Perception-Oriented Model Driven Development for Designing Data Acquisition Process in Wireless Sensor Networks

K. Indra Gandhi

***Abstract*—**Wireless Sensor Networks (WSNs) have always been characterized for application-specific sensing, relaying and collection of information for further analysis. However, software development was not considered as a separate entity in this process of data collection which has posed severe limitations on the software development for WSN. Software development for WSN is a complex process since the components involved are data-driven, network-driven and application-driven in nature. This implies that there is a tremendous need for the separation of concern from the software development perspective. A layered approach for developing data acquisition design based on Model Driven Development (MDD) has been proposed as the sensed data collection process itself varies depending upon the application taken into consideration. This work focuses on the layered view of the data acquisition process so as to ease the software point of development. A metamodel has been proposed that enables reusability and realization of the software development as an adaptable component for WSN systems. Further, observing users perception indicates that proposed model helps in improving the programmer's productivity by realizing the collaborative system involved.

***Keywords*—**Model-driven development, wireless sensor networks, data acquisition, separation of concern, layered design.

I. Introduction

WSN is a fast emerging field [6] which gives way for ubiquitous computing making the technology closer to the human community. Since WSNs are application centric, MDD can help in reusing the models developed for different applications. Also, MDD-based step wise refinement supporting software development process has been proven to be successful in developing domain specific languages [1]. Taking into consideration the separation of concern, designing a metamodel for the layered view of the components is the sole focus of the system. The layers at the high level are classified into the application, domain and technical services.

Recently, the need for abstraction and separation of concern has been realized extensively for WSN applications. This is because the wrapping of hardware and software aspects of sensor nodes has restricted the reusability of the system. A modelling framework for hardware and software aspects of the system has been developed [2] with focus on separation of concern. The phases of the modelling are classified into the analysis phase, design phase and implementation phase. In the analysis phase, the requirements of the system are discussed in a detailed fashion. Since it is application-centric modelling, there is always a chance in the evolution of the existing application. This key point is a noted advantage for providing a framework based on MDD. Henceforth, instead of evolving for each application, these frameworks enable the variation emerging in the application to be integrated with the existing system. This challenge of variation and evolution that arises in the existing system over a time period is easily manageable through the proposed layered approach.

In the design phase, the functionalities of the objects are formulated by mapping with the real-time applications. The design of the classes across the layers is emphasised such that the formulation of the overall system will be done in an efficient manner. Another significant point is that the overall process is iterative in nature, and therefore, any refinement can be carried out in the next iteration provided it is not a major change, because a major change can affect the entire focus of the system which has been determined in the analysis phase.

The implementation phase constitutes the mapping of the design to the actual coding with reference to MDD. This is termed as forward engineering, where the system is realized in terms of the actual coding. The objects with respect to the actual coding can be recognized as individual class files, and therefore, any change in one of the class can be highlighted only in that class without affecting the other classes by referring the dependency of the classes from the system design. These classes are logically related, and as such, most of the objects are reusable for some other applications. In this work, a model for data collection in WSN is developed based on MDD using the layered approach. This can be used for various applications with minimum set of change in the requirements. The perception of the user with respect to these approaches is also identified and categories through which the users are able to identify the model are also developed. These phases complete one full cycle of the modelling process. This is an iterative process and each iteration refines the model. This work is an initiative to determine whether these varying emerging fields can be subjected to MDD approach which will have a major benefit for a range of applications.

II. Related Work

There have been extensive research outcomes for data collection in Wireless Sensor Networks available in the literature. However, quite interesting works for modelling approaches are explored in order to emphasise the need for modelling.

K. Indra Gandhi is with Anna University, Chennai-600025, India (phone: 91- 44-2235-8844, e-mail: indra@annauniv.edu).

It has been studied that Model-Driven Engineering (MDE) [3] is considered as the right tool to reduce the complexity of WSN development through its principles of abstraction, separation of concerns, reuse and automation. A modular model [5] that integrates realistic descriptions of multi domain models has been proposed, which will provide flexibility for a variety of simulation studies. The modular nature of the MDD approach adopts any mixed formulation such that all the components can be integrated irrespective of their variation.

