ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА ДЛЯ ВАЛИДАЦИИ ИМИТАЦИОННЫХ МОДЕЛЕЙ

Е.Б. Замятина, Д.А. Чурин (Пермь)

Экспертизе построенной исследователями имитационной модели уделяется большое внимание, как в отечественной, так и в зарубежной литературе [1, 2, 3]. Экспертиза подразумевает выполнение этапов верификации и валидации концептуальной имитационной модели для доказательства того, что этой модели можно доверять [3], что качество информации, полученной в результате имитационного эксперимента, соответствует тому уровню, который позволяет принять правильное решение.

Итак, верификация сводится к контролю корректности переноса концептуальной модели, разработанной исследователем, в имитационную среду [4]. Валидацию можно определить как проверкуправильности поведения и представления концептуальной модели. Верификация является частью валидации, ее инструментом. Верификация проводится во время построения модели, валидация же осуществляется непосредственно после завершения создания, описания модели на конкретном специализированном языке программирования. Однако на практике, как правило, процессы верификации, валидации, а также, тестирования и реализации модели пересекаются по времени.

Наряду с понятиями валидации и верификации в литературе приводится такое понятие, как аккредитация (VV&A- Validation, Veryfication and Accreditation). Аккредитация определяется как «официальное засвидетельствование того, что модель, симуляция, или объединение моделей и симуляций является допустимым для использования для определенной цели» [5].

Аккредитация – это официальное свидетельство заказчика модели (исследователя, системного аналитика), которое заключается в том, что имитационная модель применима для решения конкретной задачи. Верификация и валидация должны отвечать на вопрос: обладает ли модель достаточной точностью. При этом следует учесть цель разработанной имитационной модели.

К примеру, большинство демонстрационных моделей не обладают высокой точностью, однако стоит помнить, что их целью является демонстрация работы исследуемого процесса, объекта или системы только в общих чертах. Если модель справляется с поставленной перед ней целью, то можно говорить о ее соответствии прототипу даже при том, что точность модели невелика. Следовательно, цель создания модели должна быть известна до проведения ее валидации.

Следует отметить, что существуют различные виды валидации [5]: (а) валидация концептуальной модели (проверка степени детализации модели);(б) валидация данных (проверка точности данных); (в) валидация по методу белого ящика (детальная (микро) проверка модели, определение точности компонентов модели); (г) валидация по методу черного ящика (общая (макро) проверка работы модели, при которой определяется, обеспечивает ли общая модель достаточно точное представление о системе реального мира). Более подробно методы верификации и валидации описаны в работе американского исследователя Османа Балчи [6].

Один из них - проверка построения концептуальной имитационной модели. Разрабатываемая концептуальная модель описывает те компоненты системы реального мира, которые должны быть включены в модель (и те, которые должны быть исключены из конечной модели), и выражается либо формально (например, с помощью диаграмм циклов деятельности), либо неформально (например, в виде списка элементов).

Чтобы разработать концептуальную модель, разработчики должны проанализировать всю полученную информацию и прийти к оптимальному решению. Для проверки концептуальной модели могут быть использованы спецификация проекта или техническое задание. Кроме того, необходима оценка экспертов, разбирающихся в предметной области и подобных системах, о соответствии концептуальной модели требованиям, описанным в документации. Итак, этап

верификации и валидации должны выполняться совместно, как разработчиками модели, так и заказчиками, которым необходимо решить конкретную задачу.

Будем различать структурную и операционную валидацию. Структурная предполагает правильный набор элементов концептуальной модели и связей между ними, операционная валидность предполагает правильное функционирование имитационной модели.

Авторы настоящей статьи предлагают разработать программное обеспечение, реализующее онтологический подход к верификации и валидации. Прежде всего, рассмотрим структурную валидность.

Структурная валидность имитационной модели

Программное обеспечение включает портал, с помощью которого выполняется взаимодействие заказчиков и разработчиков модели. Заказчики должны представить онтологию концептуальной модели (О1), разработчики также представляют свою версию онтологии концептуальной модели (О2). Используя алгоритм определения близости онтологий, выполняют сравнение двух онтологий и выявляют несоответствия в них.

В качестве инструмента имитационного моделирования была выбрана программная система Triad.Net[8]. Имитационная модель в Triad.Net представлена тремя слоями: слоем структур (описание элементов модели и связей между ними), слоем рутин (описание сценариев поведения элементов модели), слоем сообщений (структур сложных сообщений, которыми обмениваются элементы модели). В системе имитационного моделирования Triad.Net активно используется онтологический подход, который используется на различных этапах имитационного эксперимента[8]. В частности, используя онтологии, выполняется настройка на конкретную предметную область.

