Applying Model-Driven Development to Environment Monitoring System

Tran Cao De
Can Tho University, 1 Ly Tu Trong, Ninh Kieu, Can Tho, Vietnam

Keywords: Environment Management, Environment Monitoring System, Formal Method, Model-Driven Development, Sensor System.

Abstract: Environmental monitoring is critical in understanding whether the quality of our environment is getting better or worse. Information gathered by using an environmental monitoring system is important to make decisions. Vietnam is a vulnerable country of climate change. Specially, in the South of Vietnam, the Mekong delta is known as the region getting the most impact of sea level rise in Vietnam. That leads to a lot of problems making the worst effects to residents in the area, who are mainly still very poor. On the other hand, Vietnam is going on industrialization process that makes a strong effect on the environment. To deal with these challenges, different projects of environment management have been proposed and implemented and many monitoring systems have been built in those projects. Those systems are basically sensor networks with high cost in developing and maintaining. They are related to modern technology such as cloud, communication mobile and wireless. They provide the data for large community for different purposes. Therefore, building such a system is normally a long term project that requires an incremental and modular development for a complex system. This paper, on one hand, represents some common characteristics of an environment monitoring system that requires more study to develop a formal model and a methodology for their specifications, implementations and verification. On the other hand, we would like to adapt the formal model approach proposed for Intelligent Transport Systems (ITS) to an environmental monitoring system. The framework of Baobab is also introduced as an example for transformation from model to code.

1 INTRODUCTION

Environmental monitoring is critical in understanding the quality of our environment. The natural environment encompasses all living and non-living things occurring naturally on Earth or some regions. Many environment monitoring systems are concerned to large fields of interesting in environment, including air, water and other things related to social-economic development of a region.

Recently, Vietnam has been known as a vulnerable country of climate change and sea level rise. Mekong delta is the region getting the most impact of sea level rise. That leads to a lot of problems such as flooding, saltwater intrusion, and soil changes. These problems make extremely bad effects to millions of peoples, 80% of whom are farmers living poorly in rural areas. Vietnam is going on industrialization process that makes strong effects on the environment. To deal with these challenges, Vietnamese government, with support of different international program funds, is investing in many projects of environment management. Information technology has played a central role in these projects by the present of environment monitoring systems. One key step in developing an environmental management program is to assess the current state of the environment, normally, with supporting of an environment monitoring system. Such a system is based on sensor networks with high cost in development and maintenance. They are related to modern technologies such as cloud, communication mobile and wireless, and provide the data for large community including central government, local government, science and civil social-economic. The fact is that a monitoring system is complex and high cost. Therefore, it must be invested in a long term, different projects, and spent on different phases for a long term service. In one hand, it requires developing in an incremental cycle with the soundness of modular. It also requires engineering and re-engineering to adapt to new technologies of information technology, and assessing of

Cao De T..
Applying Model-Driven Development to Environment Monitoring System.
DOI: 10.5220/0005338205770584
In *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development* (CMDD-2015), pages 577-584
ISBN: 978-989-758-083-3

environment quality. In the other hand, there are a lot of entities involved, and various scripts and decisions which need to be implemented and verified to make sure the correctness in functionalities. In that context, Model Driven Development (France and Rumpe, 2007)0, (Hailpern and Tarr, 2006) appears to be a good practice on such systems.

Model-Driven Engineering (MDE) (Schmidt, 2006) is an approach to create software systems that involves creating models and applying automated transformations to them. The models are expressed in modeling languages (e.g., UML) which describe the structure and behavior of a system. MDE is often indistinctively associated to OMG's Model-Driven Architecture and Model-Driven Development. The ability to create a software design and apply automated transformations to generate an implementation helps to avoid the complexity of platforms, component technologies and frameworks. Many MDE solutions focus on generating code to fulfill the functional requirements as well as the quality attribute requirements such as performance or reliability.

In this paper, we represent some common characteristics of an environment monitoring system that can be considered as specifications in a high abstraction of these systems. Then, the proposed approach for Intelligence Transport systems is adapted to give a formal model for an environmental monitoring system. The framework of Baobab also is introduced as an example of generating code from the model in UML notation.

