Sensor/Data Fusion Design Pattern and Implementation as a Toolbox in Matlab/Simulink (SDFTool)

Majid Kazemian, Behzad Moshiri, Amir Hosein Keyhanipour, Mohammad Jamali, Caro Lucas

Control and Intelligent Processing Center of Excellence, Electrical and Computer Engineering Department, University of Tehran, Tehran, Iran
am.kazemian@ece.ut.ac.ir

***Abstract*— System level tools are becoming more popular for designing different applications. In these tools, application designers describe their designs at high level using the powerful modeling environment such as MATLAB/Simulink. In this paper, SDFTool is presented as a design pattern and implemented as an add-on toolbox to address this issue. The three major functionalities offered by this view are: integration of a domain-specific modeling technique for rapid and accurate system design, modularity which means that the new components can be easily added, reusability and easiness of use which are the most important aims of design patterns.**

I. Introduction

TODAY it is difficult to imagine the task of system development without help of a simulation. Simulation is a very important means in all fields of science and engineering. There are a number of reasons for this fact, as there are saving of development time, saving of costs, better understanding of the function, testing and finding of critical states and regions of operation, fast optimization of system and control [8, 9]. The many common modules found in various fusion algorithms sometimes force developers to repeat programming efforts in order to improve existing systems. As more data fusion algorithms are developed [1], the above problems become more serious. These problems call for an alternative way of developing data fusion methods and systems in software. This paper presents some results of such a development, SDF (Sensor/Data Fusion) Toolbox for Simulink, which offers the users a flexible and efficient software tool for studying and designing fusion structures. In this study, the SDFTool is expressed as a design pattern.

The origin of design patterns lies in work done by an architect named Christopher Alexander during the late 1970s. He began by writing two books, A Pattern Language [11] and A Timeless Way of Building [12], which in addition to giving examples, described his rationale for documenting patterns. The pattern movement became very quiet until 1987 when patterns appeared again at an OOPSLA conference. Since then, many papers and presentations have appeared, authored by people such as Grady Booch, Richard Helm, and Erich Gamma, and Kent Beck [13]. From then until 1995, many periodicals, featured articles directly or indirectly relating to patterns. In 1995, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides published Design Patterns: Elements of Reusable Object-Oriented Software [14], which has been followed by more articles in trade journals.

The SDF toolbox has a number of features, which make it an effective research and educational tool. Those features take care of the purely software development such that the programming efforts are very much reduced. The rest of this paper is organized as follow: Section 2 describes the SDF toolbox pattern subsections such as features, implementation, relationship between this SDFTool and virtual test bed and an example on the application of SDF Toolbox. Finally the conclusion and further works is given.

II. SDFTool as a Design Pattern

In this part, the SDFTool is illustrated as a design pattern. Design patterns are recurring solutions to design problems you find again and again in real-world application development. Design patterns are about design and interaction of systems, as well as providing a communication platform concerning elegant, reusable solutions to commonly encountered system development challenges. The Gang of Four (GOF) patterns are generally considered the foundation for all other patterns. They are categorized in three groups: Creational, Structural, and Behavioral. The SDFTool is categorized in creational group. The sections of this fusion pattern are described as follow:

A. Intent

The major intention of this toolbox is the development of a high-level graphical system modeling using the powerful modeling environments.

B. Aliases

SDFTool, FusionTool

C. Motivations

In its present state, fusion system analysis and design using software simulation has its own problems. Valuable time and energy is often wasted on writing and debugging codes rather than developing theoretical algorithms, and attempting to translate independently written programs can be even more difficult. Lack of standardization also means that direct comparison of algorithms and filters are always difficult or even possible. The many common modules found in various fusion algorithms sometimes force developers to repeat programming efforts in order to improve existing systems. As more and more advanced data fusion algorithms are developed [1], the above problems become more and more serious. These problems call for an alternative way of developing data fusion methods in software environments.

D. *Applicability*

The practical applications of data fusion have necessarily been seen in areas in which the required output of an analysis may not be measured directly. This is particularly important in medical imaging [15], position estimation [16] and remote sensing such as target identification and tracking [17, 18]. The methods are particularly popular in Condition Monitoring, where the purpose is to detect faults and the degradation of machine health [19].

