International Conference on Information Science and Technology
March 26-28, 2011 Nanjing, Jiangsu, China

Formalization of Multi-Resolution Modeling Based on Dynamic Structure DEVS

Yuan Li, Bo Hu Li, Xiaofeng Hu and Xudong Chai

***Abstract*—So far there exists no general, common specification for multi-resolution modeling (MRM) to represent models consistent with each other. To address this problem, a new formalism, multi-resolution DEVS (MR-DEVS) is put forward. The definition of multi-resolution entity (MRE), which is the basic object MRM deals with and base for MR-DEVS, is given. Then, MR-DEVS, combining and extending multiple popular existing dynamic structure DEVS formalisms, is elaborated in detail. MR-DEVS is capable of capturing all key characteristics of MRM essentially, including adapting to resolutions changing automatically by reflection, describing consistency mappings and modeling causation by consistency mapping functions and modeling emergence by functions of coupling among sub-entities constituting the same MRE. Any other existing formalism can't do these all.**

I. Introduction

In modeling and simulation, models are abstractions of the real world generated to address a specific problem. Since all problems are not defined at the same level of physical representation, the models built to address them will be at different levels. The modeling a simulation problem domain is too rich to ever expect all models to operate at the same level. All these imply that multi-resolution models (MRMs) and techniques to provide interoperability among them are inevitable, and so bring about multi-resolution modeling (MRM) issues.

MRM are also at the heart of many substantive problems affecting model interoperability, reusability and composability. As shown in [1], MRM, which provides the capability for interoperability between multi-level resolution models, is one of the most interesting and appealing problems which must be solved in order to advance the field of distributed simulation significantly.

The general notion of MRM has existed since at least the early 1980s whose origins are related to the modeling concept commonly called the hierarchy of models pursued since the mid of 1970s [2]. So far, many methods for MRM were developed for different domain and problems, such as Aggregation/Disaggregation [2]-[4], Multi-Representation Entity (MRE) [5]-[7], Integrated Hierarchical Variable Resolution (IHVR) [2], [4], and so on. All of these approaches address some issues for MRM, and provide much practical significance for MRM practitioners. But, in practice, multi-resolution models developed by any known MRM approach can hard or even not be asserted to be effective or consistent with each other. This is mainly because there exists no general, common formal method for MRM to describe MRMs effectively. If models for different levels of entities are developed originally not in a common way, the linkage of them is certainly intractable because of the versatility of the structure, data types, interfaces, and mappings among them. "If the joined models have quite different representations, there might not even be a good way to join them at all" [8]. In our view, a common and formal way for specifying MRMs is at least a fine start for interoperability among them.

Just as Liu pointed out in [9], "the lack of formal specifications of MRM has become a main obstacle for the development of MRM. Without formalism, it is difficult to establish a common language among different researchers and model developers, and it is impossible to develop a multi-resolution modeling framework and tools for modeling automation".

This inspires our work in the paper. We introduce a new, general and common MRM formalism, called Multi-Resolution DEVS (MR-DEVS), by combining and extending multiple popular existing dynamic structure DEVS formalisms.

The rest of this paper is organized as following. In section 2, two definitions of multi-resolution entity and multi-resolution system, which are objects MRM deals with, are to be given. In section 3, popular existing dynamic structure DEVS formalisms are to be expounded briefly and main problems with them will be pointed out. Then, in section 4, combining and extending these formalisms, MR-DEVS will be introduced and elaborated, and its key properties including closure under coupling will be shown constructively. In section 5, we will illustrate how to describe a multi-resolution system with MR-DEVS with an example. Finally, we will conclude and point out the direction of our work.

II. Issues of Variable Structure in Multi-Resolution Modeling

In the section, we will first define multi-resolution entity and multi-resolution system which are objects MRM deals with and the bases for our formalism. And then we will analyze issues of variable structure in MRM briefly.

A. Multi-Resolution Entity and Multi-Resolution System

Definition 1 Multi-Resolution Entity (MRE)

A conceptual entity that can interact with other object(s) at multiple levels of resolution.

The concept of MRE was first put forward by Natrajan, et

Manuscript received October 29, 2010. This work was supported in part by the China National Defense Key Laboratory, and the National 973 (No. 2007cb310900).

Y. Li and B. H. Li are with the School of Automation Science and Electrical Engineering, BeiHang University, BeiJing 100083, China (phone:86-10-85824382; e-mail: YuenYuenLea@yahoo.com.cn).

X. F. Hu is with the National Defense University, Beijing 100091, China.

X. D. Chai is with the Beijing Simulation Center, Beijing 100854, China.

al. in [5], defined as "an entity that can be perceived at multiple levels of resolution concurrently". Later the concept was developed and redefined in [6] as "a conceptual entity that can interact at multiple levels of resolution concurrently". The latter is closer to our definition. But we argue for that the entity can interact with other object(s) at multiple levels of resolution but perhaps not concurrently. In other words, the interactions can happen at one level of resolution at one time and at another level higher or lower at another time. An MRE can designate the entities at all resolution levels that comprise it. So it can interact with other objects (that are also MREs or maybe common objects that are not MREs) at all resolution levels. For instance, a MRE of plane representing single airplanes can interact with a tank MRE representing single tanks, and the same MRE of plane but representing the formation of planes can interact with the tank MRE representing the tank platoon.

Definition 2 Multi-Resolution System (MRS)

A complex system that involves multiple kinds of multi-resolution entities (MREs).

An MRS is different from a common system in that the former is capable of dealing with or addressing issues of the change of resolutions. Whereas these issues are common for most systems to be modeled, they, however, are usually neglected in general systems modeling. It needs to be pointed out that those general methods for systems modeling are also applicable for MRM but not the focus of our research. We focus on how to formally describe mechanisms of resolutions changing and consistency mappings among models of the same MRE at different levels of resolution.

B. Issues of Variable Structure in MRM

As Uhrmacher pointed out in [10], ""Dynamic structure" is also called "variable structure". Zeigler coined the term "variable structure models" to describe models that contain in descriptions of their behavior the possibility of altering their own structure and, consequently, their behavior." The resolutions of MREs or MRSs are changeable, which brings about the change of structure, such as selection of different components, deletion of some components, or changing interactions among components which bring about variable interfaces, and so on. So, multi-resolution systems are of systems with variable structure, and MRM are of variable structure modeling in essence.

III. DYNAMIC STRUCTURE DEVS

"Variable structure models are a prerequisite for specifying and analyzing systems that will dynamically and in part autonomously adapt their interactions, composition, and behavior patterns" [10]. The ways of modeling systems with variable structure can roughly fall into two categories, i.e. those based on DEVS (Discrete Event System Specification) [11], [12] and those based on other formalisms. The former lends the temporal dimension to structure in describing systems by extending DEVS.

DEVS, introduced by Zeigler in [11], is a formal modeling and simulation framework based on system theory and mathematical theory rigidly. DEVS has well-defined concepts for coupling of components and hierarchical, modular model composition, support for discrete event approximation of continuous systems and an object-oriented substrate supporting repository reuse [13]. So, DEVS is popularly accepted and widely used. In fact, the DEVS approach is the mainstream for modeling variable structure. This is also why we take DEVS underlying our formalism.