2 OVERVIEW ON ENVIRONMENTAL MONITORING SYSTEM

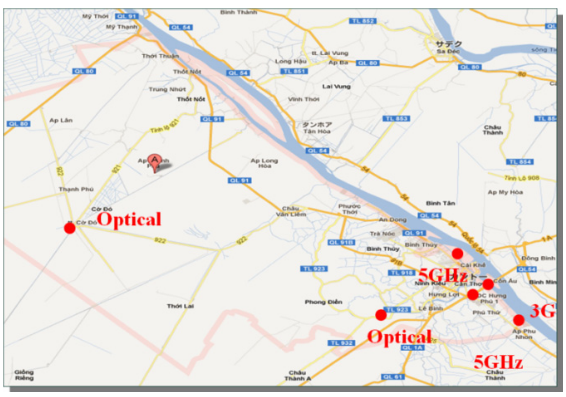

Figure 1: Six stations of the water environment monitoring system of Can Tho city.

Common Characteristics

Environmental monitoring system is a complex system that composes the processes and activities needed to characterize and monitor the quality of the environment. It is often used in the preparation of environmental impact assessments. It can be used as an information system that is designed to capture the current status of an environment or to establish trends in environmental parameters. The system automatically measures or captures the factors of environment characteristics, normally, in real time and continuity. There are innumerous concerns about the environment but a lot of monitoring systems is related to water, air, and soil. There are many parameters needed to measure in order to understand the quality of the environment. For example, some parameters of an air monitoring system of Vietnam (Circular 2/2014/TT-BNNPTNT) should be:

- Weather and climate: Temperature, wind direction, humidity, atmosphere, solar radiation, etc.
- Dust: SPM, PM10, PM2.5 and PM1
- Chemical: SO_2, NO and NO_2, etc.

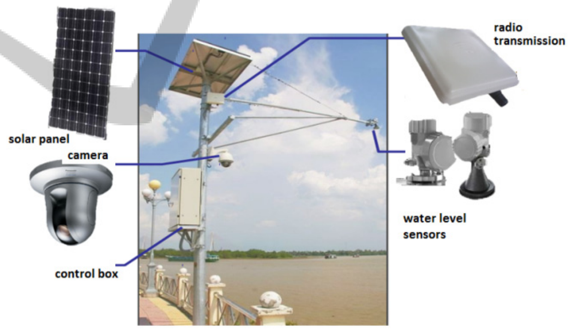

Figure 2: A fixed station for measuring water level of Can Tho city.

All environment monitoring systems (EVMS) must capture the interesting data from the environment. Therefore, they share some common characteristics: Sensor based system, Mobile system, and Distributed system.

From the architecture model, an EVMS is often designed for monitoring a geographical area that can be very different in scale from worldwide to a local area like an industrial zone. Monitoring system needs capture data from different sites in the monitoring area. Therefore, it is naturally a distributed system in which data is collected from different stations. Figure 1 gives an example about the water monitoring system in Can Tho city, which is composed by six stations.

Figure 3: Mobile station for monitoring the air.

A station may be a fixed one as in Figure 2 or mobile one as in figure 3. A station can be seen as a set of devices being set up at a place, including measurement instruments or sensors to capture environmental data. It provides data in the real time and in continuity. A station needs to be autonomy, i.e. so that it should properly work without a keeper. Therefore, it needs power to function and sends the captured data to a data center. A station can be a mobile station (see 0) to capture a status of environment in an area temporally. 0 represents a fixed station to measure water level of Can Tho city. It works in autonomy mode with solar power source, radio transmission for data communication, and a security camera. A station may use various types of sensors to get interesting data. The environmental status is interpreted from the captured data and it can be used as an indication of environmental quality at the monitored site. They are also transmitted to a monitoring center to be analyzed in associating to data coming from other stations. It is evident that this architecture can be repeated at a higher level. Many monitoring centers can share their own data to others or there is a bigger center that collects data from different monitoring centers in a wide geographical area.