E. *Structure*

The blockset that makes up the first version of the toolbox include over 10 functional blocks each performing specific tasks related to data fusion. Furthermore, since the data structures used by all blocks are well defined and standardized, the blocks can be replaced in or added to designs without the need for scenario-based modifications. There are three standard alternatives for data fusion, which are data level fusion, feature level fusion and decision level fusion. In all these levels, Information might be in discrete or continuous form. In higher levels of fusion, discrete or label output fusion will be more important.

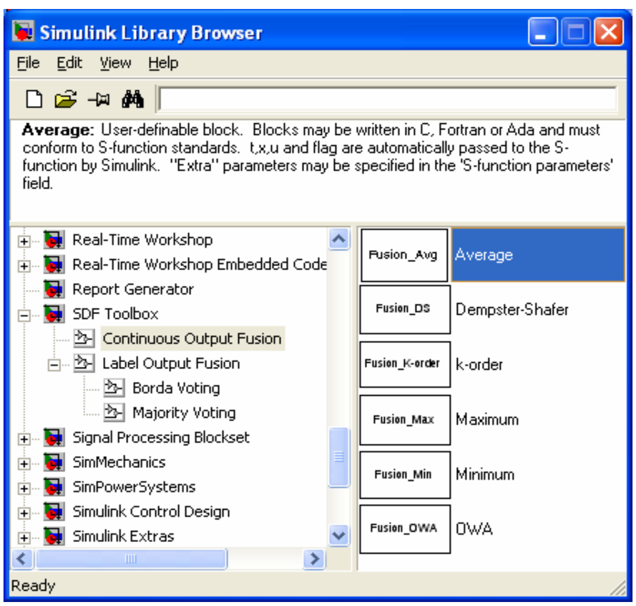

Fig. 1: schematic of SDF Toolbox installed in Simulink Library browser

F. *Participants*

Regarding the different use of fusion methods, the SDFTool is divided in two major subsystems:

1) *Continuous output fusion subsystem*

In first version of this toolbox, the continuous output fusion subsystem consists of six common blocks, which are Demster-shafer [5], Ordered Weighted Averaging (OWA), simple average, maximum, minimum, and k-order. The OWA block is implemented with two standard weighting approaches that are optimistic and pessimistic. This operator is first introduced by Yager [6]. In this block the maxness value and the type of weighting method is given by a simple user interface assigned to this block.

2) *Label output fusion subsystem*

In label-output fusion subsystem, we developed two general types of voting systems, which are Majority and Borda voting [7]. In Majority voting blockset a simple Majority, unanimity, and plural are posed. In Borda voting, simple Borda count and weighted Borda voting is implemented.

G. *Collaborations*

The collaboration of SDFTool can be considered as intra collaboration and extra collaboration. The extra collaboration declares the collaboration among different simulation environments whereas intra collaboration defines the collaboration among different blocks and subsections.

H. *Other simulation environments: Virtual Test Bed*

The Virtual Test Bed project is dedicated to developing a new environment for simulation and virtual prototyping of large-scale, multi-technical dynamic systems. Within the context of virtual prototyping, we include not only the simulation of system dynamics, but also the solid modeling of the system and visualization of the system dynamics [20]. Here, we use specifically the interactive link (wrapper) between VTB and Matlab/Simulink that is especially valuable during the process of designing control algorithms for other systems.

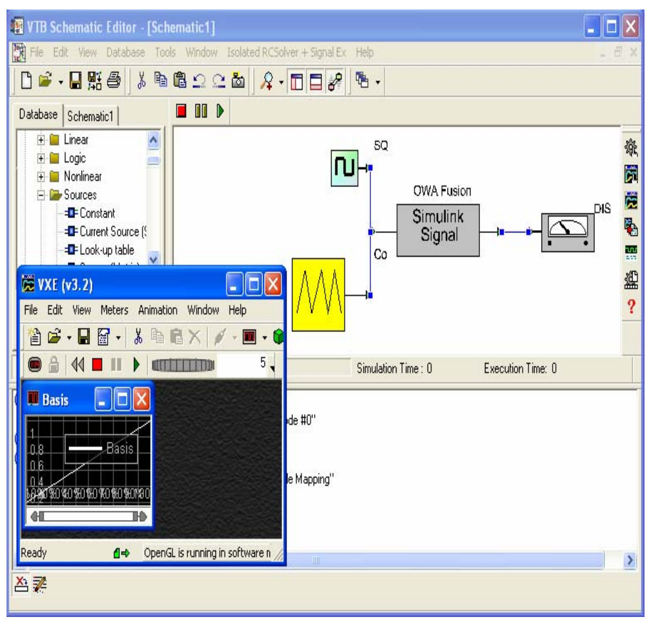

Figure 2: SDFTool in virtual test bed

I. Inter relationship

In SDFTool, each block has one output, an input vector, and a set of parameters. Using implemented data fusion algorithms, each block converts the input vector to an output. In multi layer architecture, each block's output could be an input of other block. In this circumstance, different levels of fusion could be easily implemented and tested via simulation.