However regular DEVS has a static structure with which it is difficult to adapt to changing systems dynamically. It needs to extend regular DEVS to address this problem. Some popular such DEVS variants are DSDE (parallel dynamic structure discrete event system specification) [14], dynDEVS (Dynamic DEVS) [10] and ρ -DEVS (variable ports dynDEVS) [15]. DSDE focuses on the possibility to dynamically change the system structure according to the system real requirements. In DSDE, the structural changes are carried out by the network executive χ. Each state of χ is mapped to a structure of network by the structure function γ. In this way, the dynamic evolution of network structure can be modeled by the model of network executive M_χ. However, what is the state of χ and how does it change? In fact, due to the abstraction of network executive its state is abstract and not well-defined, with the result that M_χ is incapable of describing how to transition the state of χ. So DSDE can only tell you the result of structure changing but cannot represent the mechanism for how to change network structure.

"According to Zeigler and Ören's definition, variable structure models are inherently reflective systems - reflection being defined as the ability of a (computational) system to represent, control, and modify its own behavior" [10]. The intrinsic reflective nature of variable structure models was however neglected in DSDE. Uhrmacher introduced this reflection into her formalism - dynDEVS [10]. In dynDEVS state transitions may possibly give rise to "new" models. A dynDEVS model can changes its structure, i.e. its state space and its behavior pattern, reflectively by model transition functions ρ_α. A dynDEVS model can be interpreted as a set of DEVS models with the same interface plus a transition function that determines which DEVS model succeeds the previous one [16]. From this, a dynDEVS is a multimodel [12]. Similarly, the system network, i.e. dynDEVS coupled model can give rise to new network incarnations and change the network structure by network transition function ρ_N.

However, some systems are characterized just by a plasticity of their interface. Thereby, they signalize significant changes to the external world. These phenomena can particularly be found in multi-resolution simulation. The ρ-DEVS introduced variable ports into dynDEVS, which enables the interfaces of models and the coupling among models to adapt to environment automatically [15].

Although these dynamic structure DEVS formalisms above can describe the variability of structure, they were introduced and developed not for MRM. They would meet some big problems when they are used for MRM [17]. First of all, they can't describe mechanisms for changing models with different resolutions. Secondly, they are incapable of representing consistencies among models. So, it needs to further extend and modify them. MRMS (Multi-Resolution Model system Specification) [17], [18], which is, to our knowledge, the first DEVS based formalism special for MRM and based on DSDE, was put forward by Liu Baohong in [17]. Two new key concepts of Multi-resolution model Family (MF) and Resolution Mode (RM) were put forward by Liu in MRMS. MF was introduced to represent models of the same entity at different resolution level and the relationship among those models, and RM was used to represent the combinations of different entities' resolutions in runtime. But by analyzing deeply, we find out that, according to the structure and operational semantics of the coupled model in MRMS, the information contained in MF can't be utilized in simulation. In fact, MRMS is rather another form for DSDE, and essentially provide no improvement in MRM.

Another known DEVS based formalism concerned with MRM is ML-DEVS (Multi-Level DEVS) [19], [20], developed recently by Uhrmacher. ML-DEVS is based on dynDEVS and ρ -DEVS and can model upward and downward causation and adapt to represent those systems with emergence and controlling relationship, as in military or biological system, between the micro level (high resolution) and macro level (low resolution). The modeling causation in ML-DEVS is modeling consistency among models at different levels. However, ML-DEVS was introduced for modeling systems biology and is applicable only for modeling two levels, i.e. "Micro-Macro" modeling. This means that it is not closed under coupling which is a requirement for hierarchical and modular modeling in distributed simulation.

From above, no existing formalisms including MRMS and ML-DEVS can be good candidates for MRM. So it is necessary and significant to develop a new one.

IV. MULTI-RESOLUTION DEVS

In the section, our formalism, i.e. MR-DEVS, is elaborated in detail, and its key properties including closure under coupling are shown constructively.

A. Multi-Resolution DEVS Atomic Model

Definition 3 Multi-Resolution DEVS Atomic Model

A reflective, dynamic structure DEVS, a prescriptive specification that is a meta-model used to describe Multi-Resolution Entities, with a structure described as following:

MR-DEVS

$=< X_{Res,} Y_{Res}, R, \Psi, \varphi_0, \{m_{\varphi_0}\}, \{M_r\}, \{C_{i\leftrightarrow j}\}, \{Z_\varphi\} >$ where:

$X_{Res,} Y_{Res}$: the ports to notify resolution changing. When changing resolution, the entity could send a notification through the ports Y_{Res} to other objects to change their resolutions accordingly (see also the variable λ_ρ following). The entity could also receive the notifying events for changing resolution from other entities or environment through the ports X_{Res}

R: the set of the entity resolutions, which can be regarded as the indexes of models at different levels of resolutions of the same entity

$\Psi = \{2^R - \phi\}$: the set of resolution modes the entity presents when interacting

φ_0: the initial resolution mode with $\varphi_0 \in \Psi$

m_{φ_0}: the incarnation of models under φ_0 with $m_{\varphi_0} \in M_{\varphi_0}$ ($M_{\varphi_0} \subset M_r$ which will be defined in the following)

$C_{i\leftrightarrow j}$: $S_{m_i} \leftrightarrow S_{m_j}$, the function of consistency mapping between models at different resolutions, where $i, j \in R, i \neq j$. For $\forall\, r \in R, m_r \in M_r, S_{m_r}$ is the set of states for m_r. When an MRE is executed in concurrent mode, i.e. the entity interacts with other object on multiple resolutions concurrently, **the consistency between models at different resolutions is hold by applying $C_{i\leftrightarrow j}$.** Because the effect caused by the change of one model at a resolution can be propagated to another model at a resolution higher or lower by applying $C_{i\leftrightarrow j}$, so $C_{i\leftrightarrow j}$ is in effect capable of modeling the causation between upward and downward levels of an entity. The thought is the same as that in ML-DEVS. If the entity is modeled only at one resolution, obviously $C_{i\leftrightarrow j} = \phi$.

$Z_\varphi : Y_r^i \to X_r^j$, is the coupling among different modules of the same entity in resolution mode φ with $\varphi \in \Psi$ and $r \in R$. $\forall\, r \in R$, Y_r^i, X_r^j is the output and input of the model m_r^i and m_r^j respectively, with $m_r^i, m_r^j \in M_r$, where i, j are the indexes of modules, $i \neq j$. When an entity is running under disaggregated states, those modules comprising the entity are not independent of each other. Just the reverse, they are coupled with each other to constitute a coupled model, which is the same thought as reflected in classical DEVS. **Z_φ can capture couplings among those deaggregated entities (DEs) constituting the same entity**, which is **requisite for modeling emergence but usually neglected by most other MRM approaches.** If the entity is comprised of only one module at resolution level r (generally speaking, when the r is the highest level), $Z_\varphi = \phi$.

M_r: the set of models of the entity at resolution level r with $r \in R$. For all $m_r \in M_r$, m_r is a modified DEVS model with a structure described as following:

$m_r =< X_r, Y_r, S_r, s_{r,0}, \delta_{int}, \delta_{ext}, \delta_{con}, \{\rho_{r\to i}\}, \lambda_\rho, \lambda, ta >$

where:

X_r, Y_r: the structured sets of inputs and outputs. The ports $X_{Res,} Y_{Res}$ at the level of MR-DEVS are responsible for notifying the changing of resolutions while the ports X_r, Y_r at the level of sub-entities, i.e. M_r, are responsible for sending events for simulation interactions. **Because of the variability of the inputs ports X_r and outputs ports Y_r, an MR-DEVS model has variable ports for interactions, following the same thought of variable ports as that in ρ-DEVS.**

S_r: the structured set of states

$s_{r,0}$: the initial state with $s_{r,0} \in S_r$

δ_{int}: $S_r \to S_r$, the internal transition function

δ_{ext}: $Q \times X_r{}^b \to S_r$, the external transition function with $Q = \{(s,e) | s \in S, 0 \le e \le ta(s)\}$ total state set including elapsed time e remaining at state s

δ_{con}: $S_r \times X_r{}^b \to S_r$, the confluent transition function

λ: $S_r \to Y_r$, the output function

λ_ρ : $S_r \to Y_{Res}$, the notifying function for changing resolution(s). The change of the states of the entity could require creating or deleting entities including perhaps the entity itself, or require or cause other entities to change their resolutions when the notification for these requirements could be sent through the port Y_{Res} to the higher parent network.

