# Expanding DEVS and SES Applicability: Using M&S Kernels within IT Systems

**Chungman Seo, Wontae Kang, Bernard P. Zeigler, and Doohwan Kim**

**RTSync Corp.**
**530 Bartow Drive Suite A**
**Sierra Vista, AZ, 85635, USA**
**cseo,wontae.kang,zeigler,dhkim@rtsync.com**

**Keywords:** Discrete Event System Specification (DEVS), System Entity Structure (SES), Pruned Entity Structure (PES), Active Calendar, MS4 Me Software Tool

**Abstract**

*In this paper, we show how the SES construction, pruning, and model generation process, can support a web-based system for personal wellness planning and self-management. Our focus is on the process by which modeling and simulation kernels are integrated within other layers of a web-application. The System Entity Structure (SES), Pruned Entity Structures (PES), and libraries for pruning processes from the MS4 Me™ platform technology are used to implement the personal wellness plan web application. This paper discusses the approach taken to support user graphic and web service requirements for interfaces to the embedded SES/PES kernels. We conclude with a discussion of how this example generalizes to integration of Discrete Event System Specification (DEVS) and SES with information technology systems thus increasing their applicability within such systems.*

## 1. INTRODUCTION

The capabilities of DEVS and SES for instantiating and implementing core elements of the theory of modeling and simulation are continuing to advance [1-3]. However, increasing the applicability of DEVS and SES will need improved concepts and tools for interfacing these core elements with multiple layers of hardware and software within emerging information technology (IT) systems.

Concentration in the literature so far has been in embedding DEVS components within hardware, e.g., as network controllers [14]. In this paper we show how the SES construction, pruning, and model generation process, can support a web-based system for personal wellness planning and self-management. An increasing litany of chronic conditions such as obesity, cancer, and heart problems are being linked to the nutritionally-challenged and sedentary life-style originating in America and spreading globally. Personalized plans for maintaining and improving health are designed for individuals by specialists such as nutritionists and weight trainers. These plans and associated applications are popular with those who can afford them, but beyond the budgets of the large majority of people. To reduce this disparity in ability to maintain a healthy life-style, we propose an application with which users can create their own wellness plans from templates designed by doctors, nutritionists, and health trainers over a web browser.

The System Entity Structure (SES), Pruned Entity Structures (PES), and libraries for pruning processes in the MS4 Me™ platform technology [3-5] are used underneath the personal wellness plan web application. A primary SES represents all possible wellness plans constructed with inputs from a variety of specialists such as pediatricians, exercise experts, and dieticians. This SES is partitioned into sub-SESs related to specific areas such as exercise and diet, which are pruned by users, transformed into DEVS models that can display timely recommended actions as well as reporting on their execution.

The main focus of this paper concerns the design and implementation of this process for cheap but effective use. The process is implemented in a series of web pages similar to booking a flight. The first page, called Design Wellness Plan, displays the basic SES in a form that 1) supports users' selection of entities from specializations and 2) allows users to add entities at the lowest level specializations thereby extending the SES to better meet their needs. After the final submission, the user's selections generate a corresponding PES which is pruned and transformed to a DEVS coupled model, called a Wellness Plan, (e.g., exercise, diet, medication, etc.) whose atomic models represent discrete recommended actions, their consecutive linkage, timing, and SES/PES provenance.

Users can record their performance of self-designed wellness plans through the Daily Actions web page, which shows recent past actions along with check boxes to verify performance. It also primes the user with immediately future actions to do. Finally, users can check their compliance trends through the Compliance page which calls services to track the verified actions against those recommended.

The personal wellness plan web application facilitates individually self-tailored plans for self-management at

greatly reduced cost. We will discuss the approach taken to support the Design Wellness Plan, Daily Actions, and Compliance user graphic and web service interfaces requirements using the SES/PES library functions of MS4 Me™ as well the limitations of, and research needed to improve, such a methodology.

In the rest of the paper, section 2 addresses System Entity Structure and MS4 Me Software as background. Section 3 addresses an overall concept of personal wellness planning and self-management system. An example of wellness plan is given in the section 4. Implementation of personal wellness planning and self-management in cloud environment is discussed in the section 5. The paper's summary and future work are in the section 6.

## 2. BACKGROUND

### 2.1. System Entity Structure

A system can consist of various elements which can come from other systems. System Entity Structure (SES) is to express the structure of the system in which *entities* represent elements of the systems and the system itself [3]. The SES illustrates relationships between entities with three symbolic components: *aspect, specialization, multi-aspect*. The *aspect*, depicted with a vertical line, displays decomposition relationship between an entity and its parts. The *entity* can be broken into its parts which are expressed as entities. The *specialization*, depicted with double vertical lines, denotes inheritance between entities, which come from one of an object oriented concepts. An entity with a specialization relation can be represented by selection of one of the entities. The *multi-aspect*, depicted with triple vertical lines, expresses that a system can consist of multiple (independently prunable) copies of entities.