Figure 4 gives an example of an EVMS in a general context. The system is composed of different monitoring stations. Each station can communicate with its center by a communication channel (radio transmission, for example). Then, the monitoring center collects data from its stations. In higher level, a monitoring center may be in a larger system that connects difference centers national-widely or world-widely. In a general view, collecting data from monitoring centers may not require any restriction on the type of center. These centers need not be homogeneity to provide the same kind of data, but a rich variety of data for different purposes. An EVMS is naturally sensor-based or instrument-based. EVMS uses sensors or measurement instruments to capture data from the environment. It will share common characteristics that synthesize in

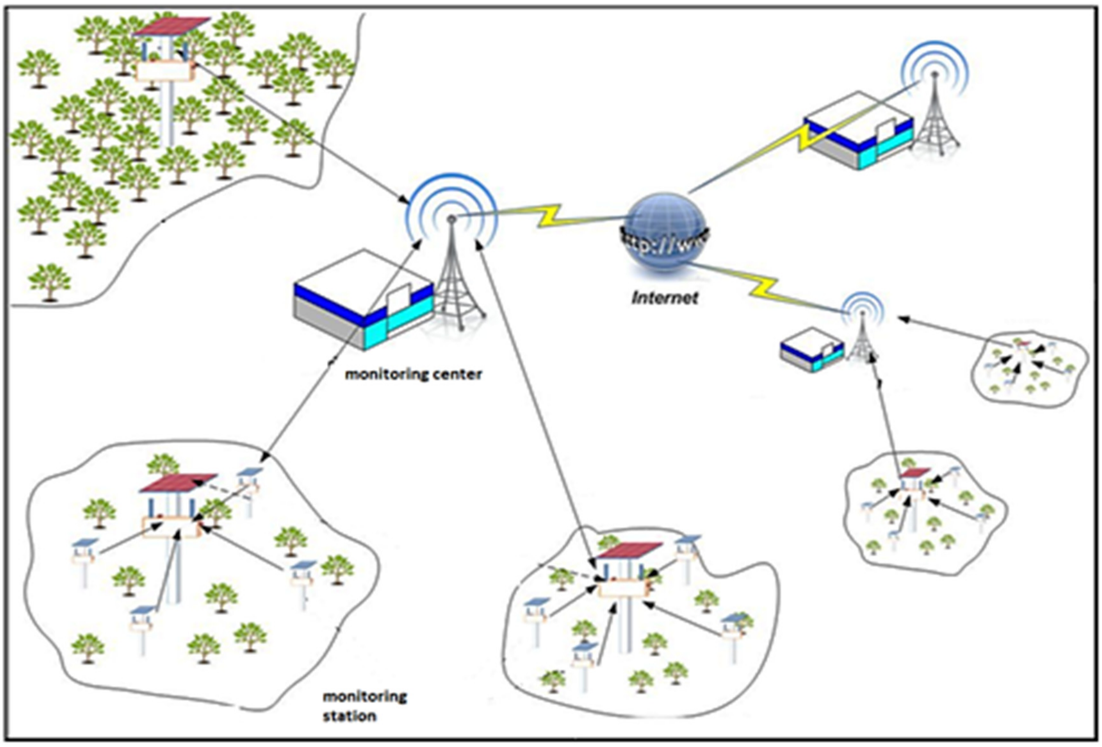

Figure 4: Distributed model of an ENMS.

(Meritxell et al., 2011), including:

- Physical distribution in deployment of system: the operation involves computation and information resources that are physically distributed.
- Resource boundedness: sensors must be able to operate though they are low-powered; their computation capabilities are low; and communication is slow and intermittent.
- Information uncertainty: incomplete information about the states of sensors, the network, and the environment
- Large scale: a numerous number of sensors, making unfeasible computational approaches that do not scale with the number of sensors.
- Decentralized control: sensors may need to coordinate to achieve tasks that cannot be achieved by the operation of a single sensor. Such coordination must occur without a centralized control.
- Adaptiveness: The operation of a sensor network is expected to be adaptive, namely the network must autonomously and dynamically adapt to either external and internal changes

Functionality

From the functionality view, a monitoring system is a long term service system. However, it is normally outdoor and vulnerability that needs maintenance to keep well-functioning. In this context, hardware may be changed that lead to re-implement the system to adapt it to new interface or services provided by the hardware. This is an important characteristic of an EVMS that may need to apply Modem-driven development to deal with.