J. Consequences

The SDFTool provides a convenient and intuitive framework for designing, developing, and applying data fusion algorithms. It can substantially reduce the amount of time and energy needed for algorithm implementation.

K. Implementation

The toolbox was developed on the graphic user interface design features of the Simulink platform using Matlab S-function technique [2, 3]. Simulink is widely used in various fields of science and engineering. The graphically based software breaks away from the traditional methods of formulating mathematical model in computer code and allows system design by connecting modular blocks. The interface of SDF Toolbox allows parameter setting with maximum flexibility. In the case where many parameters are defined for a single block, intuitive GUI functions are designed so that the user may keep track of the exact operations intended. In addition, Data ranges and types have been defined for all the block parameters, and prevent users from defining non-sensible parameters for block algorithms.

L. Sample Code - Meta classifier design

In this section, an example on the application of SDF Toolbox is briefly discussed. The following block diagram presents the Meta classifier in protein secondary structure developed by authors [4].

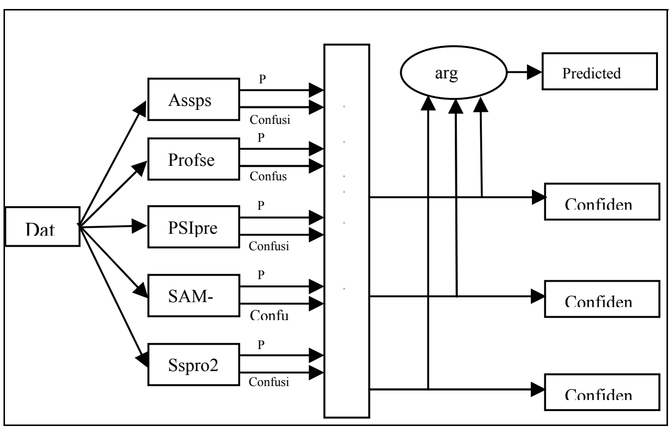

Figure 3: Meta classifier schema-(PC: predicted class)

By selecting relevant blocks in SDF Toolbox, Meta classifier algorithm can fairly be realized in a similar structure.

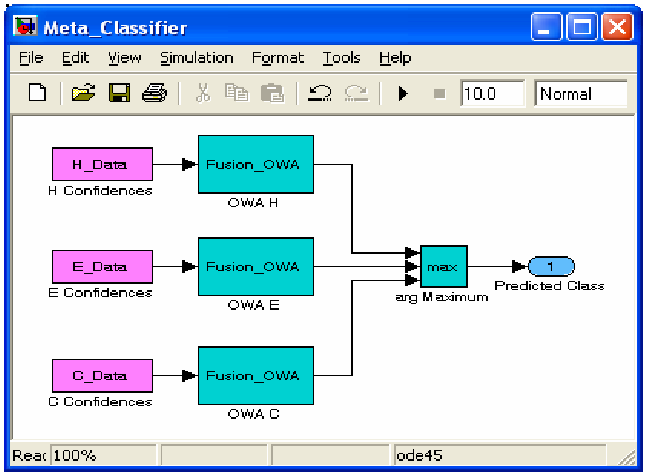

Figure 4: Meta classifier model simulation using SDF Toolbox

M. Known Uses

Data fusion is very wide in the sense of applications. Some of these applications and convenient uses are in robotics, defense systems, medical engineering, remote sensing, internet, agriculture, process control, monitoring, biological systems, and bioinformatics.

N. Related Patterns

All patterns that can be transfer to the other simulation environments such as VTB, Simulink, and Paragon, are suitable for integrating with SDFTool pattern to model the large-scale systems including control, power, electronic and other disciplines.

III. Conclusions and Future Works

The pattern introduced in this paper, SDFTool, provides a convenient and intuitive framework for designing and developing data fusion algorithms. It is suitable for researchers' new development at different levels and can substantially reduce the amount of time needed for algorithm implementation and avoid undetectable human errors. This toolbox could eventually bring a new and improved modeling and simulation standard for the wider information fusion community.