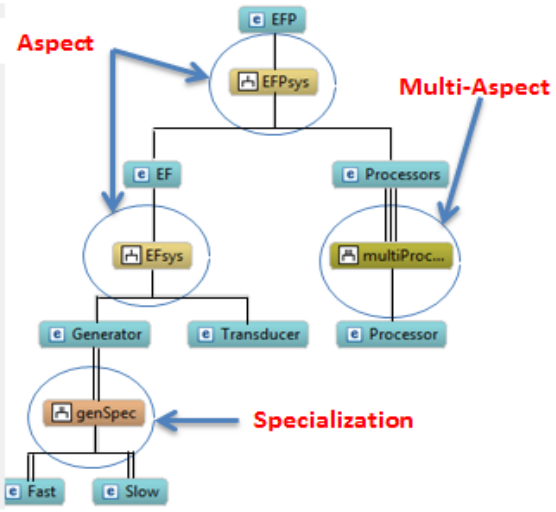


**Figure 1. SES Representation of an EFP System**

Figure 1 illustrates a tree structure of an SES document for an EFP System which is made of an Experimental Frame (EF) and Processors entities. There are identifiers, such as *EFPsys*, *EFsys*, *multiprocessor*, and *genSpec*, to name each relationship between entities. EFP and EF entity have Aspect relationships with EF and Processors entities, and Generator and Transducer entities, respectively. Processors entity has a Multi-Aspect relationship with Processor entity, and Generator entity has a Specialization relationship with Fast and Slow entities.

The EFP SES document expresses all possible instances for the EFP system just as an XML schema contains all possible documents for XML instances. A specific EFP is generated by the pruning process which includes selection of entities from the Specialization relation and restricting the number of entities with the Multi-Aspect relation. For example, if the Fast entity is selected for the Generator entity, and if Processors entity is restricted to three entities for the Processor entity, the pruned *EFP* structure consists of *EF* entity and three *Processor* entities, and *EF* entity is made of *Fast_Generator* and *Transducer* entities. A document containing such pruning rules seen in the example is called Pruned Entity Structure (PES) to generate a specific model from a SES document [3, 6].

### 2.2. MS4 Me Software

MS4 Modeling Environment (MS4 Me™) is a software tool for DEVS Modeling and Simulation based on Java computer language and Eclipse RCP environment [7] developed by MS4 Systems (available from www.ms4systems.com). One of the model construction techniques that it provides is a top down modeling methodology with a sequence diagram which represents the overall system structure as well as bottom up modeling method with a state diagram which constructs behaviors of an atomic model [8]. The sequence diagram generates System Entity Structure (SES) documents expressed in restricted natural language [3], and the state diagram is expressed in a DEVS Natural Language (DNL) document and represents a DEVS atomic model with restricted natural language [2]. MS4 Me™ automatically generates atomic models implemented in the Java language from the DNL files, and a coupled model through running a PES file generated by a pruning process for the SES document. For advanced users who are familiar with restricted natural languages for SES and DNL documents, MS4 Me™ provides editing capability with highlighted keywords used in SES documents and DNL documents.

An SES document is created by a graphical tool or a text editor in the MS4 Me™ which has several menus to display the SES document and generate a PES document from it. The sequence diagram, as graphical tool, illustrates entities

which send and receive messages among them. An entity can have three relationships to express more complex SES documents in the sequence diagram. The graphical tool produces from the sequence design a SES file which is editable in the SES editor and automatically demonstrated by an outline view in which an entity can be expanded to see children entities or collapsed to hide them. When a SES file opens in the editor, SES menus appear in the menu bar. A "Show SES Tree" button displays a hierarchical tree structure for the SES file as Figure 1 shows. A "Merge All" menu generates a new integrated SES file with existing SES files whose root entities are children entities in the main SES file. With this button, an SES document can be a module used as sub system in Systems of Systems [3]. To build a PES document from a SES document, a "Prune SES into PES" button provides a graphic user interface to select entities in specializations in the SES document. After finishing selection of specializations, a PES file is generated in a project folder. For multi-aspect pruning, a user can write restricted natural language command sentences in the existing PES document. For more detailed information about the PES commands, refer to [3].

MS4 Me™ provides two different application areas (modeling & simulation, and data engineering) using an SES and a PES document. For modeling & simulation perspective, each entity in an SES document is assigned to a DEVS model which can be generated by a DNL file (an atomic model) or pruning process (a coupled model). By executing a PES document, a coupled model is generated and displayed in a simulation viewer. The simulation viewer shows the coupled model and its component-models. The simulation can be controlled using *Restart, Pause, Run, Step, and "Run to"* buttons at the bottom of the simulation viewer. Once a Java coupled model is created, users can run the coupled model with a simulation viewer without the PES file. For data engineering perspective, the pruning process generates specific information from the general description (SES document) [9]. The PES document describes certain conditions to extract detailed data from the SES document. The specific information is contained in the names of the nodes of the tree structure that is generated.