$ta: S_r \rightarrow R^{\geq 0} \cup \{\infty\}$, the time advance function

$\rho_{r\rightarrow i}: S_r \times X_{Res,} \rightarrow M_i$, the resolution changing function which is used to change models from one resolution to another one higher or lower while holding continuity between two successive sets of models, where $i, r \in R, i \neq r$.

The definition above is required to satisfy the property (Rule 1):

1. Any incarnation n of an entity is either equal to the initial incarnation m_{φ_0}, or result from the transition by the resolution changing function $\rho_{r_m \rightarrow r_n}$, i.e., $\forall n \in \{M_r\}$: $(\exists m \in \{M_r\}: n = \rho_{r_m \rightarrow r_n}(s^m)$ with $s^m \in S^m) \vee n = m_{\varphi_0}$.

2. Given two incarnations $m, n \in \{M_r\}, n \neq m, s^m \in S^m$, D^n and D^m the variable sets that structure the state space of m, n, if $n = \rho_{r_m \rightarrow r_n}(s^m)$, then the resulting state $s^n \in S^n$ of n is defined by:

$$s_d^n = \begin{cases} s_d^m, \forall d \in D^n \cap D^m \\ s_{init_d}^m, \forall d \in D^n \setminus D^m \end{cases}$$

where s_d^n and s_d^m are the values of variable d in *n*, *m*, respectively.

From the definition above, a model m_r at resolution level r is an atomic model, i.e., a modified parallel DEVS (PDEVS). As dynDEVS and ρ-DEVS, **an MR-DEVS is a multimodel**, which could be **interpreted as sets of PDEVS models with a resolution changing function that determines which set of models at one resolution level succeed the previous set of models at another resolution level higher or lower, according to its current states and the implied resolution changing requirement.** Input, output, state space, internal and external transition, output, and time advance functions of an MR-DEVS model are the same as those in PDEVS [12]. **The resolution changing function $\rho_{r\rightarrow i}$ answers to the state changes and implied resolution changing required by other entities by generating "incarnations" of dynamic PDEVS. This is so called "reflection" as in dynDEVS and ρ-DEVS.** Accordingly we can call $\rho_{r\rightarrow i}$ as reflection function. To support continuity between incarnations it preserves the values of variables that can't be aggregated/deaggregated from the preceding set of models or are not common to two successive sets of models, i.e., m and n, and assigns "default initial" values to the "new" variables.

As shown above, an MRE has a variable structure dynamically changing caused by the changing of resolutions. From this point of view, intuitively, if an MRE is with only one resolution, it degenerates to a common entity and the MR-DEVS could accordingly degenerate to a PDEVS.

PROPOSITION 1. If an entity is modeled only at one resolution level, the MR-DEVS model for the entity degenerates to a PDEVS model.

PROOF. Given that the entity has only one resolution r. Obviously, it could not be split or deaggregated into multiple sub-entities and has only one model M_r.

In the model represented with MR-DEVS, $X_{Res} = \phi$, $R = \{r\}$, $\Psi = \varphi_0 = \{r\}$, $m_{\varphi_0} = M_r$, $C_{i\leftrightarrow j} = \phi$, $Z_\varphi = \phi$.

Now, the structure of MR-DEVS could degenerate to $< Y_{Res}, M_r >$. Obviously, in M_r, $\rho_{r\rightarrow i} = \phi$. Let $Y_r = Y_r \cup Y_{Res}$, then $\lambda = \lambda_\rho + \lambda$. So, MR-DEVS=$< Y_{Res}, M_r > = < X_r, Y_r, S_r, s_{r,0}, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta >$=PDEVS. #

B. Multi-Resolution DEVS Coupled Model

Now, we can define MR-DEVS network (coupled) model based on the MR-DEVS atomic model above.

Definition 4 Multi-Resolution DEVS Coupled Model

A reflective, higher order dynamic MR-DEVS network, a meta-model used to describe Multi-Resolution Systems, with a structure described as following:

MR-NDEVS $=< X_{NRes}, Y_{NRes}, n_{init}, \mathcal{N} >$, where:

n_{init}: the initial network with $n_{init} \in \mathcal{N}$

X_{NRes}, Y_{NRes}: the ports to notify resolution changing. When the resolution of an entity in the network changes, the entity could send a notification through the ports Y_{Res} to other entities to change their resolutions. The network could pick up the notification from Y_{Res} and transmit it to other corresponding objects in the local network (see also the definition of ρ_N in the following) or send it to other networks through the ports Y_{NRes} (see also the definition of λ_N in the following). Of course, the network could also receive requirement events from other networks through the ports X_{NRes} for some entities to change resolutions.

$\mathcal{N}$: the least set of network incarnations with the following structure:

$n =< X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,j}\}, \rho_N, \lambda_N >$, where:

X, Y: the structured sets of inputs and outputs

D: the set of entities

M_d: the MR-DEVS model of entity d for all $d \in D$

I_d: the set of influencers of d for all $d \in D \cup \{N\}$ with $I_d \subset D \cup \{N\}$, N the name of the network itself.

$Z_{i,j}$: the i-to-j output-input translation function with $i, j \in D \cup \{N\}, i \neq j$

$\rho_N: S^n \times X_{NRes} \rightarrow \mathcal{N}$, the network transition function with $S^n = \underset{d \in D}{\times} Y_{Res}^d$, a cross-product of the notification events of all entities in the local network for changing resolution.

$\lambda_N: S^n \rightarrow Y_{NRes}$, the notifying function for changing resolution(s). When the resolution of an entity changes, the entity could require entities in other networks to change their resolutions by sending a notification event which will finally be sent out through the ports Y_{NRes}.

The definition above is required to satisfy the property (Rule 2):

1. $\forall n \in \mathcal{N}$: $(\exists m \in \mathcal{N} : n = \rho_N(s^m)$ *with* $s^m \in S^m) \vee n = n_{init}$, i.e., any network incarnation n either equals to the initial network configuration n_{init}, or result from the transition by the network transition function ρ_N.

2. Given two network incarnations *n*, $m \in \mathcal{N}$, $n \neq m$, let $s^m \in S^m$, D^n and D^m the sets of entities in n and m respectively, $m_d^m \in M_d^m$ and $m_d^n \in M_d^n$ the model incarnations of entity $d \in D^m$, $d \in D^n$ respectively. If $n = \rho_N(s^m)$ then:

$$m_d^n = \begin{cases} m_d^m & \forall d \in D^n \cap D^m \quad 1) \\ m_{init_d}^n & \forall d \in D^n \setminus D^m \quad 2) \end{cases}$$

1) The application of ρ_N preserves the state and structure of entities that are not common to the composition of the "old" network incarnation m and the "new" one n. In this case, m_d^m could be calculated further by rule 1 above.