A distributed data model for an integrated Geographic information system has been developed [9]. This implies that the modelling for diverse information systems using UML has proven to be successful. The development of such an integrated software component is made possible by only using model-driven strategy. The object oriented data model design leads to seamless assimilation of the classes and their relationships directly in software development.

The process of an autonomous mobile robot with an object oriented approach has been modelled [7]. The robot task is modelled by a Business Process Modeller that links to the robot control structure modelled by Unified Modelling Language (UML). The realization of the control algorithms is extracted directly from the object oriented model. A model for information flow control which is essential for a system that manages sensitive information has been developed [11].

The object-oriented approaches lay the foundation for modelling, describing, analysing and designing a system [10]. Several models [4] and abstractions [8] have been explored in the literature. A special emphasis was made on the iterative approach of modelling. Moreover, a layered approach to various problems provides an insight into the manner the systems to be developed are dealt with.

The twisted components involved in WSN have limited the software development part which can be initiated through modelling from different viewpoints of the system. The interaction between the physical system (sensor) and the logical system (data handling) is always a matter of fact to be taken into consideration. Since the data acquired can be diverse in nature as well as the source from where the data is being relayed is again dynamic in nature, there is a necessity to place a balance between these two varying systems. The possible solution in such a scenario is to provide a model-driven layered approach in which the separation of components can be identified for a complex data acquisition system.

III. Problem Description

In WSN, sensor nodes are randomly deployed in the region of interest. The data sensed by the sensor nodes are relayed to the next-hop nodes until it reaches the sink node. Sensor nodes and the sink node require a separate set of functionality, where the former senses, relays the data and the latter processes the data. These nodes are connected through a common medium of communication. The sensed data can be realized from a display unit which is connected with the sensing unit. The focus of MDD for WSN can be separated as illustrated in Fig. 1.

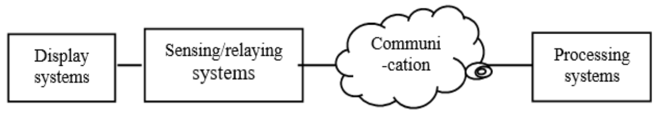

Fig. 1 Problem Separation

There are different types of sensor nodes which can differ for different applications. The sensor node is identified as an integration of hardware and software component. Also, there are different types of sensors, which is illustrated through the UML representation as shown in Fig. 2.

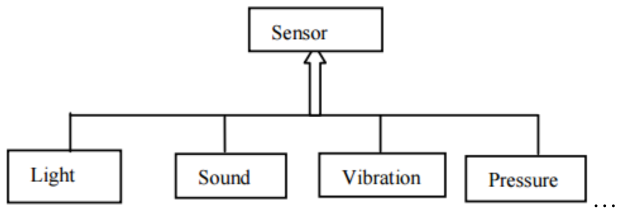

Fig. 2 Generalization of sensor nodes

The dotted line represents the incomplete generalization, which denotes that any other type of sensor if evolves can be added to the existing system. Similar generalization design can be observed for other components of Fig. 1. The software development process for data acquisition in WSN is the system under consideration. The design model for such a system is elaborated in Fig. 3. This is an initiative towards identifying the objective of the system taken into consideration. At the initial set, a sensor field consists of 'n' number of sensor nodes. The components of the sensor nodes itself constitute another extendable design. Since the aim is to concentrate on the acquisition of data, the sensor node is considered as one entity. Each rectangular box is identified as an individual class.

The data to be sensed is a complex object and itself embeds various processes. The data that is sensed is relayed to the next-hop node which is indicated by a self-association. The data is relayed through an external system which is the network subsystem, as shown in Fig. 3.

On the other end, the data is received by the sink node for further analysis. The model has been developed through open source UML tool and the respective codes have been generated; the individual classes identified can be developed according to the given specification.

A. Layer to Layer Communication

The system is modelled on the basis of the logically related components. The possibility of a direct communication across layers is less possible because the layers are concerned with a different set of requirements. The UI concentrates on the way the input can be acquired from the other sources. This information has to be received by the domain for core processing of the information.

