Modeling Networks-on-Chip at System Level with the MARTE UML profile

Majdi Elhaji, Pierre Boulet, Rached Tourki, Abdelkrim Zitouni, Jean-Luc Dekeyser, Samy Meftali

Modeling Networks-on-Chip at System Level with the MARTE UML profile

Majdi Elhaji
Abdelkrim Zitouni
and Rached Tourki
Laboratory of Electronic
and Micro-Electronic (LAB-IT06)
Monastir, Tunisia

Pierre Boulet
Jean-Luc Dekeyser
and Samy Meftali
Univ. Lille Nord de France, F-59650 Villeneuve d'Ascq, France
Univ. Lille 1, LIFL, F-59650 Villeneuve d'Ascq, France,
CNRS, UMR 8022, F-59650 Villeneuve d'Ascq, France,
INRIA Lille - Nord Europe, F-59650 Villeneuve d'Ascq, France,
Email : Firstname.Lastname@lifl.fr

***Abstract*—The study of Networks on Chips (NoCs) is a research field that primarily addresses the global communication in Systems-on-Chip (SoCs). The selected topology and the routing algorithm play a prime role in the performance of NoC architectures. In order to handle the design complexity and meet the tight time-to-market constraints, it is important to automate most of these NoC design phases. The extension of the UML language called UML profile for MARTE (Modeling and Analysis of Real-Time and Embedded systems) specifies some concepts for model-based design and analysis of real time and embedded systems. This paper presents a MARTE based methodology for modeling concepts of NoC based architectures. It aims at improving the effectiveness of the MARTE standard by clarifying some notations and extending some definitions in the standard, in order to be able to model complex architectures like NoCs.**

I. Introduction

System-on-chip (SoC) designs supply integrated solutions to challenging design problems of embedded systems and consumer electronic domains. Much of the progress in these fields allows the designer to conceive more and more complex electronic systems and reduce time to market. Future systems-on-chips will contain tens to hundreds of IP cores according to the International Technology Roadmap for Semiconductors [1]. The on-chip communication paradigm is introduced as Network-on-Chip (NoC) by Benini and De Micheli [2], Dally and Towles [3] and Sgroi et al [4]. Many propositions of NoC architectures for SoCs design have been presented in literature, such as, SPIN [5], HERMES [6] XPIPES [7], OCTAGON [8], Mesh [9], honeycomb [10].

Topology and routing algorithm are the two most important aspects that distinguish the various NoC architectures. The choice of topology is very important to provide a high level of performance. On the other hand routing on NoCs has the same principle as routing on any network. Consequently, the system becomes more and more complex, from transistors to layout. The system designer needs with a level of abstraction that focuses more on system functionality rather than low level design details and there is a demand for the development of high level modeling environments for computer-aided design.

The modeling of highly repetitive structures such as network on chip topologies in graphical form poses a particular challenge. Model driven engineering is a software development methodology where the complete system is modeled at a high abstraction level using a modeling language as UML. The field of real-time embedded software systems is a domain for which extensions to UML are required to provide more precise notations of domain specific phenomena. The UML profile for Modeling and Analysis of Real-Time Embedded systems, MARTE [11] is the current standard for the SoC domain. We study in this paper how well it is suited to model complex NoC architectures.

We note that there is a few works interested in modeling concepts of NoC. Only the approaches defined in [12] [13] come close to our contribution by using the MDE concepts for modeling repetitive structure such as multi-processor system on chip (MPSoC) but not NoC concepts. In [12] the authors show the need for modeling the distribution of a parallel application onto parallel hardware architecture and describe the notation (MARTE) for modeling only regular architecture. This notation allows distributing computations to processing elements, data to shared or distributed memories, with the aim of clarifying its usage through MPSoC examples and the comparisons to other distribution notations such as in High Performance Fortran. We note that in literature there is a little work interested in modeling NoC with MARTE, defining methodology or providing the effectiveness for notation to model all concepts of NoC such as topology, routing algorithm, switching mode and communication protocol. The aim of this paper consists in proposing a novel methodology for modeling all family of NoC topologies and show that the advanced MARTE can support the modeling of routing algorithm with the use of state machines at the end to propose an extension in the HwCommunication package.