## 3. PERSONAL WELLNESS PLANING AND SELF MANAGEMENT SYSTEM

The purpose of wellness plan and self-management system is that users create their own wellness plan and check their compliance with the wellness plan in an active calendar which shows the users what to do at the current time and collects users' indications as to whether they have carried out the recommended actions. .

The Active Calendar Application for wellness planning consists of three processes (*wellness template, wellness plan,* and *active calendar*) as Figure 2 shows. The wellness template process is to design the contents of individually-tailored wellness plans. For example, in case of an exercise wellness plan, the plan should have information for exercise items, duration, where to do it (Location), how many times users do it per week (Week Schedule), and when to do it within a day (Time Slot). A nutritionist provides a food list classified by food groups as basis for development of a diet plan.

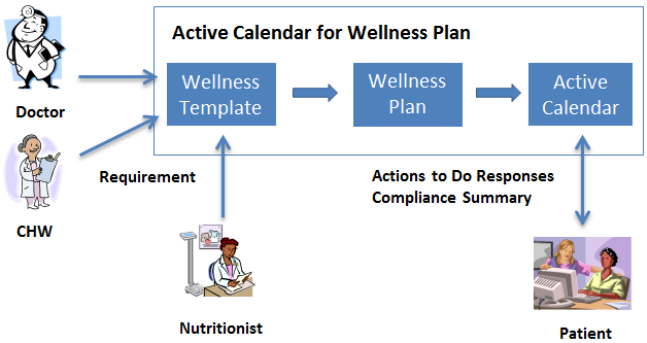


**Figure 2. Active Calendar Application**

The wellness template is built around an SES and can be considered as a multifaceted system which represents a family of possible systems through choices of decompositions and components within them. Based on the pruning process, users shape their own specific wellness plan from the wellness template. The active calendar saves the wellness plan for each user, and has user interaction functions such as displaying current plans to users, and recording users' compliance with their plans. A patient checks his/her actions to do in the active calendar at current time, affirms having done the actions or not with simple clicking of yes or no button, and receives feedback on compliance over the previous week..

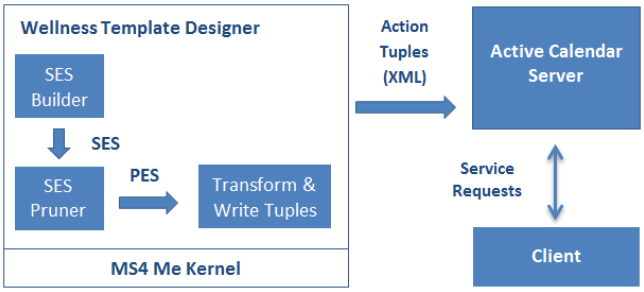


**Figure 3. Wellness Template Architecture**

A wellness template can be represented by an SES document including all required inputs from doctors, nutritionists, and other experts. As seen in Figure 3, the wellness template designer is based on SES builder, SES pruner, and transform&write tuples handled by MS4 Me™ kernel. The SES builder creates a rudimentary SES document from the requirements, and then lets users

enhance the SES with additional details. For example, the exercise plan has a location component which has sub-components: Park, RecreationCenter, Home, School, and Work (for more detail, see Exercise.ses in the Appendix). For a specific location of the Park component, users need to expand the SES document for the Park with actual nearby parks' information. Therefore, the SES builder provides tailored SES documents for the users. From users' selections of specializations in SES documents, the SES pruner generates PES documents containing pruning rules in a restricted natural language [3]. The transform&write tuples process has two steps: 1) generating a pruned tree structure through a transformation step, and 2) writing a set of tuples from the tree structure. The tuple comprises day of week, time slot, location slot, and action. The so-called action tuples are written in an XML document for a weekly plan. An Action Calendar Server creates a schedule table in a database using action tuples and provides web services in relation to wellness plan handling to respond to requests from users.