Обратимся к исследованию SDN – сетей. SDN-сеть – это программно-конфигурируемая сеть (SDN от англ. Software-defined Networking). SDN-сеть - сеть передачи данных, в которой уровень управления отделён от устройств передачи данных и реализуется программно [7].

Базовая онтология системы моделирования Triad.Net была доработана и разработчики модели смогли оперировать такими объектами, как маршрутизатор, рабочая станция, суперузел, узел, маршрутизатор и т.д.[8]. Рассмотрим конкретную модель – модель SDN-сети, в которой реализован конкретный алгоритм SBARC [7]. Для реализации алгоритма SBARC необходимо добавить в базовую онтологию подкласс «Узел» (SDNNode).

В Triad.Net модель строят с помощью графического редактора, каждый узелдля этой конкретной модели имеет определенный семантический тип («Узел»).

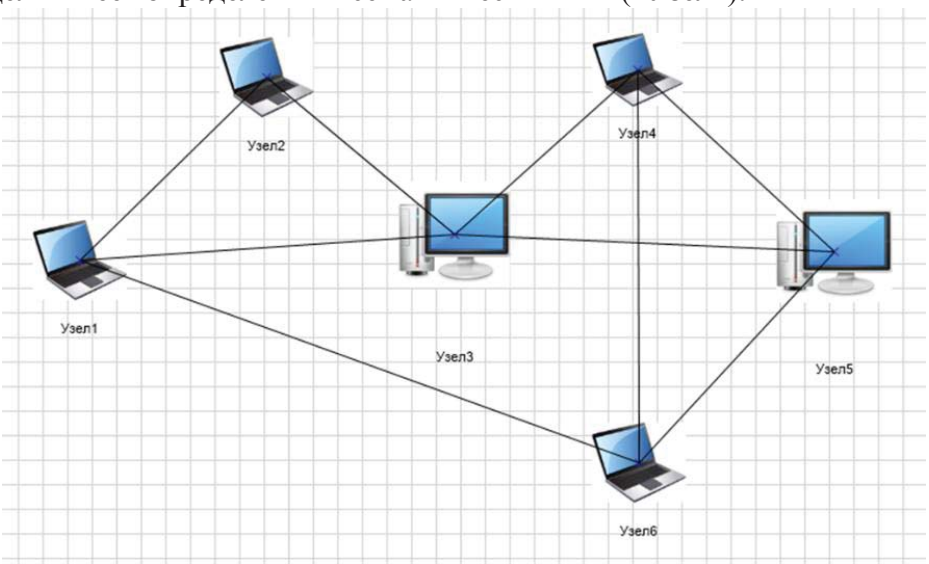

Рис.1. Модель SDN-сети для модели, реализующей алгоритм SBARC

Поведение элементу задают через контекстное меню, все возможные варианты поведения определяют из онтологии. В онтологии хранятся все экземпляры рутин данного семантического

типа. Каждая из рутин имеет некоторое количество параметров, которые пользователь может изменять после наложения рутины на элемент.

Алгоритм SBARC реализован на языке Triad рутиной SBARCRouter. Рутина SBARCRouter посылает сообщения случайно выбранному узлу в сети. Рутины представляет собой последовательность событий, запланированных на определенный момент времени. Определены несколько событий, такие как: а) посылка сообщения случайно выбранному узлу; б) посылка ответного сообщения на запрос суперузла, выполняющего функции маршрутизатора. Входное сообщение в зависимости от текущего режима работы и от вида узла может рассматриваться как параметры соседнего узла (идентификатор, вид узла и производительность (если сообщение от суперузла)), сообщения от других узлов. Модели соответствует онтограф на рис.2.