II. Network on Chip Characteristics

The NoC is a replacement for global interconnect and single bus architectures. The main immediate benefit of a NoC based

This work has been supported by the French state/region project contract Campus Intelligence Ambiante

approach is clearly due to the possibility to reuse the communication network throughout different products. Furthermore, as the complexity of integrated systems keeps growing, a NoC provides enhanced performance such as throughput and scalability in comparison with the bus communication architectures. Thus it is useful to propose a methodology for modeling NoC topologies with the aims of being in line with the evolution of semiconductor technology and reduction of the time to market. Also, from a predictability perspective, the regularity of NoC layout provides well characterized electrical and physical properties. The topology and routing algorithm are the most important aspects which distinguish the diverse NoC architectures.

A. Topologies

The topology refers to the physical structure of the network graph, i.e., how network nodes are physically connected. It defines the connectivity or the routing possibility between nodes, thus having a fundamental impact on the network performance as well as the router structure (number of ports and port width). The trade-off between generality and customization is an important issue when determining a network topology. Each topology can be characterized by a few properties. The degree of a router is the number of links connecting that node to its neighbor vertices. A topology is considered as regular when all routers have the same degree, if not it is considered as irregular. In [14], the number of ports in routers can be synthesized according to the requirement of connectivity. However, the area and power consumption of an irregular network topology may not scale predictably with the topology size. Besides, many network topologies have been proposed where most of them are proposed for minimizing the number of nodes and node degrees. Also there are many research activities for designing NoC topologies. Murali in [15] have developed a tool for automatically selecting an application-specific topology for minimizing average communication delay, area, and power dissipation. Other topologies between regular and irregular are also proposed for NoCs. For example, an interesting NoC topology is the Spirdegon of STMicroelectronics, GEXspedirgon NoC [16] and Honeycomb [10]. Spidergon, is one of the NoCs researched at STMicroelectronics and is being proposed as an evolution of the STNoC [17]. It is a packet-switched NoC, inspired from the Octagon NoC [8], deterministic routing, wormhole switching and TDMA quality of service QoS. In [10], Hemani and all present a honeycomb NoC topology and its associated methodology as solution to the design productivity problem. Besides they propose a platform to handle the complexity of emerging design of chip architecture and still allow companies to meet the time to market and make profit, but implementations and results are not presented. Another study was made in order to define a generic NoC [18] named GeNoC that is intended to serve as a reference for the design and the validation of high level specifications of communicating virtual modules, but this study is limited to topology and routing concepts. The 2D-mesh is the most used topology due to its simplicity. It consists of horizontal and vertical lines with nodes placed at their intersections. This specific structure is often used because inter-node delays can be predicted at a high level.

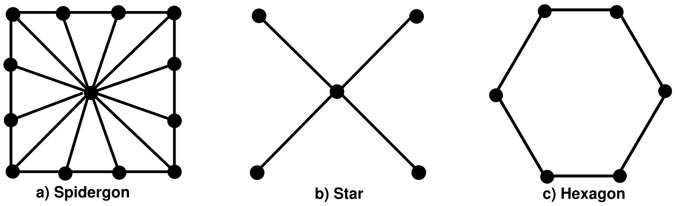

Fig. 1. Examples of standard topologies

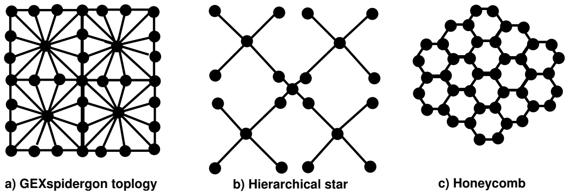

Fig. 2. Examples of hierarchical topologies

Figures 1 and 2 show some of topologies that have been proposed in the literature.