2) Entities that are newly created are initialized. In this case , since enities are represented with MR-DEVS models, their initial states and structures, i.e. $m_{init_d}^m$ are given by the initial incarnation m_{φ_0} in its initial state $s_{r,0}$ (see the definition of MR-DEVS above).

The thought line of ρ_N is the same as those of network transition functions in dynDEVS and ρ-DEVS. For detailed information about them see the references [10], [15].

Like proposition 1, if all entities in the network have only one resolution, the network would degenerates to a common system without the variability of structure or with the variability of structure but not caused by the changing of resolution. In this case, MR-NDEVS would accordingly degenerate to a general PDEVS coupled models.

PROPOSITION 2. If all entities in a network are to be modeled only at one resolution level, the MR-NDEVS model for the network degenerates to a PDEVS coupled model.

PROOF. Let R^d the resolution of entity d . $|R^d| = 1$, for all $d \in D \Longrightarrow X_{Res}^d = Y_{Res}^d = \emptyset$, for all $d \in D$ $\wedge$ $X_{NRes} = Y_{NRes} = \emptyset \Longrightarrow S^n = \times_{d \in D} Y_{Res}^d = \emptyset \Longrightarrow \rho_N = \emptyset$ $\wedge$ $\lambda_N = \emptyset \Longrightarrow$ MR-NDEVS=$< X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,j}\} >$. As already shown in proposition 1, M_d is a basic PDEVS model. So, MR-NDEVS is a PDVES coupled model. #

Besides, MR-NDEVS is closed under coupling.

THEOREM 1. (Closure under coupling) The MR-DEVS formalism is closed under coupling, i.e., any network obtained by coupling models specified by the MR-DEVS formalism is itself specified by the MR-DEVS formalism.

PROOF. Suppose the MR-DEVS coupled model be:

MR-NDEVS=$< X_{NRes}, Y_{NRes}, n_{init}, \mathcal{N} >$, $\mathcal{N}$ set of network incarnations with the following structure:

$$n =< X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,j}\}, \rho_N, \lambda_N >$$

Now, it need to be proved that MR-NDEVS could be represented as a MR-DEVS atomic model with structure as following:

MR-DEVS=

$< X_{Res}, Y_{Res}, R, \Psi, \varphi_0, \{m_{init}\}, \{M_r\}, \{C_{i\leftrightarrow j}\}, \{Z_\varphi\} >$, where M_r has a structure:

$M_r =< X_r, Y_r, S_r, s_{r,0}, \delta_{int}, \delta_{ext}, \delta_{con}, \{\rho_{r,i}\}, \lambda_p, \lambda, ta >$.

Let $X_{Res} = X_{NRes}$, $Y_{Res} = Y_{NRes}$, $R = \bigcup_{d \in D} R_d$, R_d the set of resolutions of entity *d*. Let $\Psi = \{2^R - \phi\}$, $\varphi_0 = \bigcup_{d \in D_{init}} R_{init}^d$, D_{init} the set of entities in the initial network configuration, R_{init}^d the set of resolutions of entity *d* in the initial resolution mode. Let $\{m_{init}\} = \bigcup_{d \in D_{init}} \{m_{init}^d\}$, m_{init}^d the set of models of entity *d in* the initial resolution mode. Let $\{C_{i,j}\} = \bigcup_{d \in D} \{c_{i,j}^d\}$, $\{c_{i,j}^d\}$ the set of mapping functions of entity *d*. Let $\{Z_\varphi\}$ $=\{Z_{i,j}\} \cup (\bigcup_{d \in D} Z_\varphi^d)$, Z_φ^d the coupling function of modules in entity *d* in resolution mode φ.

Now, it is key to prove that the network incarnation n could be represented as the form of M_r . In M_r , the constructions of $X_r, Y_r, S_r, s_{r,0}, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta$ are the same as in the proving process for the closure under coupling of basic PDEVS [12]. Let $\lambda_\rho = \lambda_N$. The network transition function ρ_N of the network finds its counterpart in the resolution changing function $\rho_{r\rightarrow i}$, both functions adhere to the principle of value preservation and initialize new variables and new components, respectively [10].

Now, all variables in MR-NDEVS are translated into the variables in MR-DEVS. The proposition holds. #

C. Discussion

As shown in Fig.1, the thought line of the resolution changing function $\rho_{r\rightarrow i}$, which can change structure reflectively and adapt to environment automatically, inherits from the model transition function ρ_a with reflection in dynDEVS and ρ-DEVS. However the $\rho_{r\rightarrow i}$ is different from ρ_a in that the transition by $\rho_{r\rightarrow i}$ is from one set of models to another set of models while the transition by ρ_a is from one model to another model. Due to the variability of ports X_r, Y_r with different resolution r, the ports for interactions are changeable. So as in ρ-DEVS, a MR-DEVS model has variable ports which can adapt to the demand on changeable interfaces for interactions or other requirements from environment automatically. As MRMS, it is also capable of describing resolution modes including concurrent mode which is necessary for distributed simulations especially for joint military training. Besides, it is capable of representing consistency mappings between different levels of resolutions by consistency mapping function $C_{i\leftrightarrow j}$. The consistency mapping is another form for causation that captures the affecting between upward and downward levels. Furthermore, it can model emergence behavior by the couplings among modules or sub-entities of the same entity, i.e. "Z_φ" function. To our knowledge, other researchers deal with DEs as independent of each other. Even more, Liu argued for that the independency among DEs is a premise for aggregating DEs into one entity [9], [17], [18]. But just the reverse, in our view, those DEs maybe interact and couple with each other, which is so common in the real world. Emergence behaviors common in many systems such as system biology and military domains are such phenomena caused by interactions among DEs. Any other formalism except for ML-DEVS can't model consistency mapping, causation and emergence. In ML-DEVS, the causation and emergence can be modeled by lending states to coupled model and by value coupling function v_{down} , downward output function λ_{down} , and activation function act_{up} together. However, due to the orientation of these functions an ML-DEVS coupled model could not be translated into an ML-DEVS atomic model. This means ML-DEVS is not closed under coupling which is simply a fatal defect for MRM.

In summary, MR-DEVS can model dynamic structure, non-linearity, emergence behavior reflectively and adaptively while holding continuity and consistency among models. So it complies with Complex Adaptive Systems (CAS) theory.

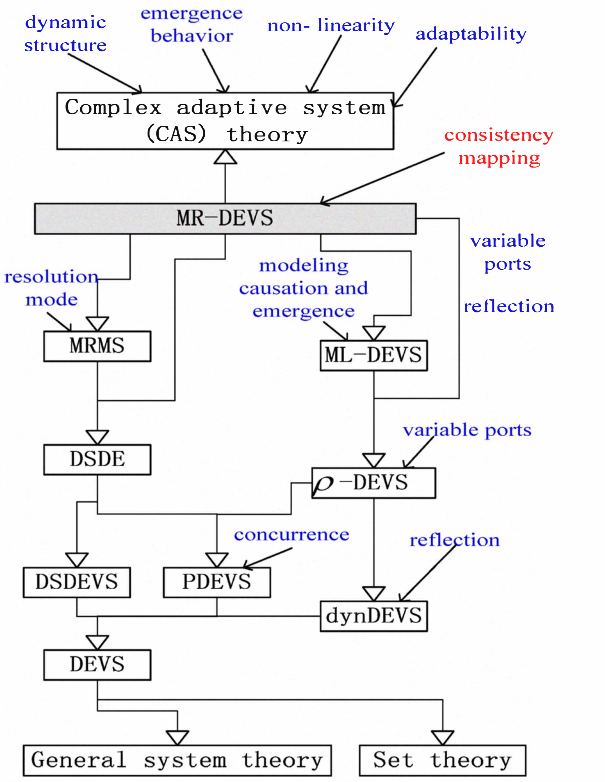

Fig.1. Relationship among MR-DEVS and other dynamic structure DEVS formalisms

V. AN EXAMPLE FOR MRM USING MR-DEVS

In this section, we will illustrate how to represent an MRS with MR-DEVS formalism with an example.