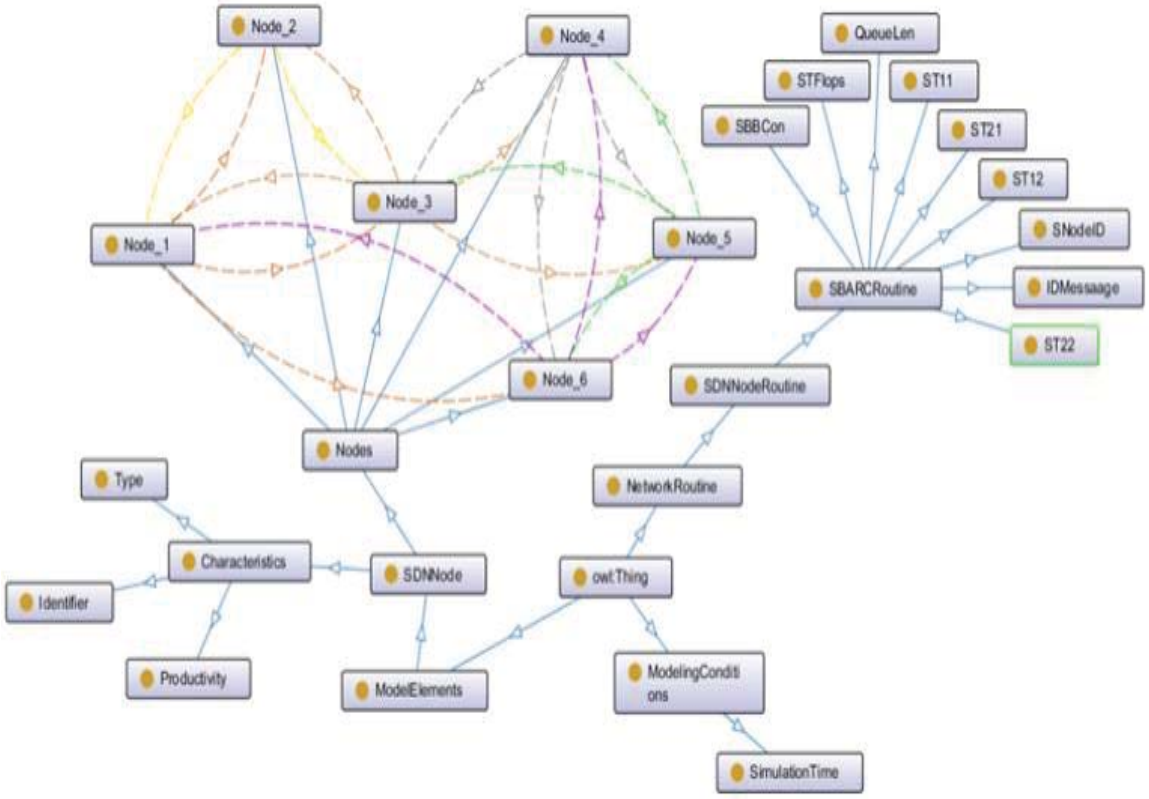

Рис. 2. Онтограф для модели, реализующей алгоритм SBARC

Теперь надо построить онтологию концептуальной модели, предложенной заказчиком, а затем определить степень их близости.

Для построения онтология необходимо прибегнуть к средствам извлечения знаний, а именно, методу REX (метод прямого извлечения знаний с помощью средств DataMining и TextMining из регламентов и документов) или MASK (метод сбора знаний, основанный на интервьюировании экспертов).

Воспользуемся методом MASK [9], чтобы наиболее точно определить требования эксперта. Представление модели, получаемой в ходе применения метода MASK, является схожим с классическими графовыми представлениями знаний в понятном для эксперта виде. При этом рекомендуется, чтобы построение модели знаний производилось непосредственно в диалоге с экспертом, чтобы он мог видеть модель и вносить свои корректировки, выражая свое видение изучаемого объекта. Сбор знаний по методу MASK помогает эксперту сосредоточиться на своей предметной области, чтобы описать ее, подчеркнув основные характеристики.

В ходе интервьюирования эксперта с целью создания онтологии концептуальной модели, ему были заданы следующие вопросы, в частности: (а) «какие элементы сети используются для построения модели?»; (б) «сколько элементов сети каждого вида используется в модели?» и т.д.

Также был составлен онтограф для построенной онтологии концептуальной модели (см. рис. 3).