B. Routing algorithm

Routing algorithms define the path followed by each message or packet routing on an NoC. A routing algorithm defines how the data are transmitted from sender to receiver. The choice of a routing algorithm depends on several metrics such as minimizing power, minimizing logic and routing tables to achieve a lower area, increasing performance by reducing delay and maximizing traffic utilization of the network. A lot of works aim at improving the hardware block for the routing algorithm in the NoC.

In this paper, routing algorithms are divided into two groups, deterministic and adaptive algorithms. Static routing algorithms ignore the network path diversity and are not sensitive to the network state. They are also simple and inexpensive to implement. Besides, it is often a simple way to provide the ordering of packets. Static routing also permits packets to be split among multiple paths between a source and destination, in a predetermined way. If only a single path is used, static routing usually guarantees in-order delivery of data packets. This eliminates the need for adding bits to packets at the NI (Network Interface), in order to correctly identify and reorder them at the destination.

Adaptive routing algorithms, also named dynamic algorithms, use information about network traffic and/or channel status to avoid congested or faulty regions of the network. In dynamic routing, routing decisions are made according to the current state of the network, considering a load on links. As such it is possible that the path between the source and destination changes over time, as traffic conditions and requirements of the application change. This adaptive behavior requires additional hardware resources that control the state of the network and routing paths. Besides, dynamic routing is able to utilize alternate paths when some directions become congested.

Routing algorithms can be implemented in various ways. The most interesting ways consist of either looking at a routing table or executing a finite-state machine in software or hardware. In this paper we focus on the modeling of routing algorithms as a state machine to demonstrate the ability to model the routing on NoCs in MARTE.

III. Proposed methodology

Designing an efficient NoC architecture, while satisfying the application performance constraints, is a complex process. The design issues require several abstraction levels, ranging from high-level application modeling to physical layout level implementation. Some of the most important phases in designing the NoC include: NoC topology modeling for the application, choosing the routing algorithm, communication protocol and mode switching.

This methodology can be expressed in two phases. In the first phase, we begin by detailed concepts relating to a NoC, such as mathematical studies of topology and routing algorithm. Second, specify the package we will use and make the relation between hardware architecture concepts and notation that also exist in MARTE. We illustrate this below with the two main characteristics of NoCs identified above: topology and routing algorithm.

A. Topology Modeling

Several topologies, such as hypercube, star, ring and tree, can be modeled easily with the MARTE profile using the RSM (Repetitive Structure Modeling) package (see [11], annex E, pages 517–533). The more complex such as the GEXspidergon and honeycomb are not as easy but we will demonstrate below that this RSM package is powerful enough to model them. In order to model these topologies in a compact way, one should make a mathematical study of the topology to identify some information of the graph such as degree, valence, the number of links, the number of input/output ports, then classify routers by their numbers of port and factorize routers if possible.

The modeling consists in using a class (and part) by router kind, express the number of routers using the « shaped » stereotype and connect them which links stereotyped with a link topology, either « interRepetition » for the simple cases or « reshape » for the more complex kinds.

We give below two examples of complex topology modeling.

1) Modeling of honeycomb Topology: The honeycomb mesh (see Figure 2c), based on hexagonal plan tessellation network. Hexagonal tessellations were used in literature for various applications such as cellular phone. Hexagonal and honeycomb mesh are dual graph one can obtain the honeycomb mesh by joining the center of each triangle in hexagonal mesh with center of neighboring triangle. One hexagon is a honeycomb mesh of size 1, the honeycomb mesh of size 2 is obtained by adding 6 hexagons to the boundary edge of hexagon size 1. A honeycomb mesh size m is obtained from honeycomb size $m-1$. the number of nodes of the honeycomb mesh of size t is $6m^2$, and the number of links is $9m^2-3m$, the diameter of honeycomb mesh of size t is $4m-1$. The Honeycomb Architecture is constructed based on an elementary hexagonal network that can be described, in parallaxis syntax as following:

Configuration Hex [1....2], [1...3];
Connection:
East: $Hex[i,j] \rightarrow Hex[i,j+1]$
West: $Hex[i,j] \rightarrow Hex[i,j-1]$
North: $Hex[i,j] \rightarrow Hex[i-1,j]$
South: $Hex[i,j] \rightarrow Hex[i,j-1]$

The Honeycomb graph is constructed by multiplying this algorithm. This architecture is modeled on the same following algorithm to model GEXspidergon; in a customizable way on the environment GASPARD by using the RSM package, in order to take into account the modeling of link topologies. The concerned topology is composed of the repetition of a single element (Router or port). Each potential instance of this element is connected to other potential instances of the same element. In our case each instance is connected to neighbors located at north, south, east and west. The inter Repetition topology enables to specify the position of every neighbor of every potential instance of a model element with a multidimensional shape. With this architecture it is to the best of our knowledge the first time that a reshape connector is used with the same port as source and destination, allowing to represent regular but not uniform dependencies between repetitions of a part.

Figure 3 shows the modeling of honeycomb topology.

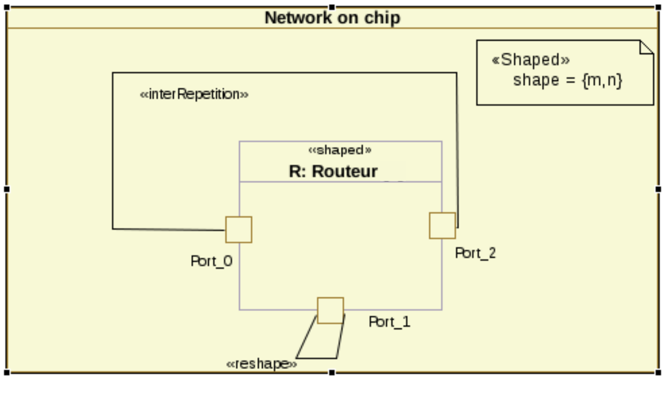

Fig. 3. Modeling of a honeycomb topology

Mathematical models for Reshape and InterRepetition are described below:

Inter Repetition={1,0}
Reshape include Tiler source and Tiler target:
Pattern shape={0}, Repetition space={0}
Source Tiler
origin=$\begin{pmatrix}0\\0\end{pmatrix}$ paving=$\begin{pmatrix}2 & 1\\0 & 1\end{pmatrix}$ fitting=$\begin{pmatrix}0\end{pmatrix}$

Target Tiler
origin=$\begin{pmatrix}0\\0\end{pmatrix}$ paving=$\begin{pmatrix}2 & 1\\0 & 1\end{pmatrix}$ fitting=$\begin{pmatrix}0\end{pmatrix}$

2) Modeling of GEXspidergon Topology: The GEXspidergon (see Figure 2a) can be seen as a hierarchical and globally regular, locally regular topology. It is an academic topology for NoC. This study presents a generic NoC architecture based on a configurable router. This router integrates a sophisticated dynamic arbiter, the wormhole routing technique and can be configured in a manner that allows it to be used in many possible NoC topologies such as Mesh 2-D, Tree and Polygon architectures. This makes it possible to improve the quality of service (QoS) required by the proposed NoC. This study [17] shows that the Spidergon architecture is characterized by the lower latency and saturation. This architecture is constructed based on an elementary polygon network which is a combination of the star and the ring architectures. This elementary network is formed by $4R+1$ ($R = 1, 2$, etc.) routers including a central router that is connected with the $4R$ peripheral routers via point to point links. The peripheral routers are connected to each other in the form of a ring. The elementary network is thus characterized by its valence ($m = 4R$) that represents the number of the peripheral routers. These routers necessitate $2m$ links to be connected to the central router. Each peripheral router is connected to 4 input/output ports and the central router is connected to $m+1$ input/output ports. This elementary network can be described in Parallaxis[19] as following:

Configuration Poly [2...n], [2...m]

Periphery connection:

Vertical connection:

North: Poly$[i] \to$ *Poly*$[i-1]$

South: Poly$[i] \to$ *Poly*$[i+1]$

Horizontal connection:

East: Poly$[j] \to$ *Poly*$[j+1]$

West: Poly$[j] \to$ *Poly*$[j-1]$

Central Router:

Connection many to one

Poly$[i,j] \to$ *Poly*$[0,0]$

The GEXspidergon graph is constructed by iterating this algorithm in two dimensions as illustrated in Figure 2. Routers are categorized into four groups according to their degree:

- Corner Routers.
- Middle Routers.
- Horizontal Routers, package of two.
- Vertical Routers, package of two.

Figure 4 shows a possible model of this topology.

To express the link topology between vertical/horizontal routers and middle routers there was some problem since the considered tiles are not sets of regularly spaced point and in MARTE/RSM the points of the tile must be regularly spaced (as they are built from the reference point of the tile by the linear combination of the column vectors of the fitting matrix). We had to use two reshape links to express such a topology. These two reshapes are detailed below.

Reshape1: VR to MR

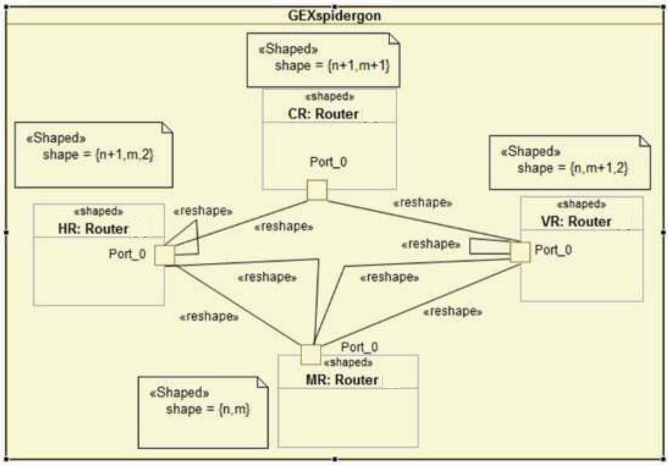

Fig. 4. Modeling of a Gexspidergon topology

Repetition Space={n, m}
Pattern Shape={2}
Source Tiler:
origin=$\begin{pmatrix}0\\0\\0\\2\end{pmatrix}$ paving=$\begin{pmatrix}1 & 0\\0 & 1\\0 & 0\\0 & 0\end{pmatrix}$ fitting=$\begin{pmatrix}0\\0\\1\\0\end{pmatrix}$
Target Tiler:
origin=$\begin{pmatrix}0\\0\\0\end{pmatrix}$ paving=$\begin{pmatrix}1 & 0\\0 & 1\\0 & 0\end{pmatrix}$ fitting=$\begin{pmatrix}0\\0\\1\end{pmatrix}$

Reshape2: VR to MR

Repetition Space={n, m}
Pattern Shape={2}
Source Tiler:
origin=$\begin{pmatrix}0\\1\\0\\0\end{pmatrix}$ paving=$\begin{pmatrix}1 & 0\\0 & 1\\0 & 0\\0 & 0\end{pmatrix}$ fitting=$\begin{pmatrix}0\\0\\1\\0\end{pmatrix}$
Target Tiler:
origin=$\begin{pmatrix}0\\0\\5\end{pmatrix}$ paving=$\begin{pmatrix}1 & 0\\0 & 1\\0 & 0\end{pmatrix}$ fitting=$\begin{pmatrix}0\\0\\1\end{pmatrix}$

With this case study we have shown that we can model the complex regular topologies with MARTE/RSM in a compact way, exploiting fully the regularity of the architecture of the NoC.

B. Modeling Routing Algorithms

In this paper routing algorithms, are divided into two groups, deterministic and adaptive algorithms. We are interested in modeling an example for each family.