## 4. EXAMPLE OF WELLNESS TEMPLATE

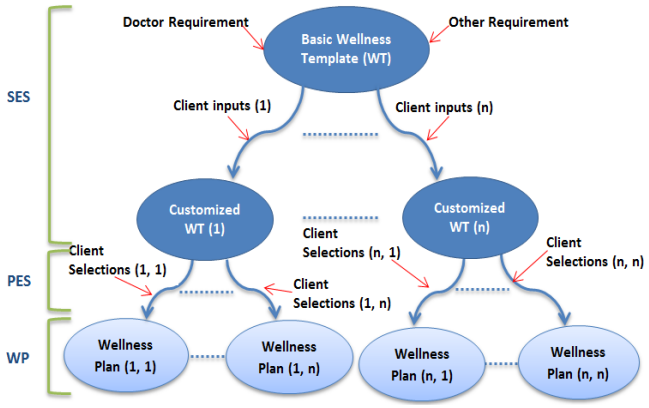


**Figure 4. Creating multiple SESs from the basic Wellness Template**

The wellness template is created as a basic wellness template (WT) with experts' requirements for a wellness plan. The basic WT contains general components for all users. However, the users want to have specific information for the general components. In the wellness template in our system, users can build their own WT with their inputs (adding entities in a specialization, and aspects to entities). With the customized WT (SES), users select components in specialization entities to produce a PES document with which action tuples are generated through transform&write tuples. As seen in Figure 4, multiple customized WT (SESs) are created from the basic wellness template (basic SES) through appending components of entities in the basic SES (client inputs). Each customized WT (SES) can have multiple PES documents, resulting in multiple wellness plans generated.

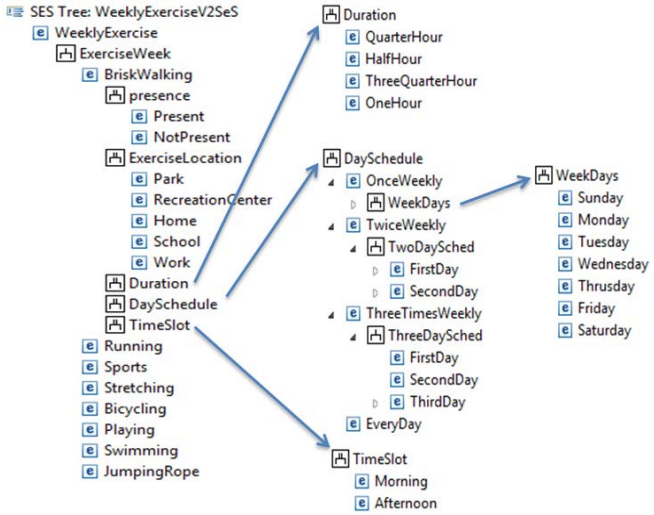


**Figure 5. Outline of a basic exercise plan**

For example, an exercise plan contains exercise items each of which has five specialization relationships (*presence, ExerciseLocation, Duration, DaySchedule, and Timeslot*). The *presence* specialization has Present and NotPresent entities used to select an exercise item. The ExerciseLocation specialization has Park, RecreationCenter, Home, School, and Work entities which indicate a place to work out. The DaySchedule specialization has OnceWeekly, TwiceWeekly, ThreeTimesWeekly, and EveryDay entities to select how often a user works out per week. The OnceWeekly entity has a WeekDays specialization to select the day in the week in which to exercise. The TwiceWeekly entity has a TwoDaySched aspect which consists of FirstDay and SecondDay entities having the WeekDays specialization. Likewise, the threeTimesWeekly has a ThreeDaySched aspect which comprises FirstDay, SecondDay, and ThirdDay entities having the WeekDays specialization. The TimeSlot specialization has Morning and Afternoon entities. Figure 5 illustrates an outline of the basic exercise plan using MS4 Me$^{TM}$. As the basic SES for the exercise plan shows, exercise items are fixed, and entities in the ExerciseLocation are not assigned to specific places for the users. The user can add exercise entities and specific place entities for the Park entity to the basic SES with the wellness template designer. It causes producing a new SES (customized Wellness Template).

The Exercise.ses in the appendix is an example of the customized Wellness Template which expands the Park entity and the RecreationCenter entity. The Park entity has a ParkSpecificLocation specialization which has FortLowell and Udall entities, and the RecreationCenter entity has a RecreationCenterSpecificLocation specialization which has

a LAFitness entity. Figure 6 illustrates selections from the Exercise SES in red. .

```
BriskWalking can be Present or NotPresent in presence!
BriskWalking can be Morning or Afternoon in TimeSlot!
BriskWalking can be QuarterHour,HalfHour,ThreeQuarterHour, or OneHour in Duration!
BriskWalking can be OnceWeekly, TwiceWeekly,ThreeTimesWeekly, or EveryDay in DaySchedule!
BriskWalking can be Park, RecreationCenter, Home, School, or Work in ExerciseLocation!
RecreationCenter can be LAFitness in RecreationCenterSpecificLocation!
Bicycling can be Present or NotPresent in presence!
Bicycling can be Morning or Afternoon in TimeSlot!
Bicycling can be QuarterHour,HalfHour,ThreeQuarterHour, or OneHour in Duration!
Bicycling can be OnceWeekly, TwiceWeekly,ThreeTimesWeekly, or EveryDay in DaySchedule!
Bicycling can be Park, RecreationCenter, Home, School, or Work in ExerciseLocation!
Park can be FortLowell or Udall in ParkSpecificLocation!
```