An EVMS might be a large system, rich in hardware and functionalities but it might follow some patterns:

Sensor Level: at the input of any monitoring system, there are a lot of sensors for capturing different aspects of the environment. However, each sensor provides a kind of raw data, for example, water temperature, PH of the water, etc. In some case, more than one sensor is used to measure one feature of the environment. Normally, sensors can provide data continuously, but they later is collected and stored periodically. The first simple function of any monitoring system is to capture the simple value of each environmental quality factor and to make some warnings as soon as possible if the values obtained are out of acceptable range. Table 1 represents a sample data of a water monitoring station in Can Tho city, including water level, temperature, total dissolved solid, oxygen, electrical conductivity, turbidity, and salinity.

Figure 5: An air monitoring station in Ho Chi Minh City.

Table 1:A sample data of a monitoring station in Can Tho city.

Time	Water level	Temperature (°C)	PH	Total Dissolved Solids (TDS)	Oxygen (DO)	Electrical conductivity	Turbidity	Salinity
2013/12/31 00:01	2.54	23.69	8.21	0.1	0.09	0.16	2083.0	0.08
2013/12/31 00:06	2.57	23.7	8.21	0.1	0.09	0.16	2083.0	0.08
2013/12/31 00:11	2.6	23.69	8.17	0.1	0.09	0.16	2084.0	0.08
2013/12/31 00:16	2.63	23.68	8.19	0.1	0.09	0.16	2083.0	0.08
2013/12/31 00:21	2.65	23.69	8.19	0.1	0.09	0.15	2084.0	0.08
2013/12/31 00:26	2.67	23.68	8.16	0.1	0.09	0.15	2083.0	0.08
2013/12/31 00:31	2.7	23.67	8.16	0.1	0.09	0.15	2084.0	0.08
2013/12/31 00:36	2.72	23.68	8.15	0.1	0.09	0.15	2083.0	0.08
2013/12/31 00:41	2.74	23.67	8.15	0.1	0.09	0.15	2083.0	0.08
2013/12/31 00:46	2.76	23.66	8.12	0.1	0.1	0.15	2083.0	0.08
2013/12/31 00:52	2.78	23.65	8.18	0.1	0.1	0.15	2083.0	0.08
2013/12/31 00:57	2.8	23.66	8.14	0.1	0.1	0.15	2084.0	0.07

Station Level: a station may be composed by many sensors of several types to give a set of features about the environment. Its sensor system can be designed to give a set of values for a special purpose. Figure 5 gives an example of air monitoring station in Ho Chi Minh City. The functionalities of a monitoring station may be quite simple and programmed to manage a set of sensors, collecting data from its sensors periodically and sending them to a data center to store and use. This collection task and transferring data to data center, normally, are the charge of a programmable controller. In this simple model, a monitoring station may be designed to work as an autonomy system including: power system and the security system (for the station itself). It also is designed for providing more complex function, for example, displaying collected data and the status of the environment or alerting automatically if any environment characteristic violated.
A station can be seen as an outdoor system, some important characteristics are required:

- Fault tolerance: a station still works properly even if some sensors are corrupted.
- Easy to maintain: all components of the station can be replaced easily for example, replacing the power system (or battery), replacing the sensor, replacing the main controller board. The replacement may not need to keep the same kind of hardware.

A station is programmed to function. All stations in a monitoring system are not necessary homogeneity, i.e. the same kinds of hardware, drivers and architecture. Developing software for a station may need a flexible model to deal with these characteristics, specially, to deal with change (hardware and drivers) and regression test (or retest) the system. Such a model is also needed to deal with the instantiation of a system for the same functionality but different on hardware and architecture. Figure 6 represents the detailed model of water monitoring system of Can Tho city. In the system there 6 stations with the same functionality that is measuring a set of characteristics of water, but they are not homogeneity, for example in data transmission way. How Model-driven development can contribute to this issue?