1) Deterministic XY algorithm: The XY routing is considered as a kind of deterministic routing algorithms. It can be used in a 3×3 Mesh topology NoC. In this architecture each router is identified by its coordinates (x, y).

This algorithm begins by comparing the current router address (C_x, C_y) to the destination router address (D_x, D_y) of the data, stored in the header. Packet must attain the port of the router when (C_x, C_y) is equal to (D_x, D_y) address. If not, D_x is compared to Cx address. Packet will be routed to the west port when $C_x > D_x$, to east when $C_x < D_x$ and if $C_x \cong D_x$ the packet is horizontally routed. When the last condition is true, the D_y address is compared to the C_y. When $C_y > D_y$ the packet is routed to the North but if $C_y < D_y$ it will be routed to the South. Packet will be blocked in the Fifo when the chosen port is busy. This explication can be described with the following algorithm.

```
Destination router: $(D_x, D_y)$
Current router: $(C_x, C_y)$
If $(D_x > C_x)$ // Move East
Return E
Else if $(D_x < C_x)$ // Move West
Return w
Else if $(D_x = C_x)$ // same direction
If $(D_y < C_y)$ // Move South
Return S
Else if $(D_y > C_y)$ // Move North
Return N
Else if $(D_y = C_y)$ //same router
Return C
```

Thus the routing algorithm can be modeled via MARTE profile as a hardware component where the behavior is described with a behavioral state machine in MARTE as shown in Figure 5.

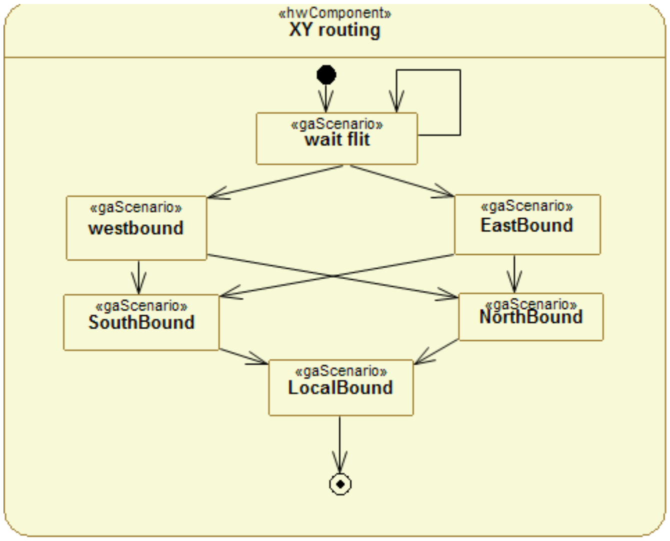

Fig. 5. A high level abstraction of XY routing via UML/MARTE profile

2) Adaptive routing: In adaptive routing, paths are dynamically changed during the routing process according to network conditions, such as the presence of congestion. If a channel is busy, the other channel has priority over the busy channel. However, if both output channels are not used, a cost function (CF) decides which output channel will be used. This cost function is required when an adaptive routing is implemented. The adaptive routing algorithm increases the complexity of the hardware architecture of the router. That is why it will be significant to model this algorithm at a high level of abstraction. In this case a typical example of adaptive routing algorithm is modeled, the double Y-channel routing algorithm. This algorithm applies to a 2D mesh where router are connected by one bidirectional links in the X dimension and by two bidirectional links in the Y dimension. The network is divided into two sub-networks. The X1 sub-network uses paths in the ascending X dimension. The X2 sub-network uses paths in the descending X dimension.

We consider S and D the source and destination router of a packet. Their coordinates along the X axis are noted S_x and D_x. If, at some router the destination of a packet is on the right the packet uses the X1 sub-network; if the destination is on the left the packet used the X2 sub-network. When $D_x = S_x$ a cost function decides the channel to use. In each sub-network, several minimal paths are possible. A packet crosses the network through a single sub-network. This explication can be described with the following algorithm.

```
Destination router: $(D_x, D_y)$
Source router: $(S_x, S_y)$
If $(D_x > S_x)$ // Move in X1
Return X1
Else if $(D_x < S_x)$ //Move in X2
Return X2
Else if $(D_x = C_x)$ // CF
Return CF
```

This algorithm can be simply modeled in state machine via UML/MARTE profile as shown in Figure 6.