**Figure 6. Selection Example for Exercise.ses**

MS4 Me™ kernel creates a coupled model through a pruning process with the Exercise.ses and the Exercise.pes. The coupled model (WeeklyExercise) has two atomic models (for Bicycling and BriskWalking) which represent user's selections of entities. For example, Present_FortLowell_Park_Morning_ThreeQuarterHour_EveryDay_Bicycling, name of an atomic node, contains pruning information of the Bicycling exercise item. From the information of node names, elements of the tuple are built up for a week. For the Bicycling's name, a day of week is assigned to EveryDay, time slot is assigned to Morning, location slot is assigned to FortLowell Park, and action is assigned to Bicycling_for_ThreeQuarterHour.

To generate an action tuple XML, EveryDay information is mapped to 7 days in a week. Therefore, the user will ride a bike at FortLowell Park for three quarter hour in the every morning for a week. The action tuple XML contains 14 exercises actions for Bicycling and BriskWalking. Figure 7 shows xml tags for tuple's elements. A dayExercisePlan tag represents a tuple of a wellness plan and a day tag for the day of week contains a number for a day.

```
<dayExercisePlan>
    <day>3</day>
    <timeSlot>Afternoon</timeSlot>
    <locationSlot>LAFitness RecreationCenter</locationSlot>
    <action>BriskWalking_for_HalfHour</action>
</dayExercisePlan>
```

**Figure 7. A tuple in the action tuple XML**

## 5. IMPLEMENTATION OF PERSONAL WELLNESS PLANNING AND SELF MANAGEMENT ON CLOUD ENVIRONMENT

The wellness planning and self-management environment is implemented on Google App Engine which is a platform as a service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers [10, 11, 12]. To produce web applications, Java and Java-based web programing technology such as Servlet and JSP are employed [13]. For client web pages, HTML, Javascript, and Ajax are used. As for data storage, the Google App Engine provides Datastore to save data generated in the web applications. For action tuple generation, MS4 Me™ kernel libraries are used in the web server.

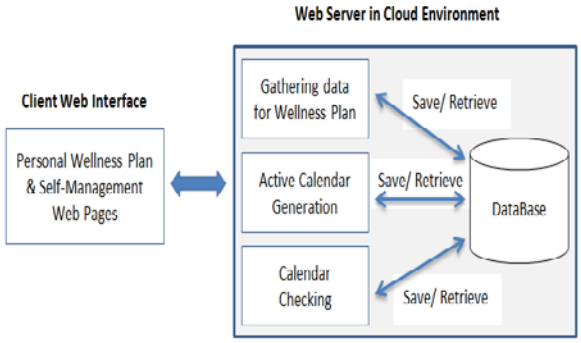


**Figure 8. Personal Wellness Plan and self-management Implementation on Cloud Environment**

As Figure 8 shows, in the web server, there are three functions (Gathering data for wellness plan, Active Calendar Generation, and Calendar Checking). Client web pages for designing wellness plan display basic wellness plan data which are saved in the database. They have a capability of adding entities to database. Through client web pages to design wellness plan, entities' information is retrieved from the database to show all possible choices for the wellness plan, and saved to the database to reflect user's inputs to the web pages. The first web server function (gathering data for wellness plan) is to design a customized wellness template and to select pruning entities in the customized wellness template. The pruning data are also saved in the database.

The second web service function is to generate active calendar based on saved data for the wellness plan and the pruning process. After finishing user's design and selection, an SES document and a PES document are dynamically generated using saved information and saved to the database. Afterwards, weekly action tuples are created using MS4 Me™ kernel libraries for the SES document and the PES document. The action tuples are saved into a schedule table which interacts with users through calendar checking services. The schedule table has a "done" column which indicates if the user performed the designated action or not. Default values for the "done" column are assigned zeroes. The column is set to one, after the user responds to the compliance question affirmatively. The third web service function is to check the calendar and monitor compliance rate for the wellness plan. A web page called "Daily Actions" shows the last action and enables responding as to whether it was done or not. A Compliance web page shows compliance rate for the previous week actions.