Monitoring Center: main functions of a monitoring center are to receive the collected data from different stations, store them to analyze the environment status, and then predict the environment trends. It also exchanges its own data with the other monitoring systems world-widely or national-widely. Therefore, it can be seen as a data center for multi-purposes for a long time. For example (see figure 6) the water monitoring system of Can Tho city collects data from its six stations, store them in a private cloud. It also transfers its own data to NISCI, a national monitoring center via the Internet. Actually, this system is in function to capture environment data. The use of these data now is quite simple: showing the variations daily, weekly, monthly or yearly (see figure 7); alerting the violation of standard safety criteria.

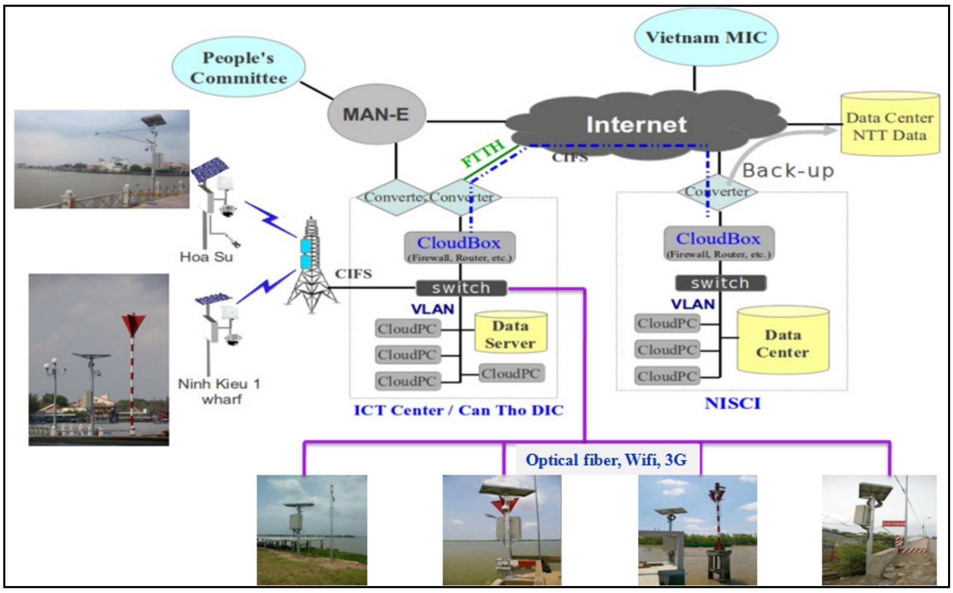

Figure 6: The detailed model of water monitoring system of Can Tho city.

The functionality of the water monitoring system of Can Tho illustrates a simple function of a monitoring center: data collection, data store, and simple analysis to alert the standard violation, providing information for administration and management purposes, and transferring data to another monitoring system at a higher level.

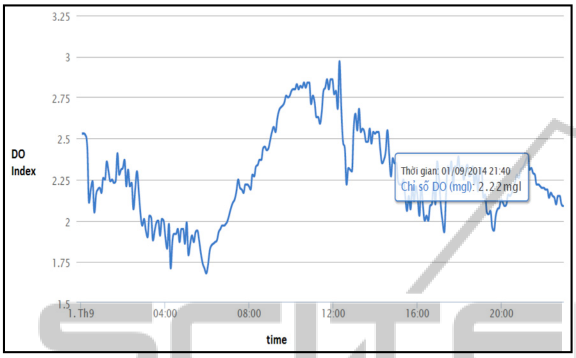

Figure 7: Variation of DO indices at the station E of the water monitoring center of Can Tho on September 1st, 2014.

Exploiting collect data for complex analysis to establish environment status or predict the trends of the environment is very special issues that need different advance math models or machine learning techniques. We do not intent to go in depth to these subjects. Here we limit ourselves at simple functionalities of a monitoring center.