Let's consider such a system in [8]. As illustrated in Fig. 2, the system uses three different abstraction levels to model the duel between hostile attacking airplanes and anti air defense units. At the operational level, flight attack units, i.e. airplane formation, and air defense companies as the fundamental entities are to be modeled. On a tactical level, we have representations for single airplanes and single anti air defense units including air-surveillance, i.e. radar station and three firing units, i.e. air defense artillery. On the lowest level we will model bomb and missile impacts on a technical level with detailed representations for every single soldier, vehicle, missile and bomb.

The resolution changing criterion is: if the distance between the two hostile units (on the operational level) gets small enough, or if a bomb is dropped (at the tactical level), the appropriate disaggregation process is initiated. When the smallest distance between air defense units and airplanes is large enough or all units of one side are eliminated, the status information of each one of the soldiers and the unit's different capabilities are aggregated into the state of a firing or surveillance unit and only units (on the operational level) are to be modeled. Besides, if the director or some trainees who are authorized in the exercise want to have a view of some entities on two or more resolution levels at the same time, the resolution changing could be done manually by force.

Now, let's represent the system with our MR-DEVS formalism. As illustrated in Fig.2 described with another MRM formalism System Entities Structure (SES) [21], the system network model is represented by *MR-PA*, involving two MREs, plane and artillery, for which the models are *MR-P* and *MR-A* respectively. Suppose the simulation start at the operational level when the resolution of plane is formation and artillery is at company level. The models for airplane formation and artillery company are M_F and M_C respectively. Let the threshold distance for disaggregation/aggregation or resolution changing between the two hostile units is D.

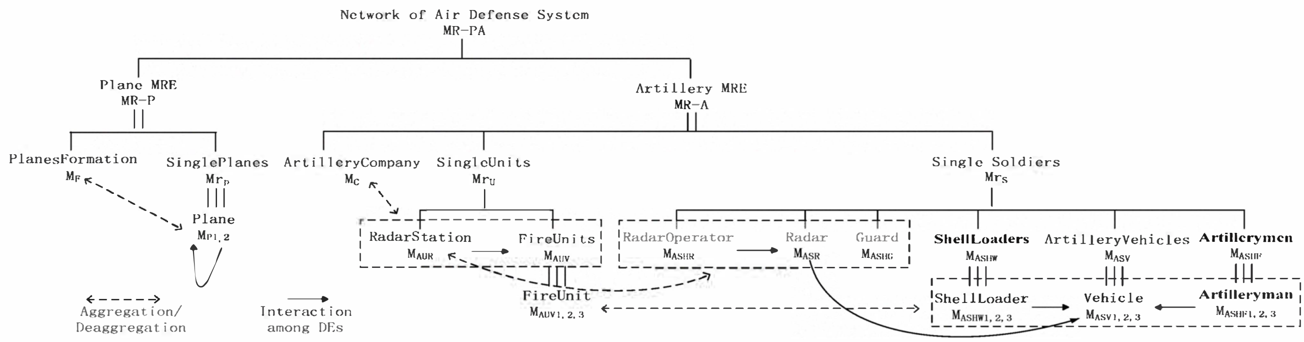

Fig.2. System Entities Structure (SES) of air defense system network

A. Plane MRE

The model for plane MRE, i.e. MR-P is represented with MR-DEVS atomic model.

$$MR-P \;=< X_{Res}^{P}, Y_{Res}^{P}, R_P, \Psi_P, \varphi_0^P, \{m_{\varphi_0}^{P}\}, \{M_r^P\}, C_{r_F\leftrightarrow r_P}^{P}, \{Z_\varphi^P\} > \text{ where:}$$

$X_{Res}^{P}=\{x^P|x^P \in\{F,P,FP\}\}$, where F, P, FP are input events for plane MRE to be executed at formation level, single plane level and concurrently at both levels respectively

$Y_{Res}^{P}=\{y^P|y^P \in\{DesC,DesU,AggU,AggS\}\}$, where DesC, DesU, AggU, AggS are output notifying events for artillery MRE to deaggregate or aggregate. Respectively, DesC is for disaggregation from the level of company to single units, DesU from single units to single vehicles and soldiers, AggU is for aggregation from single units to company, and AggS from single vehicles and soldiers to single units.

$R_P=\{r_F, r_P\}$, where r_F formation level of resolution, r_P single plane level of resolution

$\Psi_P=\{r_F, r_P, (r_F, r_P)\}$, resolution modes for plane MRE where (r_F, r_P) designates concurrent mode

$\varphi_0^P=\{r_F\}$

$m_{\varphi_0}^P=M_{r_{F,0}}$

$\{M_r^P\}=\{M_{r_F}, M_{r_P}\}$, where $M_{r_F}=\{M_F\}$ the set of models for plane MRE at formation level that has only one model M_F, $M_{r_P}=\{M_{P_1}, M_{P_2}\}$ the set of models for plane MRE at single plane level that has two models M_{P_1} and M_{P_2}

Model at formation level (operational level):

$M_F=<X_F, Y_F, S_F, s_{F,0}, \delta_{F,int}, \delta_{F,ext}, \delta_{F,con}, \rho_{r_F\to r_P}, \lambda_{F,\rho}, \lambda_F, ta_F>$, where the descriptions for those variables concerned with resolution are:

S_F ={FCor,FV,FS|{FCor= (Fx,Fy,Fz)|Fx,Fy,Fz $\in$ R)},FV $\in$ R,FS $\in$ {0,1,2}}, where FCor, FV, FS position, velocity and status (number of planes remained) of airplane formation respective

$\rho_{r_F\to r_P}: S_F \times X_{Res}^P \to \{M_{P_i}\}, i \in \{1,2\}$: if |FCor-ACor|<D $\vee$ x^P=FP $\vee$ x^P=P where ACor the position of artillery company, then the changing of resolution from formation level to single plane level would happen when the new models M_{P_i} for single planes would be created and initialized according to rule 1. Then the models of M_F would be removed out of simulation if x^P=P, otherwise $M_{P_i}$and M_F are executed concurrently if x^P=FP. In the initialization of M_{P_i}, the key is to call disaggregation function to get new states, i.e. (S_{P_1}, S_{P_2}) =$DeAgg_{r_F\to r_P}(S_F)$. During the process, the values preserved in the preceding aggregation would be utilized to hold continuity (see $\rho_{r_{P_i}\to r_F}$ following). It is an important, necessary and intractable work for MRM practitioners to design such disaggregation functions which are subject to many factors, such as military doctrine and command and control relationship.