### 5.1. Design Wellness Plan Web Page for the Customized Wellness Plan

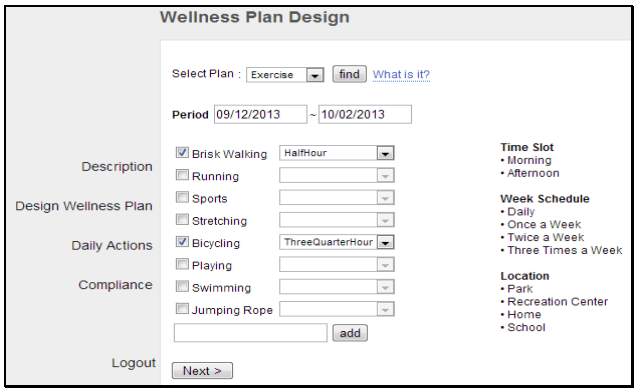


**Figure 9.1 Design Wellness Plan Web Page for Exercise Plan**

The design wellness plan web page for Exercise plan displays a plan period, exercise items with check boxes, and duration combo boxes. An "add" button enables a user to add extra exercise items to the Exercise plan. In the Figure 9.1, the user selected BriskWalking and Bicycling items with HalfHour and ThreeQuarterHour duration, respectively. Once finished with selecting the exercise items and their durations, the user needs to click the "Next" button to select the other items such as Time Slot, Week Schedule, and Location.

Figure 9.2 shows radio buttons for selected exercise items whose other items need to be selected independently. After selecting other items, the user should click a "Save Item" button to save selected information to the database. The Location item can have more children entities through adding a new entity. If the user configures the exercise plan correctly, the SUBMIT button is activated. The SES, PES, action tuples, and active calendar are generated by clicking the SUBMIT button. The full SES and PES documents for this example are in the appendix.

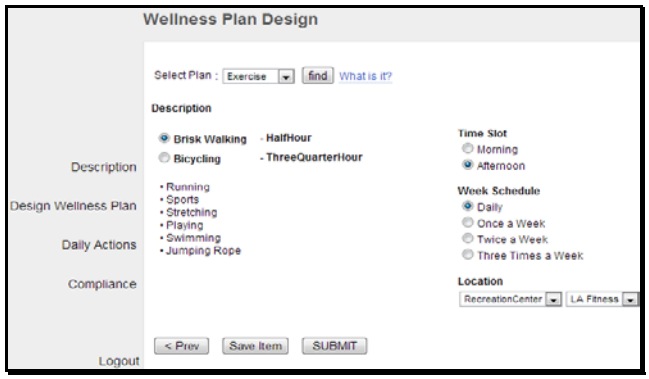


**Figure 9.2. Design Wellness Plan Web Page for Exercise Plan**

### 5.2. Daily Actions Web Page for the Exercise Plan

The Daily Actions web page seen in Figure 10 is used to show last action and next action, and record what actions the user took for the exercise plan. With clicking a "Check Action" button, the last action is displayed in "Did you do last action?" list component. If the user did the last action, he/she need to select the last action in the list and click the Yes button. If not, the user should click the No button to see the next action to do. The Yes button makes the active calendar save done actions. When either button is pushed or eventually in any case, the next action is shown in the "Next Action To Do" list component.

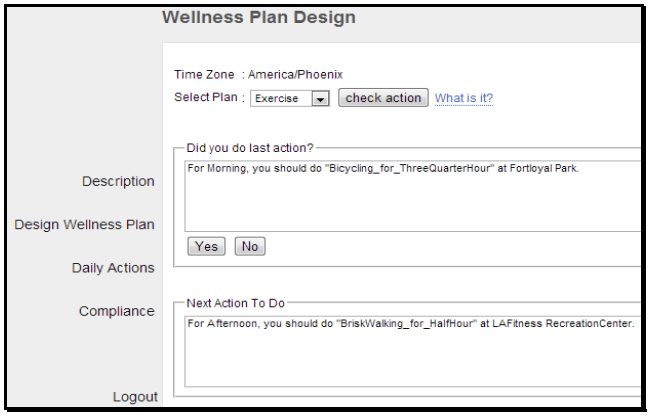


**Figure 10. Daily Actions Web Page for Exercise Plan**

### 5.3. Compliance Web Page for the Exercise Plan

As Figure 11 shows, The Compliance web page has a "check compliance" button to show compliance rate for the last week. The compliance rate is displayed under the "check compliance" button.

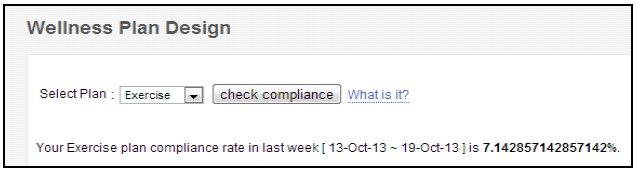


**Figure 11. Compliance Web Page for Exercise Plan**

## 6. CONCLUSIONS

Increasing the applicability of DEVS and SES for embedded M&S components will need improved concepts and tools for interfacing these core elements with multiple layers of hardware and software. While others have focused on embedding of DEVS components, in this paper we showed how the SES construction, pruning, and model generation process, can support web-based systems in cloud environments. Our focus was on the process by which modeling and simulation kernels are integrated within other layers of a web-application.