In data collection and storing, the system needs to be changed to add new indices of new characteristics of the environment. At the processing tasks to alert the standard violation criteria, the system need to be changed for updating new calculation model or math formula or adapting to new criteria (i.e. new thresholds).

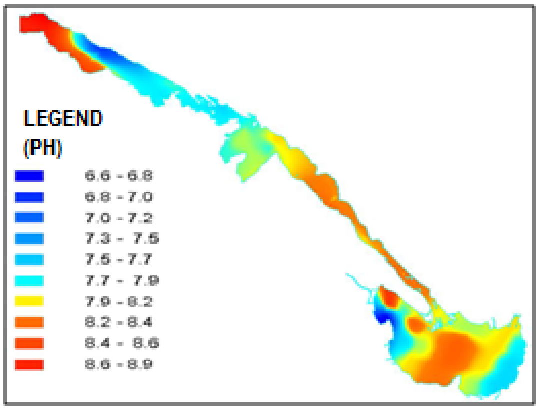

Figure 8: Average of PH of water of Tam Giang lagoon in the dry season (no rain fall).

Providing information to administrative or management is a complex issue. The collected data need to be analyzed, synthesized, and represented in an appropriate format. For example, in figure 7, the raw data about DO index on 1/9/2014 is synthesized and drawn continuously. A similar synthesis may be done weekly or monthly. The collected data from different stations can be synthesized and represented as a map to give a general sense about the environment as in figure 8.

The web page of Vietnam Environment Administration (http://www.quantracmoitruong.gov.vn) provides draw data and synthesized information in a few formats from a lot of monitoring centers in Vietnam.

In a higher hierarchy level, a monitoring center can be a branch of another monitoring center, or simply transferring its own data to a lager data center to store and use.

3 APPLICATION OF FORMAL MODEL

In the previous section, the common functionalities and characteristics of an EVMS have been represented. Such a system shares many characteristics of an ITS (Intelligent Transport Systems) introduced in (Bonnefoi et al.,2007). In fact, an EVMS also has 4 problems identified by (Bonnefoi et al, 2007):

Problem 1, Modeling Diagrams: it needs to ensure that the specifications will be adapted to perform a formal verification for a large set of main concerns, including sensors, batteries (or power sources), controller boards, communication devices, ports, etc. The standard UML notion and the adapted methodology proposed in (Bonnefoi et al.,2007) might be applied.

Problem 2, Structuring the Specification: EVMS has the same context as ITS. Beside of a large set of heterogeneous elements to specify, there are also numerous actors that are communicating and interacting in parallel, such as time event, sensor status, threshold came over… An appropriate template as in the case study of (Bonnefoi et al.,2007), that enables variation of architectures and components parameters, might be a good idea.

Problem 3, Specifying Behaviors: UML State diagram is good for expressing internal behaviors of components. However, as stated in (Bonnefoi et al.,2007)., under some conditions, it is possible to transform a state chart into a formal notation like Petri Nets for analysis purpose.

Problem 4, Analyzing the System: Analysis of a system, formal methods as theorem proving and model checking are normally used.

By sharing the common characteristics and problems between EVMS and ITS, we adapt the solutions proposed in for ITS to EVMS. This approach, by using UML and Petri net, can resolve some issues, including:

- Modular architecture of the system
- Definition of components interfaces
- Definition of components behavior
- Assembling of the formal model
- Analysis.

The proposed approach is compatible with software design; the methodology relies on UML and sticks to the V-model for software life cycle. It introduces a guideline to designers to achieve EVMS specifications that is suitable for formal analysis and verification. It also shows a way to enhance the use of the achieved results from one modeling formalism to the others. The interesting point is the full automation of translations from UML models into Symmetric Nets models in the framework of a development tool as Eclipse Modeling Framework (Budinsky et al., 2003), a basis for such an implementation.

The case study represented in (Bonnefoi et al.,2007). illustrates that we can use UML notation and Petri net to specify an ITS that can be implemented by a development tool as Eclipse. However, there is no detail of the implementation, neither transformation to code. For this issue, Baobab (Bahar et al., 2009) might give us a good example.