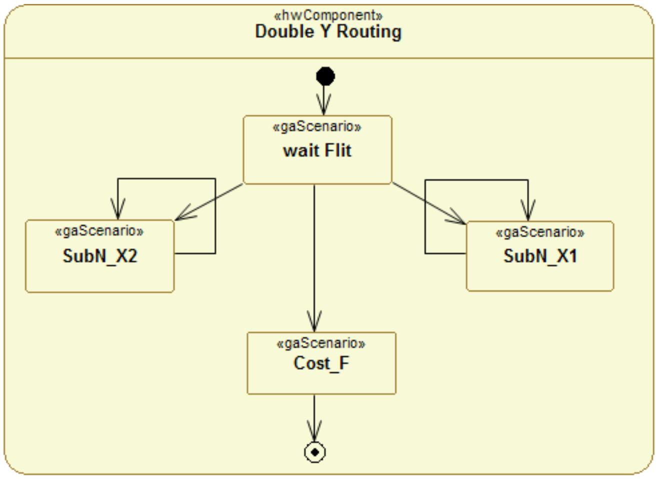

Fig. 6. A high level abstraction of adaptive routing via UML/MARTE profile

For the QoS there are a lot of work that integrate several services in the routing algorithm such as a dynamic arbiter, guaranteed throughput and TDMA. The designer can consider all these concepts in the modeling. On the other side, the switching technique can be an enumerated type in the MARTE profile.

The last thing that we need to model NoCs at a high level of abstraction is to decide which modeling artifact we will use to model a router. Indeed, this element is the basic building block of NoCs and with which the topology can be described and that should be characterized by a behavioral state machine describing the routing algorithm. It should thus be a structured class with an adequate stereotype.

The HwCommunication package in the MARTE profile defines some concepts for modeling the communication infrastructure of embedded systems as « HwEndPoint » which may present the network interface of the NoC, « HwBridge » to make connection between resources and the « HwArbiter » to control the communication. But in this package the router (the basic building of NoCs as identified above) is not found. That is why we propose to add the « HwRouter » stereotype as a specialization of the « HwMedia » stereotype in the HwCommunication package. This addition would enable the modeling of NoCs at an abstraction level useful for specifying and analyzing the communication possibilities of the current (and probably future) networks used in recent systems-on-chip.

IV. Conclusion

We have proposed in this paper a methodology for modeling NoCs with the MARTE standard profile. We have identified two main characteristics of NoCs that are useful to design, analyse and understand the structure and behavior of NoCs: the topology of the network and the routing algorithm.

Several topologies have been proposed in the literature and we have shown that the Repetitive Structure Modeling package of MARTE is powerful enough to model as well the simple regular ones as the more complex ones. To achieve this, we have proposed the first use of the « Reshape » connector to connect some ports of the same part, enabling the description of regular but non uniform connections between repeated parts. The proposed modeling methodology enables to take full advantage of the regularity in the topology to factorize the model. Such factorization could then be used to generate factorized analysis or synthesis code and thus tackle large networks (think many-cores with several hundreds or thousands of routers) as easily are small ones.

The second characteristics that we have identified is the routing algorithm. We have proposed to model it with a standard UML behavioral state machine. This state machine should be attached to a component stereotyped « HwRouter », a new stereotype that we propose to identify the basic building block of NoCs. With this minor addition, the MARTE profile is complete enough to model a very large number of NoC proposals (actually, all those that we have found), especially those that exhibit some regularity enabling a scaling of the network. Some work is on-going to synthesize such networks in VHDL from such models.