$\lambda_{F,\rho}: S_F \to Y_{Res}^P$: if |FCor-ACor|<D then y^P=DesC

Models at single plane (tactical) level:

$M_{P_i}=<X_{P_i}, Y_{P_i}, S_{P_i}, s_{P_i,0}, \delta_{P_i,int}, \delta_{P_i,ext}, \delta_{P_i,con}, \rho_{r_{P_i}\to r_F}, \lambda_{P_i,\rho}, \lambda_{P_i}, ta_{P_i}>$ with $\boldsymbol{i} \in \{\mathbf{1},\mathbf{2}\}$

where the descriptions for those variables concerned with resolution are:

S_{P_i}={PCor,PV,PS,PFire|{PCor=(Px,Py,Pz)|Px,Py,Pz$\in$ R)}, PV$\in$R, PS$\in${0,1}, PFire$\in${0,1}}, where PCor position, PV velocity, PS status (0-destroyed, 1-live), PFire fight status(0-normal, 1-fire)

$\rho_{r_{P_i}\to r_F}: S_{P_i} \times X_{Res}^P \to M_F$: if |(PCor1+PCor2)/2-ACor|≥D $\vee$ x^P=FP $\vee$ x^P =F then as in $\rho_{r_F\to r_P}$, a process for resolution changing is to be done. Here the modelers need to design a aggregation function, i.e. $S_F= Agg_{r_P\to r_F}(S_{P_1}, S_{P_2})$: FCor = (PCor1+PCor2)/2, FV = (PV1+PV2)/2, FS= PS1+PS2, but not a disaggregation one in $\rho_{r_F\to r_P}$. Besides, the states of M_{P_i}, i.e S_{P_i} must be preserved to hold continuity between two successive resolution changing due to the loss of information during the aggregation. The values preserved would be used when the successive disaggregation happens (see $\rho_{r_F\to r_P}$ above).

$\lambda_{P_i,\rho}: S_{P_i} \to Y_{Res}^P$: if |(PCor1+PCor2)/2-ACor| ≥ D then y^P=AggU, If PFire=1 then y^P=DesU

Consistency mapping functions:

$C_{r_F\to r_P}^P: S_F \to (S_{P_1}, S_{P_2})$: $(S_{P_1}, S_{P_2})=DeAgg_{r_F\to r_P}(S_F)$

$C_{r_P\to r_F}^P: \times_{i\in\{1,2\}} S_{P_i} \to S_F : S_F= Agg_{r_P\to r_F}(S_{P_1}, S_{P_2})$

In simulation, if the plane MRE is required to executed concurrently at both levels, it would receive the event for this requirement through the ports X_{Res}^P, i.e. x^P=FP when the consistency mapping functions $C_{r_F\to r_P}^P$ or $C_{r_P\to r_F}^P$ would be called to maintain consistency between models at the two levels.

$Z_\varphi^P: Y_{P_1} \to X_{P_2}$, $\varphi \in \{r_P, (r_F, r_P)\}$. If the plane is modeled at single plane level or in concurrent mode, the wing plane (represented by M_{P_2}, for instance) must receive commands from the leading plane (M_{P_1}) (see Fig.3(b) or (c)).

From above, it can be found that the disaggregation/aggregation functions are the same as consistency mapping functions. Indeed it is so.

In practice, the design of aggregation function is much easier than that of disaggregation function. So we could maintain consistencies through aggregation but not disaggregation as possible. In our scenario, we take the aggregation process, i.e. $C_{r_P\leftarrow r_F}^F$, and the aggregation function is very simple, i.e. $S_F = Agg_{r_P\to r_F}(S_{P_1}, S_{P_2})$: FCor = (PCor1+PCor2)/2, FV= (PV1+PV2)/2, FS= PS1+PS2.

Now, we have illustrated how to represent the plane MRE with MR-DEVS in detail.

B. Artillery MRE

In the section, we will represent the artillery MRE with MR-DEVS, the process of which is similar to that of plane MRE above. To save space, we will only enumerate models and their key variables concerned with resolution but not elaborate them in detail.

$$MR-A=<X_{Res}^A, Y_{Res}^A, R_A, \Psi_A, \varphi_0^A, \{m_{\varphi_0}^A\}, \{M_r^A\}, \{C_{r_C\leftrightarrow r_U}^A, C_{r_U\leftrightarrow r_S}^A\}, \{Z_\varphi^A\}>, \text{ where:}$$

$R_A=\{r_C, r_U, r_S\}$, the set of resolutions for company level, single units level and single vehicle and soldier level respective

$\Psi_A=\{r_C, r_U, r_S, (r_C, r_U), (r_U, r_S), (r_C, r_U, r_S)\}$

$X_{Res}^A=\{x^A | x^A \in \{C,U,S,CU,US,CUS\}\}$, where C, CUS input event for the artillery MRE to be executed at company level and concurrently at all three levels respectively

$Y_{Res}^A=\{y^A | y^A \in \{DesF\}\}$, where DesF output notifying event for the plane MRE to deaggregate from formation to single plane level

$\varphi_0^A=\{r_C\}$

$\{m^A_{\varphi_0}\}$=$\{M_{C,0}\}$

$\{M^A_r\}$=$\{M_{r_C}, M_{r_U}, M_{r_S}\}$, where

M_{r_C}=$\{M_C\}$ set of models at company **(operational)** level r_C with only one mode M_C for artillery company, M_{r_U}=$\{M_{AUR}, \{M_{AUV_i}\}\}$, i ∈ {1,2,3} set of models at single units **(tactical)** level r_U with a model M_{AUR} for radar station and three models $\{M_{AUV_i}\}$ for three artillery vehicles, M_{r_S} =$\{ M_{ASHR}, M_{ASR}, M_{ASHG}, \{M_{ASV_i}\}, \{M_{ASHW_i}\}, \{M_{ASHF_i}\} \}$, i ∈ {1,2,3} set of models at single vehicle and soldier **(technical)** level r_S with M_{ASHR} for radar operator, M_{ASR} for radar, M_{ASHG} for radar guard, M_{ASV_i} for artillery vehicle, M_{ASHW_i} for shell loader, M_{ASHF_i} for artilleryman

Consistency mapping functions:

$C^A_{r_U \to r_C}$: $S_{\mathrm{AUR}} \times (\underset{i\in\{1,2,3\}}{\times} S_{AUV_i}) \to S_C$

$C^A_{r_S \to r_U}$: $S_{\mathrm{ASHR}} \times S_{\mathrm{ASR}} \times S_{\mathrm{ASHG}} \to S_{AUR}$

$S_{ASV_i} \times S_{ASHW_i} \times S_{ASHF_i} \to S_{AUV_i}, i \in \{1,2,3\}$

$C^A_{r_C \to r_U}$: $\mathrm{S_C} \to (\mathrm{S_{AUR}}, \mathrm{S_{AUV_1}}, \mathrm{S_{AUV_2}}, \mathrm{S_{AUV_3}})$

$C^A_{r_U \to r_S}$: $\mathrm{S_{AUR}} \to (\mathrm{S_{ASHR}}, \mathrm{S_{ASR}}, \mathrm{S_{ASHG}})$

$S_{AUV_i} \to (S_{ASV_i}, S_{ASHW_i}, S_{ASHF_i}), i \in \{1,2,3\}$

Functions of coupling among deaggregated entities (see Fig.3(b) and (c)):

Z^A_φ: $Z^A_{r_U}$: $Y_{AUR} \to X_{AUV_i}$, $i \in \{1,2,3\}$

$Z^A_{r_S}$: $Y_{ASHR} \to X_{ASR}$

$Y_{ASR} \to X_{ASV_i}$, $i \in \{1,2,3\}$

$Y_{ASHW_i} \to X_{ASV_i}$, $i \in \{1,2,3\}$

$Y_{ASHF_i} \to X_{ASV_i}$, $i \in \{1,2,3\}$

C. System Network Model

The plane MRE and artillery MRE together constitute a network of system with a structure as following:

MR-PA=$< X_{NRES}, Y_{NRES}, n_{init}, \mathcal{N} >$, where:

X_{NRes}=$\{(x^P, x^A) | (\mathrm{x^P}, x^A) \in \Psi_P \times \Psi_A\}$, input events for changing the structure of network that is one of combinations (cross-product) of resolution modes for plane MRE and that for artillery MRE

Y_{NRes}=∅

$\forall n \in \mathcal{N}$ the set of structures:

$n =< X_N, Y_N, D, \{M_d\}, \{I_d\}, \{Z_{i,j}\}, \rho_N, \lambda_N >$

where:

D={P,A}, P index of plane MRE, A artillery MRE

$\{M_d\}$=$\{MR - P, MR - A\}$

I_P={*N,A*}

I_A={*N,P*}

I_N={*P,A*}, where N the network itself

Z_P: $X_N \times Y_A \to X_P$

Z_A: $X_N \times Y_P \to X_A$

Z_N: $Y_P \times Y_A \to Y_N$

λ_N=∅

ρ_N: $Y^P_{Res} \times Y^A_{Res} \times X_{NRes} \to \mathcal{N}$

Now, let's enumerate several network incarnations from the start of the initial configuration n_{init}.