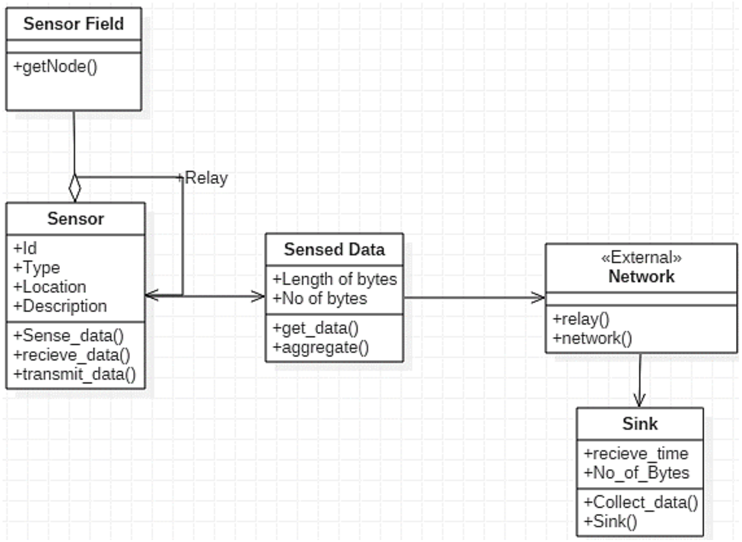

Fig. 3 High Level Design Class Diagram

In Fig. 4, the layered representation of the system under consideration is explained in detail. Each layer illustrates a package where all the logically related classes are wrapped in. The User Interface (UI) layer constitutes the display unit where the data sensed is as presented in the user readable form of information. The logical layer explains the core domain of the problem to be elaborated. The services layer describes the non-functional requirements of the system such as persistence, security, logging details etc. For example, the data has to be stored and retrieved in persistence storage and this is taken care of the persistence package. The logging details of the data are a necessary component which is handled for web services.

Since the domain layer establishes the core logic of the information processing, a common intermediary class is required to handle the incoming requests from the UI layer. This is handled by the System controller (Root system), as shown in Fig. 4. The sink node has its own set of functionality which is considered to be a subsystem from the sensor node's perception. A persistence service initiator is required in order to access the persistence subsystem. Logging information of the data acquisition by the sensor nodes can be accessed through a logging interface.

Therefore, all the subsystems have to be communicated across layers through interfaces. This is because any change in one of the subsystem should not affect the robustness of the system. This MDD based data acquisition software system can be extended, evolved, and iterated further with reference to the requirement of the system which are then considered to be the base for software development process.

B. Metamodel for Data Acquisition Process

Since the workflow model of WSN is complex in nature, there is always a strong need for developing a metamodel of the system. This will help to extract and unify the software components which are embedded with the hardware components of the system. The presence of a metamodel will provide a clear workflow of the system such that the consistency of the system can be maintained. In addition, the metamodel will also provide the possible ways for developing the domain specific modelling language.

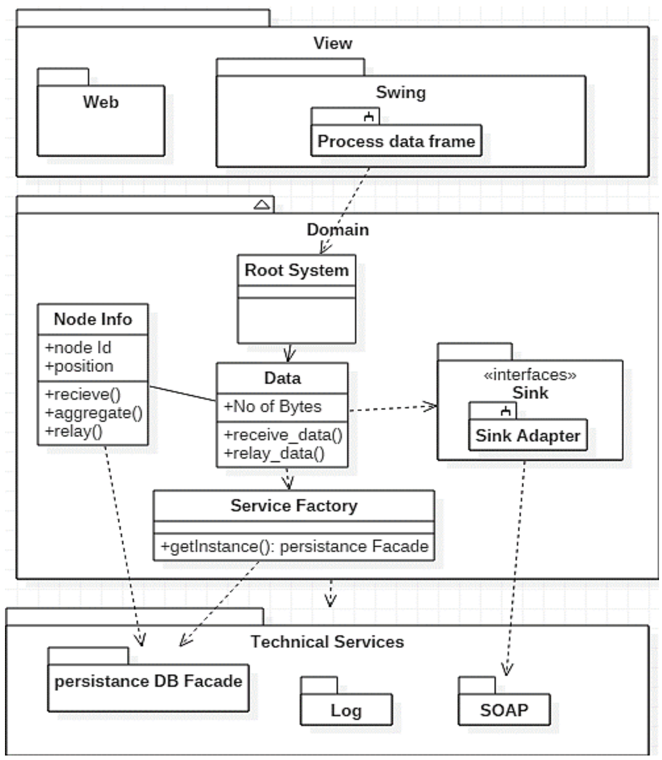

Fig. 4 Layered Design for Data Acquisition

The data gathering from the sensor nodes in a multi hop fashion comprises the logical structure of the system. The metamodel consists of the communication model involved in the network-driven process and the data acquisition model which involves the data driven process. The metamodel for a network driven process is illustrated in Fig. 5.