To exemplify such integration we detailed the implementation of a personal wellness plan and self-management system that provides web services for designing personal wellness plans, checking daily actions, and querying for compliance with the registered wellness plan. To generate the wellness plan we employed SES and PES documents with MS4 Me$^{TM}$ kernel libraries that extract and process selected information (pruning and model-transformation). The approach enables flexibility in both modification of the SES (to expand its elements of choice) as well as in the PES (to enable the user to make the sections from choices provided by the SES.) Employing web technologies, the SES and PES documents are dynamically created from user inputs through a browser. We discussed the approach taken to support user graphic and web service requirements to interface to the embedded SES/PES kernels.

In the future, information technology (IT) systems are expected to employ embedded M&S components to support their core information processes. Such systems will require interfacing M&S kernel systems with other functional layers as well as with end-users. The example discussed here points the way to how to integrate the SES, PES, and transformation functionalities with the larger system so that the combinatorial and model generation capabilities are fully exploited. The detailed presentation of the wellness template provides a first example of how to integrate DEVS and SES within a complex software/hardware system and thereby to increase their applicability to today's web and cloud- based information technology.

Many capabilities of the SES/PES/DEVS methodology as supported by MS4 Me$^{TM}$ remain to be exploited in future IT systems. For example, the wellness plan for diet could benefit from the capability to support hierarchical composition and component reuse. In particular, the DASH eating plan [15] provides recipes for the user to follow (e.g., soups, salads. dressings, etc.) so as to construct dishes that can become parts of meals that satisfy dietary specifications. In future work, we will extend the interfaces discussed here to enable employing the model base capability supported by MS4 Systems to store reusable components (SESs representing dishes) that can be hierarchically composed and pruned to form families of meal composites [6]. The user will be able to start with an SES and prune it from scratch, or start from an existing PES representing a meal variation that can be re-pruned with minimal work to generate a meal that fits the desired criteria. Another challenge will be integrating the wellness template system within the emerging encompassing health networks supporting maintenance and exchange of electronic health records.

In conclusion, the SES/PES/DEVS methodology offers a rich set of M&S kernels for integration into future IT systems and tools are available to support research in these directions.

**References**


[1] Zeigler, B.P., Kim, T.G., and Praehofer, H., Theory of Modeling and Simulation, 2nd ed., Academic Press, New York, 2000.

[2]Zeigler, B.P and Phillip Hammonds (2007), "Modeling&Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange", Academic Press, Boston,. 448 pages

[3] Zeigler, Bernard P., Sarjoughian, Hessam S. Guide To Modeling And Simulation Of Systems Of Systems Series: Simulation Foundations, Methods And Applications Springer Pub. Co., pp. 330, 2013.

[4] MS4 Me$^{TM}$ Software : www.ms4system.com

[5] Chungman Seo, Bernard P. Zeigler, Robert Coop, and Doohwan Kim, "DEVS Modeling and Simulation Methodology with MS4 Me Software Tool", Proceedings of the 2012 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium (TMS-DEVS), April, 2013, San Diego, CA.

[6] Bernard P. Zeigler, Chungman Seo, and Doohwan Kim, "System Entity Structures for Suites of Simulation Models", International Journal of Modeling, Simulation, and Scientific computing, Volume 04, Issue 03, September 2013.,

[7] http://www.eclipse.org/home/categories/rcp.php

[8] MS4 Me user guide (www.ms4system.com)

[9] Bernard P. Zeigler, Ernest Carter, Chungman Seo, Cynthia K Russell, and Brenda A. Leath, "Methodology and Modeling Environment for Simulating National Health Care" 2012 Autumn Simulation Multi-Conference (AutumnSim'12) October 28-31, 2012 San Diego, CA,

[10]Voorsluys, William; Broberg, James; Buyya, Rajkumar (February 2011)."Introduction to Cloud Computing". In R. Buyya, J. Broberg, A.Goscinski. *Cloud Computing: Principles and Paradigms*. New York, USA: Wiley Press. pp. 1–44. ISBN 978-0-470-88799-8.

[11]E. Keller and J. Rexford. The "Platform as a Service" Model for Networking. In Proc. INM WREN, 2010
Google app engine :

[12]http://en.wikipedia.org/wiki/Google_App_Engine

[13]JSP : http://www.oracle.com/technetwork/java/javaee/jsp/index.html

[14]Rodrigo Castro, Iván Ramello, Gabriel A. Wainer, "M&S-Based Design of Embedded Controllers on Network Processors", Proceedings of 2012 Spring Simulation Conference (SpringSim12), DEVS/TMS Symposium, March 2012

[15] http://www.nhlbi.nih.gov/health/public/heart/hbp/dash/new_dash.pdf

## Appendix

### Exercise.ses

**From the** ExerciseWeek **perspective,** WeeklyExercise **is made of** BriskWalking, Running, Sports, Stretching, Bicycling, Playing, Swimming, **and** JumpingRope!