At a higher abstraction, EVMS naturally is a kind of wireless sensor system (WSN). Baobab is known as a model driven development (MDD) framework for WSN applications. It provides a generic meta-model (GMM) which is versatile across different application domains including EVMS. Baobab allows meta-model users to extend GMM for defining their own domain-specific meta-models and platform-specific meta-models. Baobab's model-to-code transformer type-checks and validates a given application model and generates application code in nesC (Gay et al., 2003) for TinyOS (ww.tinyos.net/). Here, we do not intend to go further in Baobab. We just introduce its meta-models and models for WSN applications. In fact, Baobab defined Generic Meta-model Elements which represent sensor devices used in WSNs. All sensor classes, representing a specific type of sensors, extend from the base class *Sensor*. The most common sensor types that can be used in a variety of applications are defined in the generic meta-model.

For functional requirement specification, Baobab also defines several elements to express the most common functional aspects of WSNs. The functional tasks can be modeled by using *ConditionalFunctional* element of the GMM and they can further be specialized into *RepetitiveTask*. There are other tasks used to define the receipt of data as *DataReceipt* and to request a waiting period before another task can be called, as *WaitingTask*. Sensing phenomena can be modeled with *SensingTask*. This element retrieves the new-created *SensorData* from the Sensor.

The non-functional requirements can be modeled explicitly by means of *NonFunctionalTask* class in the GMM. Some examples of non-functional tasks defined in the GMM are: *ClusterFormation* for dividing the network into clusters; *ChangeSleepTime* for adjusting the sleep time to minimize energy. Main classes defined in Baobab are represented in Figure 9.

Baobab provides a domain specific modeling language to specify WSNs and generate code of such systems. Moreover, it also illustrated that we can build the model with UML notation and implements it by using a development tool as Eclipse.

4 CONCLUSION

In this paper, we presented common characteristics of EVMS that share characteristics of ITS. An EVMS is a complex system based on sensor networks, organized in hierarchy structure, including sensor level, station level and monitoring center. An EVMS consists of sensing devices distributed in a wide area, which coordinate to produce meaningful information. Sensor networks are usually wireless, dynamic networks composed of a large number of (possibly heterogeneous) sensors. These sensors acquire pieces of information that may substantially differ in content, resolution, and accuracy. It is related to a large set of devices that are placed outdoor, in the environment. Therefore, it needs to do a lot of work on maintenance to ensure it works properly. As a consequence, it is necessary to have some appropriate approaches to facilitate the specification, design, development and maintenance of the system. In fact, it needs some formalism for modeling, structuring the specification, and