The nodes relay data to each and every other node in the network. The relaying of data from one node to the other node is based upon various protocols across various layers. The communication across these layers is through transmission of packets from one layer to the other layer. The packet structure holds the state of different parameters. The data driven process can be through various processing techniques which is shown in Fig. 6.

The sensor nodes may continuously monitor the sensing region such as habitat monitoring and the sensing process may be based upon some event such as earthquake, volcano explosion etc. The data acquisition process may vary depending upon the type of sensing. The data acquisition can be processed through various techniques which inherits its own set of complexities. The data sensed by the various sensors may be of different format which should be generalized for further processing. There may be other upcoming techniques which can also be included in the extension of the developed metamodel.

The high level structure and behaviour of the metamodel with reference to the context have been illustrated through UML diagrams and the validation process have been carried out successfully. The metamodel definition with respect to the data acquisition context has been presented in Fig. 7. Initially the sensors have to undergo a self configuration process which acts as the preliminary step for their association with the other sensor nodes. Since the sensor nodes are deployed randomly in the region of interest, the communication of these sensor nodes for relaying of data has to be initiated. This is a networking process and hence to be handled by the communication model. The process of collection of these sensed data from the sensor nodes is handled by the data driven model. This separation is crucial from the software point of view, since the variation point leads to various challenges in the system development.

The responsibilities of the individual objects have been modelled so as to develop the operations of these objects whenever a variation arises. The data acquisition process may comprise the various techniques involved in aggregating the data and then relaying to the next-hop node until it reaches the sink node. The sensor taken into consideration may be different for different sensing environment.

Through the starting point, main () is responsible for setting up the initial communication across the sensor nodes. The data acquisition process is initiated through the data collection techniques. The behaviour of the defined process is elaborated in Fig. 8 as a collaborative illustration.

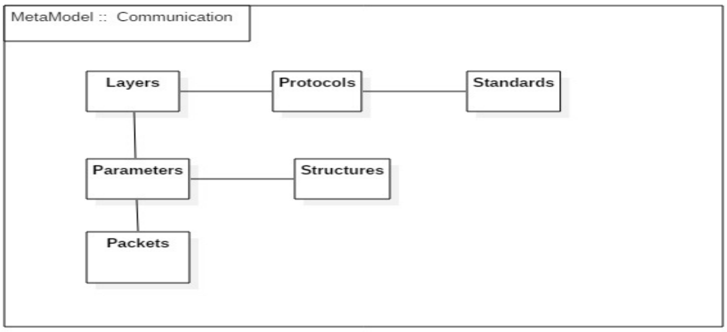

Fig. 5 Metamodel for the network driven process

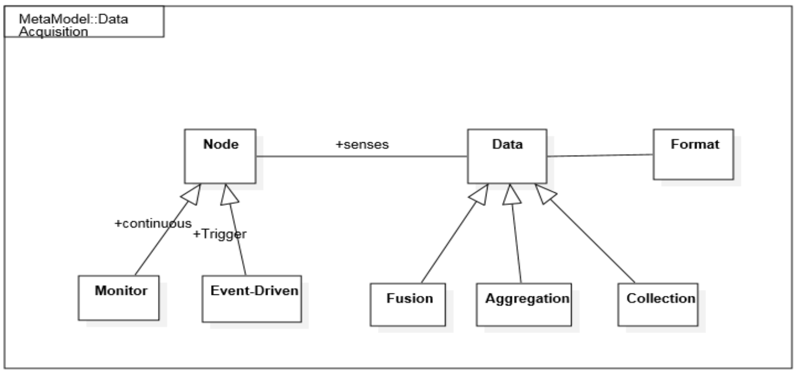

Fig. 6 Metamodel for data acquisition process

The main class initiates the process, self configuration for sensing, initiating the relaying process and the collection of data through the defined process has been designed. This model and their behaviours exploration will further motivate to elaborate the required software development involved in the process taken into consideration. This will further improve in minimizing the complexity involved in developing domain specific data acquisition systems in WSN. This is very crucial for WSN, since the system requirements differ for different applications.