Note: in the following, the indications of the subscripts of network incarnation n are: f, p, fp indicate plane is to be simulated at formation level, single plane level and concurrently at both levels respectively, the indications of c, u, s, us, cu, cus for artillery MRE are similar. The sign ⊕ indicates a relationship of combination or interaction or resolution mode. For instance,$fp \oplus cu$ indicates concurrent interactions between plane MRE at formation and single plane levels and artillery MRE at company and single unit levels.

n_{init}=$n_{f\oplus c}$ the initial network configuration is in "airplane formation-artillery company" mode (see Fig.3(a)):

$\{M^{init}_d\}$=$\{ MR - P_{init}, MR - A_{init}$ =$\{M_{\mathrm{F},0}, M_{\mathrm{C},0}\}$

Z_F: $X_{N,\mathrm{FC}} \times Y_C \to X_F$

Z_C: $X_{N,FC} \times Y_F \to X_C$

$Z^{f\oplus c}_N$: $Y_C \times Y_F \to Y_{N,FC}$

$\rho^{f\oplus c}_N$: if y^F=*DesC* ∨ y^A=DesF ∨ (x^P, x^A)= (P,U) then $n_{p\oplus u}$=$\rho_{n_{f\oplus c} \to n_{p\oplus u}}$

$n_{p\oplus u}$（"single plane-fire unit" mode, see Fig.3(b)）:

$\{M^{p\oplus u}_d\}$=$\{M_{\mathrm{r_p}}, M_{\mathrm{r_u}}\}$=$\{\{M_{P_1}, M_{P_2}\}, \{ M_{AUR}, M_{AUV_1}, M_{AUV_2}, M_{AUV_3}\}\}$

Z_{P1}: $X_N \times (\overset{3}{\underset{j=1}{\times}} Y_{AUV_j}) \to X_{P_1}$

Z_{P2}: $\overset{3}{\underset{j=1}{\times}} Y_{AUV_j} \to X_{P_2}$

Z_{AUR}: $\overset{2}{\underset{i=1}{\times}} Y_{P_i} \to X_{AUR}$

Z_{AUV_i}: $\overset{2}{\underset{j=1}{\times}} Y_{P_j} \to X_{AUV_i}$, $i \in \{1,2,3\}$

$Z^{p\oplus u}_N$: $Y_{AUR} \to Y_N$

$\rho^{p\oplus u}_N$: *if* y^P=*DesU* ∨ (x^P, x^A)= (P,US) then $n_{p\oplus us}$=$\rho_{n_{p\oplus u} \to n_{p\oplus us}}$

$n_{p\oplus us}$（"single plane- fire unit-single soldier" **concurrent mode**, see Fig.3(c)）:

$\{M^{p\oplus us}_d\}$=$\{M_{\mathrm{r_p}}, M_{\mathrm{r_u}}, M_{\mathrm{r_s}}\}$=$\{\{M_{P_1}, M_{P_2}\}, \{ M_{AUR}, M_{AUV_1}, M_{AUV_2}, M_{AUV_3}\}, \{M_{ASV}, M_{ASHW}, M_{\mathrm{ASHF}}\}\}$

Z_{P1}: $X_N \times (\overset{3}{\underset{j=1}{\times}} Y_{AUV_j}) \to X_{P_1}$

Z_{P2}: $\overset{3}{\underset{j=1}{\times}} Y_{AUV_j} \to X_{P_2}$

Z_{AUR}: $\overset{2}{\underset{i=1}{\times}} Y_{P_i} \to X_{AUR}$

Z_{AUV_i}: $\overset{2}{\underset{j=1}{\times}} Y_{P_j} \to X_{AUV_i}$, $i \in \{1,2,3\}$

$Z^{p\oplus u}_N$: $Y_{AUR} \to Y_N$

$\rho^{p\oplus us}_N$: if y^P=*AggU* then $n_{f\oplus c}$=$\rho_{n_{p\oplus us} \to n_{f\oplus c}}$; if (x^P, x^A)= (FP,CUS) then $n_{fp\oplus cus}$=$\rho_{n_{p\oplus u} \to n_{fp\oplus cus}}$. In this case, the network structure would become the most complex because all levels of all entities would be simulated concurrently, which would bring about the most intricate and perhaps intractable interaction relationships among entities and their sub-entities at different resolution levels.