There are still open issues ahead:

- extending the algorithms and blocks of fusion
- developing some standard models for data fusion
- integrating this pattern with those in software
- achieving the hardware design from the highest level of design (e.g. UML).

References

[1] B. Vo and S. Challa, "Multi-Sensor Data Fusion For Target Tracking in Clutter", FUSION2001, Montreal, Canada, August, 2001.

[2] The MathsWorks Inc, Simulink – Dynamic System Simulation for MATLAB: Using Simulink, MA, USA, 2000.

[3] The MathsWorks Inc., Simulink – Dynamic System Simulation for MATLAB: Writing S-Functions, MA, USA, 2000.

[4] Majid Kazemian, Behzad Moshiri, Hamid Nikbakht, Caro Lucas. "*Protein Secondary Structure Classifiers Fusion using OWA*", Lecture

Notes in Computer Science. Springer-Verlag Berlin Heidelberg 3745 - pp. 338 -345, 2005.

[5] L. Xu, A. Krzyzak, and C.Y. Suen, “Methods of Combining Multiple Classifiers and their Applications to Handwriting Recognition”, IEEE Trans. SMC, vol. 22, No. 3, pp 418-435, 1992.

[6] Yager, R. R., On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE transactions on Systems, Man and Cybernetics 18, pp. 183-190, 1988.

[7] H. de Swart, A. Deemen, E. Hout, P. Kop, Categoric and Ordinal Voting: An Overview. Eds. Theory and Applications of Relational Structures as Knowledge Instruments. ISBN 3-540-20780-5, no. 2929 in Lect. Notes in Comput. Sci., Springer-Verlag, pp. 147 – 195, 2004.

[8] Jingzhao Ou and Viktor K. Prasanna, "PyGen: A MATLAB/Simulink based Tool for Synthesizing Parameterized and Energy Efficient Designs Using FPGAs," IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM), 2004.

[9] Tu Le, Gregory Donohoe, David Buehler and Pen-Shu Yeh, “Graphical Design Environment for Reconfigurable Processor”, MAPLD International Conference, Washington, D.C, 2004.

[10] Dougal, T. Lovett, A. Monti, E. Santi, “A multilanguage environment for interactive simulation and development of controls for power electronics,” Proc. IEEE Power Electronics Specialists Conference (PESC'01), Vancouver, Canada, pp. 1725-1729, 2001.

[11] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, S. Angel, “A PATTERN LANGUAGE”, Oxford University Press, New York, 1977.

[12] C. Alexander, “THE TIMELESS WAY OF BUILDING”, Oxford University Press, New York, 1979.

[13] Booch, Grady, “Object-Oriented Analysis and Design with Applications”, ISBN 0-8053-5340-2, Second Edition, 1995.

[14] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, (Gang of Four) “Design Patterns, Elements of Reusable Object-Oriented Software”, 1995.

[15] R. HoseinNezhad, B. Moshiri, M. R. Asharif, "Improved pose estimation for mobile robots by fusion of odometry data and environment map", Journal of Intelligent and Robotic Systems, Vol. 36, No. 1, 89-108, January 2003.

[16] K.Farzan, B. Moshiri, H.Soltanian-Zadeh, “A Fuzzy Integral Approach for Data Fusion in MRI”, ICBME_ 96, Eighth Iranian Conf-on Biomedical Engineering, Dec. 19-23, 1996.

[17] B. Moshiri, F.Besharati, "Maneuvering Target Tracking", WSEAS Transactions on Circuit and Systems, Issue 1, Volume 3, 176-182, January 2004.

[18] Farhad Samadzadegan , B. Moshiri , Hassan Bagherzadeh, " A comprehensive evaluation of spectral and intensity potential of data fusion techniques in Remote Sensing ", Geomatics 82, Tehran, IRAN, 2003.

[19] Lou K N, Lin C J, Intelligent sensor fusion system for tool monitoring on a machining centre, International Journal of Advanced Manufacturing Technology, Vol.13, No.8, pp.556-565, pub. Springer, ISSN: 0268-3768, 1997.

[20] R. Dougal, T. Lovett, A. Monti, E. Santi, “A Multilanguage Environment For Interactive Simulation And Development Of Controls For Power Electronics”, IEEE PESC01, Vancouver (Canada), 2001.