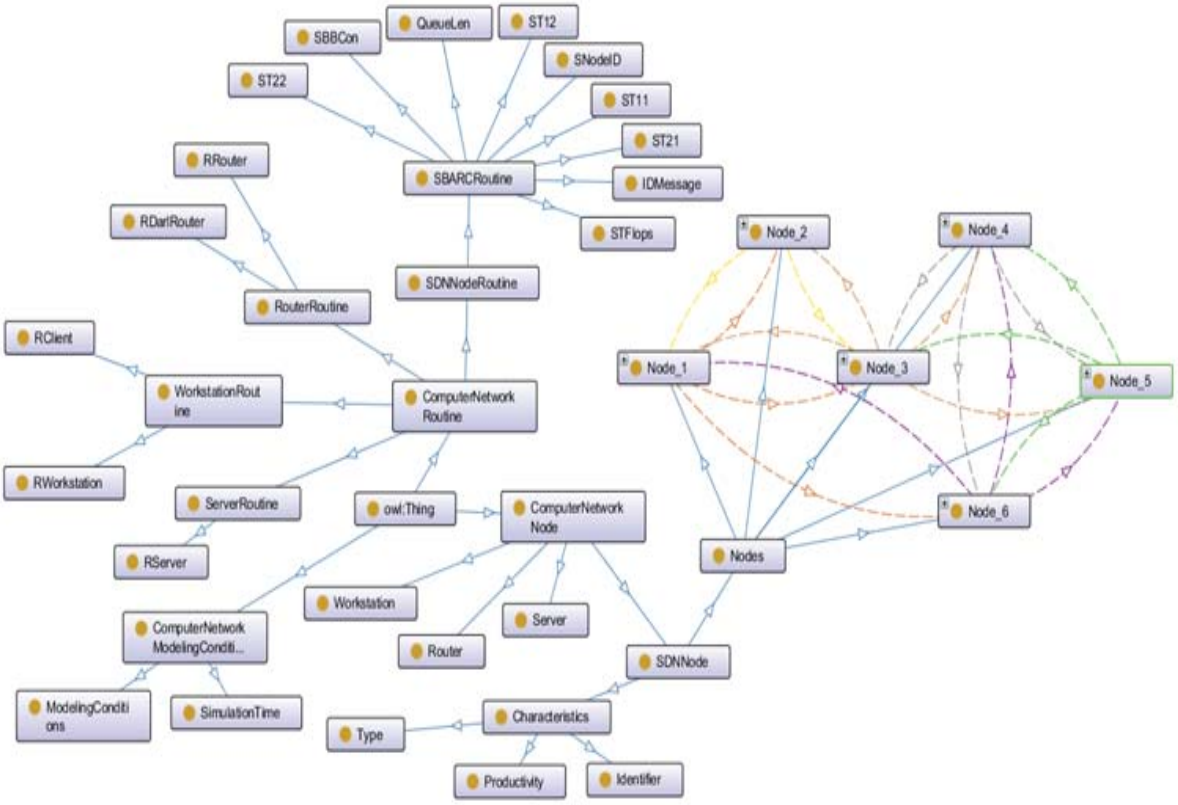

Рис. 3. Онтограф концептуальной модели

Заключительным этапом исследования является сравнение (определение меры близости) онтологий, построенных разработчиком и заказчиком (проверка соответствия логики концептуальной модели). В идеальной ситуации они должны быть абсолютно идентичны, но, на практике, это практически недостижимо из-за возможных особенностей специальных сред моделирования. При сравнении онтологий будем использовать подход, описанный в исследовании [10].

В методе выделяется 5 шагов: (а) определение наборов O1\O2 (набор классов, представленных в онтологии O1 и не представленных в онтологии O2); O2\O1 (набор классов, представленных в онтологии O2 и не представленных в онтологии O1), O1^O2 (набор классов, представленных в обеих онтологиях); (б) оценка меры семантической близости между понятиями каждого набора классов; (в) расширение онтологий O1 и O2, используя набор O1/\O2.

На этом шаге, для каждого класса X из O1/\O2 мы ищем их потомков Yiв O1 (соответственно в O2) и добавляем их как потомков класса X в O2 (соответственно в O1), если они не существуют в конкретной онтологии. (г) Определение классов O'1∧O'2, которые представляют собой набор общих понятий двух онтологий O'1 и O'2; (д) оценка сходства между онтологиями. В результате поведенных вычислений близость двух онтологий составляла 81%. Полученное значение можно объяснить тем, что онтология O1 была построена на базе понятий среды имитационного моделирования Triad.Net, которые не учитывались при построении заказчиком онтологии O2 концептуальной модели.