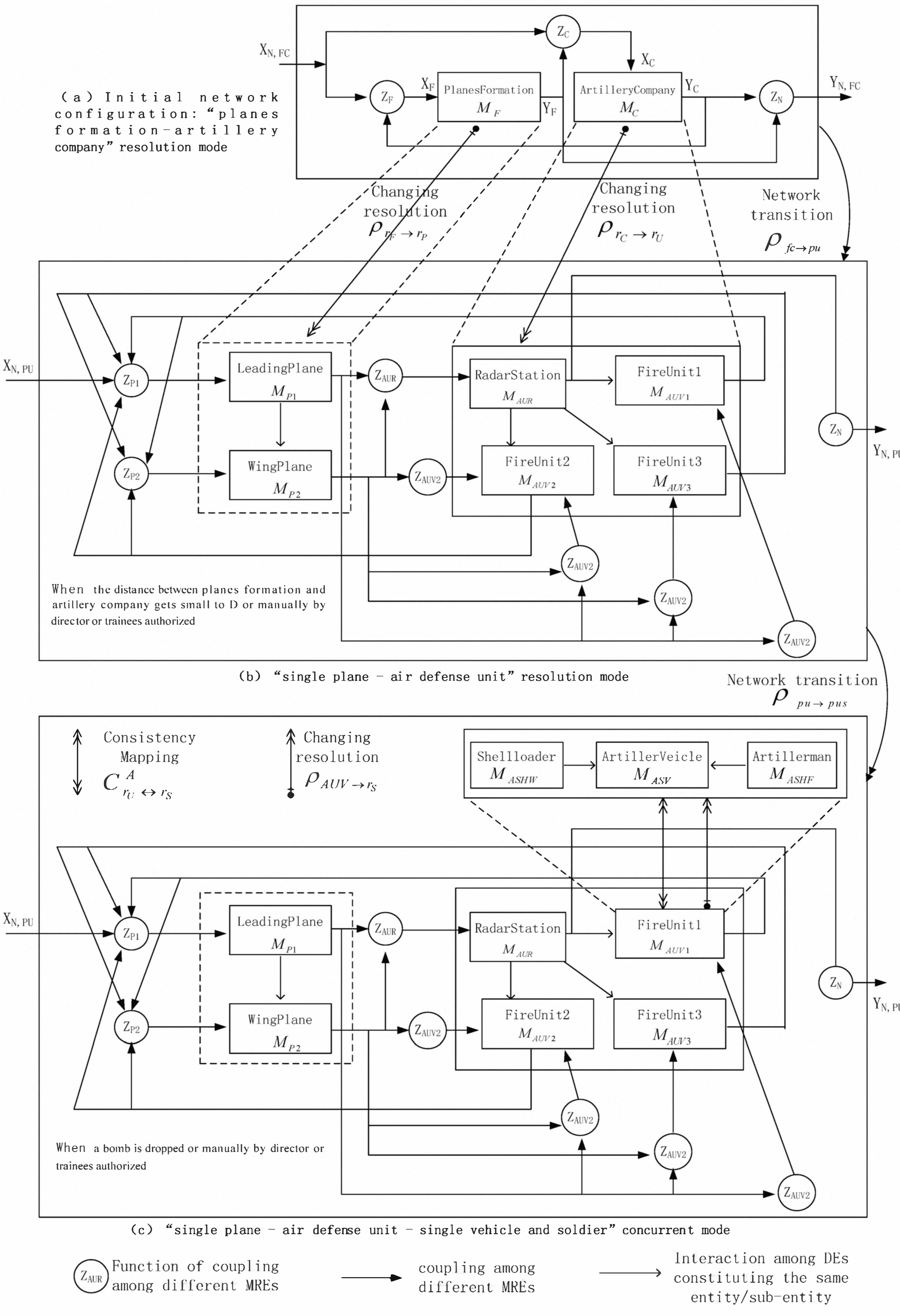

Fig.3. Incarnations of the structure of air defense system network and their transitions

The structures of the three incarnations and transitions among them are illustrated as Fig. 3. We can also represent the remained structures and their transitions by following the process above.

From above, we can fully see the complexity of issues for MRM. The system in our scenario is just a typical but simple one. In practice, those systems or issues MRM deals with maybe much more complicated and intractable. But those MREs and MRSs can still be represented with MR-DEVS no matter how complex they are.

VI. CONCLUSION

MR-DEVS combines and extends multiple popular existing dynamic structure DEVS formalisms. By comparing Fig.2. described with SES and Fig.3. with our formalism MR-DEVS, it could easily be found out that MR-DEVS is capable of describing the dynamic evolution of structures caused by resolution changing while SES can only describing the static structure or a "snapshot" of the structures of an MRS. Besides, and of course, MR-DEVS has other many advantages over SES and other formalisms. In sum, MR-DEVS is capable of explicitly capturing and describing all aspects of key characteristics of MRM, i.e. self-adaptability or reflection, modeling continuity and consistency, modeling emergence and causation. The example shows that our formalism has these capabilities and is powerful and totally self-contained. And so MR-DEVS can meet the goal of and requirements for MRM.

In near future, we will develop simulator algorithms for the MR-DEVS which support executing MR-DEVS models and simulation effectively and efficiently, and develop a tool for modeling any MRSs, which can assert consistencies among models and the goals of MRM to be obtained automatically.

REFERENCES

[1] S. Straßburger, T. Schulze, and R. Fujimoto, "Future Trends in Distributed Simulation and Distributed Virtual Environments," The Fraunhofer IFF in Magdeburg, Peer Study Final Report, 2008, Version 1.0.

[2] P. K. Davis and R. K. Huber, "Variable-Resolution Combat Modeling: Motivations, Issues, and Principles". RAND Technical Rep., N-3400-DARPA, 1992.

[3] J. Klír, "Variable Resolution Modeling in Interactive Simulation," Department of Computer Science and Engineering of University of West Bohemia, Czech Republic, Technical Rep. No. DCSE/TR-2004-07, 2004.

[4] P. K. Davis and J. H. Bigelow, "Experiments in Multiresolution Modeling (MRM)," RAND Technical Rep. MR-1004-DARPA, 1998.

[5] A. Natrajan, P. F. Jr., Reynolds and S. Srinivasan, "Consistency Maintenance using UNIFY," Dept. of Computer Science, University of Virginia. Technical Rep. No. CS-95-28, June 16, 1995.

[6] P. F. Jr., Reynolds, A. Natrajan, and S. Srinivasan, "Consistency maintenance in multiresolution simulations" *ACM Transactions on Modeling and Computer Simulation* , vol.7, no.3, pp.368-392, July 1997.

[7] A. Natrajan, "Consistency Maintenance in Concurrent Representations," Ph.D. dissertation, School of Engineering and Applied Science of University of Virginia, 2000.

[8] T. Seiger, G. Holm and U. Bergsten, "Aggregation/Disaggregation Modeling in HLA-based Multi Resolution Simulations," in *Proc. of the 2000 Fall Simulation Interoperability Workshop.*

[9] B. H. Liu, G. Li, and L. Yan, "Theoretical Analysis and Object-Oriented Design of Aggregation and Disaggregation in Multi-Resolution Modeling System," in *Proc. Of the 2009 Asia Simulation Conf.*

[10] A. M. Uhrmacher, "Dynamic Structures in Modeling and Simulation: A Reflective Approach," *ACM Transactions on Modeling and Computer Simulation*, vol. 11, no. 2, pp. 206–232. April 2001.

[11] B. P. Zeigler, *Theory of Modelling and Simulation*, New York: John Wiley & SONS, 1976.

[12] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of modeling and simulation*, 2nd Edition. New York: Academic Press, 2000.

[13] Shang H., G. Wainer, "A Simulation Algorithm for Dynamic Structure DEVS Modeling," in *Proc. of the 2006 Winter Simulation Conf.*, pp.815-822.

[14] F. J. Barros, "Modeling formalisms for dynamic structure systems," *ACM Transactions on Modeling and Computer Simulation*, vol. 7, no. 4, pp.501–515,Oct. 1997.

[15] A. M. Uhrmacher, J. Himmelspach, M. Röhl, and R. Ewald, "Introducing Variable Ports and Multi-Couplings for Cell Biological Modeling in DEVS," in *Proc. of the 2006 Winter Simulation Conference*, pp.832-840.

[16] G. A. Wainer, *Discrete-Event Modeling and Simulation-A Practitioner's Approach.* CRC Press of Taylor & Francis Group, 2009.

[17] B. H. Liu, "Research on Theory of and Key techniques for Multi-Resolution Modeling," Ph.D. dissertation, University of National Defense Science and Technology, Changsha, China, 2003 (in Chinese).

[18] B. H. Liu, "A Formal Description Specification for Multi-Resolution Modeling Based on DEVS Formalism and its Applications". *The Journal of Defense Modeling and Simulation: Application, Methodology, Technology.* Vol.4, No.3, July 2007.

[19] A. M. Uhrmacher, R. Ewald, M. John, C. Maus, M. Jeschke and S. Biermann, "Combining Micro and Macro-Modeling in DEVS for Computational Biology," In *Proc. of the 2007 Winter Simulation Conf.*

[20] C. Maus, M. John and A. M. Uhrmacher. (2010). A Multi-Level and Multi-Formalism Approach for Model Composition in Systems Biology. [Online]. Available: http://www.diemosiris.de.

[21] B. P. Zeigler. "A Systems Methodology for Structuring Families of Models at Multiple Levels of Resolution," in *Proc. of Conference on Variable-Resolution Modeling*, Washington, DC, 1992, pp. 377-408.