IV. User Perception

In this study, 40 students who have been familiar with modelling concepts and UML tools were taken into consideration. They were provided with the system description with special focus on separation of concern. These models were collected and observed for further analysis. From the models provided by the users, it has been identified that the users were placed into five categories.

From the separation point of view, the users were able to model the data acquisition system, which is illustrated in Fig. 9. In the first iteration, the users were able to perceive the separation of hardware/software and this group were placed in category I; this is because since the users focus was on software development, they were able to segregate the software specifications from the hardware specifications. There was a mixed response for category II since the users argued that the network-driven specification also consists of a flow of data which again leads to data-driven approach. This is an important view to be discussed since only the separation of the layers makes them to identify the focus of the system.

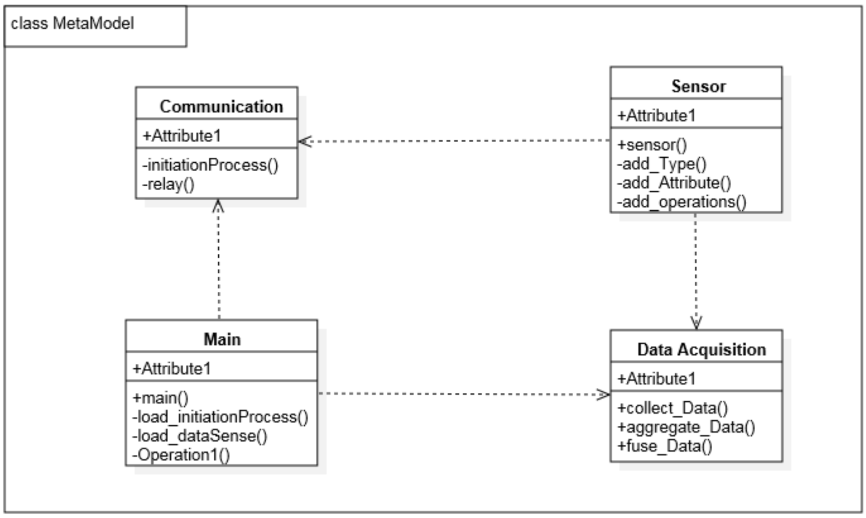

Fig. 7 Domain specific metamodel definition

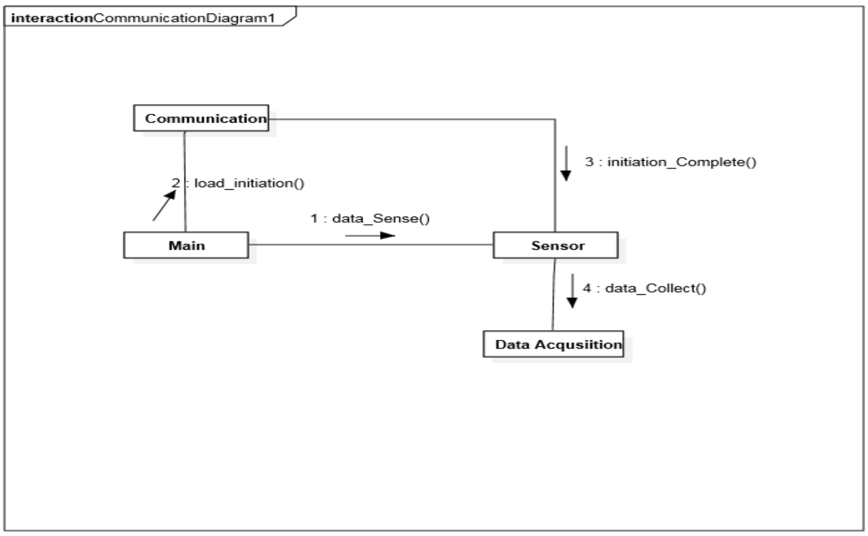

Fig. 8 Behaviour of data acquisition process

TABLE I
User Modelling Category

User Group	Category
I	Software/Hardware separation
II	Network-driven/Data-driven separation
III	UI/Logical layered separation
IV	Logical/Service layer separation
V	No Separation of Layers

Under category III, the users were able to separate the logical and UI layer, since they are familiar with programming, and henceforth, were able to separate the processing logic from the standard input and output. It was a tough task to separate the services and the logical layer under category IV, since users were unable to make a clear picture of the functional requirements (data acquisition) and the services provided to these requirements such as security, persistence, logging etc.