References

[1] International Technology Roadmap for Semiconductors, "ITRS 2009," 2009. [Online]. Available: http://www.itrs.net/Links/2009ITRS/Home2009.htm

[2] L. Benini and G. D. Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=976921

[3] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 684 – 689.

[4] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, and A. Sangiovanni-Vincentelli, "Addressing the system-on-a-chip interconnect woes through communication-based design," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 667 – 672.

[5] U. Y. Ogras and R. Marculescu, "Application-specific network-on-chip architecture customization via long-range link insertion," in *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*, Nov. 2005, pp. 246 – 253.

[6] F. Moraes, "HERMES: an infrastructure for low area overhead packet-switching networks on chip," *Integration, the VLSI Journal*, vol. 38, no. 1, pp. 69–93, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=1056486

[7] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini, "Xpipes: a latency insensitive parameterized network-on-chip architecture for multiprocessor SoCs," in *Computer Design, 2003. Proceedings. 21st International Conference on*, 2003, pp. 536–539.

[8] F. Karim, A. Nguyen, S. Dey, and R. Rao, "On-chip communication architecture for OC-768 network processors," in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 678–683.

[9] R. Holsmark and S. Kumar, "Design issues and performance evaluation of mesh NoC with regions," in *NORCHIP Conference, 2005. 23rd*, 2005, pp. 40–43.

[10] A. Hemani, A. Jantsch, S. Kumar, A. Postula, M. Millberg, and D. Lindqvist, "Network on chip: An architecture for billion transistor era," in *Proceedings of the IEEE NorChip Conference*, Nov. 2000. [Online]. Available: http://ihome.ust.hk/~ldcse/CSIT560_papers/NOC/norchip-noc.pdf

[11] Object Management Group, "UML profile for MARTE: modeling and analysis of Real-Time embedded systems, version 1.0," Nov. 2009, formal/2009-11-02. [Online]. Available: http://www.omg.org/spec/MARTE/1.0

[12] P. Boulet, P. Marquet, E. Piel, and J. Taillard, "Repetitive allocation modeling with MARTE," in *Proceedings of the Forum on Specification & Design Languages*, Barcelona, Spain, Sep. 2007.

[13] A. Cuccuru, J. Dekeyser, P. Marquet, and P. Boulet, "Towards UML 2 extensions for compact modeling of regular complex topologies," in *Model Driven Engineering Languages and Systems*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3713, pp. 445–459, 10.1007/11557432_34. [Online]. Available: http://dx.doi.org/10.1007/11557432_34

[14] K. Srinivasan, K. Chatha, and G. Konjevod, "Linear-programming-based techniques for synthesis of network-on-chip architectures," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 407–420, 2006.

[15] S. Murali and G. D. Micheli, "SUNMAP: a tool for automatic topology selection and generation for NoCs," in *Design Automation Conference*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 914–919.

[16] M. Zid, A. Zitouni, A. Baganne, and R. Tourki, "Nouvelles architectures génériques de NoC= news generics architectures for NoC," *TSI. Technique et science informatiques*, vol. 28, no. 1, p. 101–133, 2009.

[17] M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra, "Spidergon: a novel on-chip communication network," in *System-on-Chip, 2004. Proceedings. 2004 International Symposium on*, 2004, p. 15.

[18] J. Schmaltz and D. Borrione, "A generic network on chip model," in *Theorem Proving in Higher Order Logics*, ser. Lecture Notes in Computer Science, J. Hurd and T. Melham, Eds. Springer Berlin / Heidelberg, 2005, vol. 3603, pp. 310–325, 10.1007/11541868_20. [Online]. Available: http://dx.doi.org/10.1007/11541868_20

[19] T. Braunl, "Parallaxis-III: a structured data-parallel programming language," in *Algorithms and Architectures for Parallel Processing, 1995. ICAPP 95. IEEE First ICA/sup 3/PP., IEEE First International Conference on*, vol. 1, 1995, pp. 43–52 vol.1.