., Cardell-Oliver, Rachel. (2006). Network management in wireless sensor networks. *Handbook of Mobile Ad Hoc and Pervasive Communications*, 1–20.

[10] Liu, Yunhao., Liu, Kebin., Li, Mo. (2010). Passive diagnosis for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 18 (4) 1132–1144.

[11] Luke, Sean., Cioffi-Revilla, Claudio., Panait, Liviu., Sullivan, Keith., Balan, Gabriel. (2005) .Mason: A multiagent simulation environment. *Simulation*, 81 (7) 517–527.

[12] Manjunatha, P., Verma, AK., Srividya, A (2008). Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method. In Industrial and Information Systems. ICIIS . IEEE Region 10 and the Third international Conference on, 1–6. IEEE

[13] Miao, Xin., Liu, Kebin., He, Yuan., Papadias, Dimitris., Ma, Qiang., Liu, Yunhao. (2013). Agnostic diagnosis: Discovering silent failures in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 12 (12) 6067– 6075.

[14] Middleton, Blackford., Shwe, Michael., Heckerman, David., Henrion, Max.,Horvitz, Eric., Lehmann, Harold., Cooper, Gregory. (1991). Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base. *Medicine*, 30. 241–255 .

[15] Myers, Jack D. (1987).The background of internist i and qmr. *In*: Proceedings of ACM conference on History of medical informatics, 195–197. ACM.

[16] Pearl, Judea. (1988). Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

[17] Pearl, Judea. (2000). Causality: models, reasoning and inference, 29. Cambridge Univ Press.

[18] Provan, Gregory. (1995). Abstraction in belief networks: the role of intermediate states in diagnostic reasoning. *In* : Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, 464–471.

[19] Ramanathan, Nithya., Chang, Kevin., Kapur, Rahul., Girod, Lewis., Kohler, Eddie., Estrin, Deborah. (2005). Sympathy for the sensor network debugger. *In*: Proceedings of the 3rd international conference on Embedded networked sensor systems, 255–267. ACM.

[20] Ruiz, Linnyer Beatrys., Siqueira, Isabela G., Wong, Hao Chi., Nogueira, José Marcos S., Loureiro, Antonio AF (2004). Fault management in event-driven wireless sensor networks. *In*: Proceedings of the 7th ACM International Symposium on Modeling, analysis and simulation of wireless and mobile systems, 149–156.

[21] Shwe, Michael A., Middleton, B., Heckerman, DE., Henrion, M., Horvitz, EJ., Lehmann, HP., Cooper, GF.(1991). Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base. Methods of information in Medicine, 30 (4) 241–255.

[22]Skiena, S . (990). Dijkstra's algorithm. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley, p. 225–227.

[23] Van der Ham, Jeroen., Grosso, Paola., Van der Pol, Ronald., Toonk, Andree., Laat, Cees De (2007). Using the network description language in optical networks. In Integrated Network Management, 2007. IM'07. 10*th* IFIP/IEEE International Symposium on, p. 199–205. IEEE.

[24] ZG Wan, YK Tan, Yuen, C. (2011). Review on energy harvesting and energy management for sustainable wireless sensor networks. *In:* Communication Technology (ICCT), 2011 IEEE 13*h* International Conference on, p. 362–367.

[25] Wooldridge,M J. (2000). Reasoning about rational agents. Intelligent robotics and autonomous agents. MIT Press.

[26] Yu, Mengjie., Mokhtar, Hala., Merabti, Madjid (2007). Fault management in wireless sensor networks. *Wireless Communications, IEEE*, 14 (6) 13–19.

[27] Zhao, Yonggang Jerry., Govindan, Ramesh., Estrin, Deborah . (2002). Residual energy scan for monitoring sensor networks. *In*: Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, 1, p. 356–362.