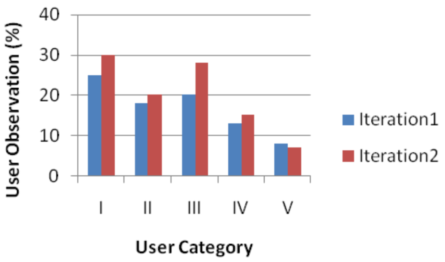

Fig. 9 User observation

Before the second iteration, the users were provided with the strategies involved in separation, impact of layered approach, reusability of the existing components, the expected evolution of the existing components and the users were made to refine their models accordingly. Motivated with this perception, the users were able to come up with a logical/layered refinement in the second iteration. Around 7% of the users were wrapping all the components of the system into a single who are observed to be poor performers.

In the second iteration, a major refinement was not observed in such users. However, 25% of the users were able to move from average to good performers by designing the model through a layered approach. Around 15% of the users, being good performers, were able to separate the model through a layered approach. The performances of the users were measured through their programming development time and reusable approach of the existing components. The layered approach facilitated the users to observe the separation of concern existing in the data acquisition system and thereby distinguishing the software development process from the other collaborating systems.

V. Conclusion

In this paper, a MDD based layered approach that differentiates the software development for data acquisition WSN has been proposed. The core idea is to identify the components of the individual layers and to logically relate the system which is otherwise physically connected. The solutions for the various problems related to data collection can be presented individually depending upon the complexity of the problem through the metamodel development. Irrespective of the problem, the model can be used as the basic building block upon which various other approaches can be built. The user perception oriented design indicates that by using the proposed approach, the users based upon the category identified, refine and develop their system efficiently. Through the modelling of the system, 30% of the users showed refinements which have been observed in successive iterations. Future work is to evaluate for a larger number of iterations for different user groups who have been involved in model-driven software development systems, and thereby developing a cognitive model for such systems.

References

[1] A K. Tei, R. Shimizu, Y. Fukazawa and S. Honiden, "Model-Driven-Development-Based Stepwise Software Development Process for Wireless Sensor Networks," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, no. 4, pp. 675-687, April 2015.doi: 10.1109/TSMC.2014.2360506.

[2] K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostarda and H. Muccini, "A model-driven engineering framework for architecting and analysing Wireless Sensor Networks," 2012 Third International Workshop on Software Engineering for Sensor Network Applications (SESENA), Zurich, 2012, pp. 1-7. doi: 10.1109/SESENA.2012.6225729.

[3] I. Malavolta and H. Muccini, "A Study on MDE Approaches for Engineering Wireless Sensor Networks," 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, Verona, 2014, pp. 149-157. doi: 10.1109/SEAA.2014.61.

[4] Barnaghi.P (2010), "Publishing linked sensor data ", International Semantic Web Conference, 5, pp. 1-16, Shangai, China.

[5] Gianni Ferretti et al. (2013), Object-oriented Modelling for general flexible multibody systems, Mathematica; and computer modelling of dynamic systems.DOI: 10.1080/13873954.2013.807433.

[6] I. F. Akyldiz et al. (2008), Wireless Sensor Networks: A Survey, Computer Networks, 38,393-422.

[7] JO Junior et al. (2004), Modelling Autonomous Mobile Robot system with an Object-Oriented Approach, ABCM Symposium series in mechatronics,1,25-32, ABCM, Brazil.

[8] Kenji Kodama et al. (2008), A Rule-based Acceleration Data Processing Engine for Small Sensor Node, Proceedings of the 3rd International Workshop on Middleware for sensor networks, pp.1-6, ACM, USA.

[9] Kumar et al. (2010), An Object-Oriented Shared Data Model for GIS and Distributed Hydrologic Models, 24, 1061-1079.

[10] Larman.C (2010), Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Pearson Education.

[11] Shih-Chien and Chia-Wei Lai (2007), An Information Flow Control Model for both Object-Oriented and Non-Object Oriented Systems 23,479-496.