BriskWalking **can be** Present **or** NotPresent **in** presence!
Running **is like** BriskWalking **in** presence!
Sports **is like** BriskWalking **in** presence!
Stretching **is like** BriskWalking **in** presence!
Bicycling **is like** BriskWalking **in** presence!
Playing **is like** BriskWalking **in** presence!
Swimming **is like** BriskWalking **in** presence!
JumpingRope **is like** BriskWalking **in** presence!

BriskWalking **can be** OnceWeekly, TwiceWeekly,ThreeTimesWeekly, **or** EveryDay **in** DaySchedule!
Running **is like** BriskWalking **in** DaySchedule!
Sports **is like** BriskWalking **in** DaySchedule!
Stretching **is like** BriskWalking **in** DaySchedule!
Bicycling **is like** BriskWalking **in** DaySchedule!
Playing **is like** BriskWalking **in** DaySchedule!
Swimming **is like** BriskWalking **in** DaySchedule!
JumpingRope **is like** BriskWalking **in** DaySchedule!
OnceWeekly **can be** Sunday, Monday, Tuesday, Wednesday, Thrusday, Friday, **or** Saturday **in** WeekDays!
**From the** TwoDaySched **perspective,** TwiceWeekly **is made of** FirstDay **and** SecondDay!
**From the** ThreeDaySched **perspective,** ThreeTimesWeekly **is made of** FirstDay,SecondDay, **and** ThirdDay!
FirstDay **is like** OnceWeekly **in** WeekDays!
SecondDay **is like** OnceWeekly **in** WeekDays!
ThirdDay **is like** OnceWeekly **in** WeekDays!

BriskWalking **can be** Park, RecreationCenter, Home, School, **or** Work **in** ExerciseLocation!
Running **is like** BriskWalking **in** ExerciseLocation!
Sports **is like** BriskWalking **in** ExerciseLocation!
Stretching **is like** BriskWalking **in** ExerciseLocation!
Bicycling **is like** BriskWalking **in** ExerciseLocation!
Playing **is like** BriskWalking **in** ExerciseLocation!
Swimming **is like** BriskWalking **in** ExerciseLocation!
JumpingRope **is like** BriskWalking **in** ExerciseLocation!

BriskWalking **can be** Morning **or** Afternoon **in** TimeSlot!
Running **is like** BriskWalking **in** TimeSlot!
Sports **is like** BriskWalking **in** TimeSlot!
Stretching **is like** BriskWalking **in** TimeSlot!
Bicycling **is like** BriskWalking **in** TimeSlot!
Playing **is like** BriskWalking **in** TimeSlot!
Swimming **is like** BriskWalking **in** TimeSlot!
JumpingRope **is like** BriskWalking **in** TimeSlot!

BriskWalking **can be** QuarterHour,HalfHour,ThreeQuarterHour, **or** OneHour **in** Duration!
Running **is like** BriskWalking **in** Duration!
Sports **is like** BriskWalking **in** Duration!
Stretching **is like** BriskWalking **in** Duration!
Bicycling **is like** BriskWalking **in** Duration!
Playing **is like** BriskWalking **in** Duration!
Swimming **is like** BriskWalking **in** Duration!
JumpingRope **is like** BriskWalking **in** Duration!

Park **can be** FortLowell **or** Udall **in** ParkSpecificLocation!
RecreationCenter **can be** LAFitness **in** RecreationCenterSpecificLocation!

### Exercise.pes

**select** Afternoon **from** TimeSlot **for** BriskWalking!
**select** HalfHour **from** Duration **for** BriskWalking!
**select** RecreationCenter **from** ExerciseLocation **for** BriskWalking!
**select** LAFitness **from** RecreationCenterSpecificLocation **for** RecreationCenter **under** BriskWalking!
**select** EveryDay **from** DaySchedule **for** BriskWalking!
**select** Morning **from** TimeSlot **for** Bicycling!
**select** ThreeQuarterHour **from** Duration **for** Bicycling!
**select** Park **from** ExerciseLocation **for** Bicycling!
**select** FortLowell **from** ParkSpecificLocation **for** Park **under** Bicycling!
**select** EveryDay **from** DaySchedule **for** Bicycling!
**select** Present **from** presence **for** BriskWalking!
**select** NotPresent **from** presence **for** Running!
**select** NotPresent **from** presence **for** Sports!
**select** NotPresent **from** presence **for** Stretching!
**select** Present **from** presence **for** Bicycling!
**select** NotPresent **from** presence **for** Playing!
**select** NotPresent **from** presence **for** Swimming!
**select** NotPresent **from** presence **for** JumpingRope**!**