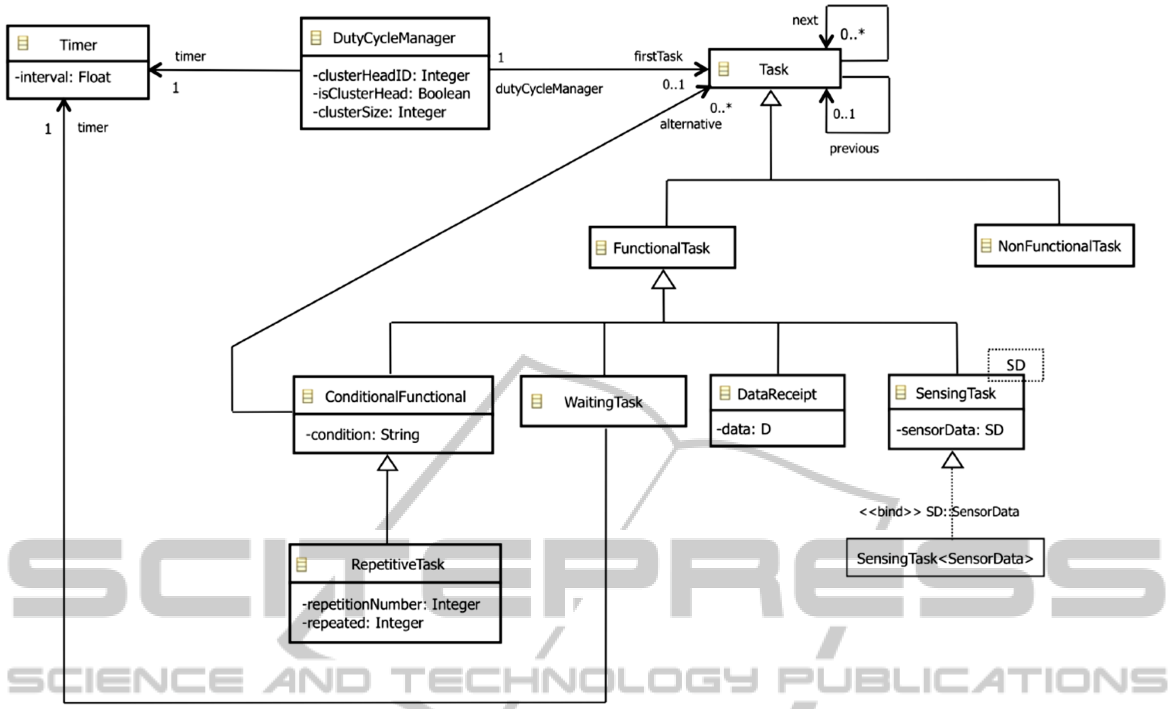

Figure 9: Functional Aspects of the Generic Meta-model in Baobab (Bahar et al., 2009).

specifying system behaviors and analyzing. Specially, it also needs a flexible model to deal with changes in its life cycle. The formalism proposed for an ITS based on UML and Petri net might be well-adapted to an EVMS. It is suitable for adding new sensors, replacing sensors, changing data collection schedules, changing data representation, adding new environmental criteria.

An EMVS is basically a sensor network and might be a kind of wireless sensor network. Therefore, the framework provided by Baobab has illustrated a way to specify an EMVS (specialized of WSN) and generate its code. It can be seen that the formalism in Baobab can be expressed in UML notation and implemented in other language as well. These features provide a lot of advantages for incrementally developing and maintaining an EVMS as a long term project.

REFERENCES

France R, Rumpe B., Model-Driven Development of Complex Software: A Research Roadmap, *Proceeding Future of Software Engineering*, Fose'07, 2007, Pp. 37-54.

Hailpern B. and Tarr P., Model-Driven Development: the Good, the Bad and the Ugly, Ibm Systems Journal, Vol. 45, No. 3, Pp. 451-461, 2006.

Schmidt, D.: Model-Driven Engineering. Ieee Computer Magazine 39 (2), 2006.

Circular 32 of the Ministry of Agriculture and Rural Development, 32/2014/Tt-Bnnptnt, Dated 10/09/2014, Vietnam.

Meritxell Vinyals, Juan a. Rodriguez-Aguilar, Jesus Cerquides , a Survey on Sensor Networks from a Multi-Agent Perspective, the Computer *Journal Archive, Volume 54 Issue 3, March 2011,* Pp. 455-470.

Bahar Akbal-Delibas, Pruet Boonma, Junichi Suzuki, Extensible and Precise Modeling for Wireless Sensor Networks, Information Systems: Modeling, Development, and Integration, Lecture Notes in *Business Information Processing Volume 20*, 2009, Pp 551-562.

David Gay, Phil Levis, Rob Von Behren, Matt Welsh, Eric Brewer, and David Culler, the Nesc Language: a Holistic Approach to Networked Embedded Systems, *Proceedings of Programming Language Design and Implementation (Pldi)* 2003, June 2003.

Http://Www.Tinyos.Net/

Bonnefoi, F., Hillah, L.M. ; Kordon, F. ; Renault, X. Design, Modeling and Analysis of Its using Uml and Petri Nets, Intelligent Transportation Systems Conference, Itsc 2007. Pp. 314-319.

Http://Www.Quantracmoitruong.Gov.Vn/En-Us/En/Home.Aspx

F. Budinsky, D. Steinberg, E. Merks, R. Ellersick, and T. Grose, Eclipse Modeling Framework, Ser. the Eclipse Series, E. Gamma, L. Nackman, and J. Wiegand, Eds. *Addison-Wesley Professional*, August 2003.