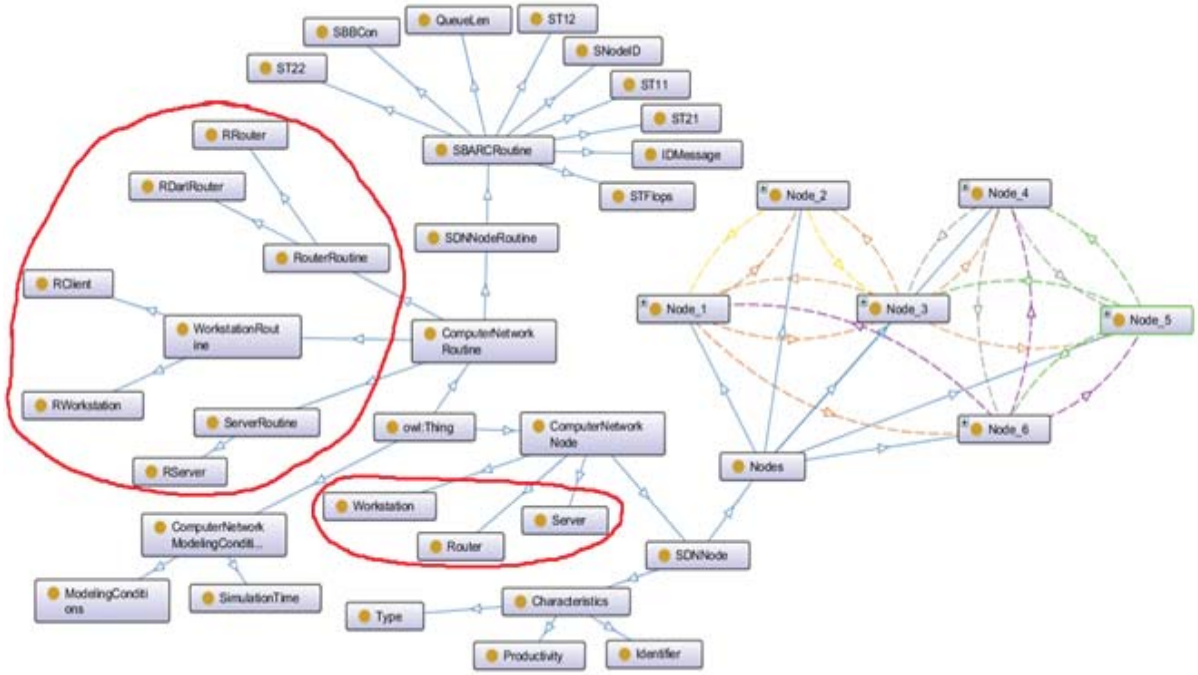

Рис. 4. Несоответствия, выявленные при определении семантической близости двух онтологий

Исключив из онтологии О1 классы, отражающие особенности среды моделирования Triad.Net (см. рис. 4), такие как: RouterRoutine, WorkstationRoutine, ServerRoutine. Исключим также соответствующие элементы сети: Server, Router, Workstation. Затем снова вычислим меру близости двух онтологий. В результате вычислений получаем, что мера близости двух онтологий равна 100%, что является идеальным и желаемым результатом при валидации имитационной модели. Итак, модель М1 (модель, построенная заказчиком) является валидной, т.е. соответствует ожиданиям и требованиям эксперта, изложенным в концептуальной модели. Если бы онтологии и после корректировки онтологии сетевой модели не были идентичны, то это бы означало наличие ошибок в реализации компьютерной модели, при этом различия в строении онтологий указывали бы конкретно на участок модели, в котором допущена ошибка.

Метод проведения экспертизы концептуальной имитационной модели, представленный выше, может быть полезен при разработке сложных имитационных моделей.

Операционная валидность имитационной модели

Операционную валидность имитационной модели (правильность функционирования модели) предполагается устанавливать с использованием специального программного обеспечения.

Объекты имитационной модели во время имитационного прогона управляются специальным алгоритмом. Назовем его алгоритмом имитации.

Для сбора информации о поведении имитационной модели в системе имитационного моделирования Triad используют «алгоритм исследования», который включает «информационные процедуры» и «условия моделирования». Информационные процедуры по сути дела являются «датчиками», которые следят за изменениями переменных во время имитационного прогона. Кроме того, информационные процедуры следят за последовательностью выполнения событий и фиксируют поступления входных сообщений в «рутины».

«Условия моделирования» определяют алгоритм исследования, поскольку включают список информационных процедур, используемых для сбора информации о поведении модели. Алгоритм исследования отделен от модели, существует возможность оперативно изменить его, используя другие «условия моделирования» с иным списком информационных процедур.

Информационные процедуры могут быть описаны следующим образом:

information procedure<имя>(<список настроечных параметров>)(<входные и выходные формальные параметры)

initial <последовательность операторов> endi

<последовательность операторов> processing<последовательность операторовs>…endinf

Вчасти «processing» выполняется итоговая обработка данных (определение суммы собранных ранее данных, среднего и т.д.)

Условия моделирования описывают следующим образом:

Conditions of simulation <*имя*> (<список настроечных параметров>) (<входные и выходные формальные параметры>) initial <последовательность операторов> *endi* <список имен информационных процедур> <последовательность операторов> *processing* <последовательность операторов> …*endcond*

В части*processing*возможна обработка результатов нескольких информационных процедур.

На рис. 5 представлен фрагмент графического редактора и форма для редактирования информационной процедуры.

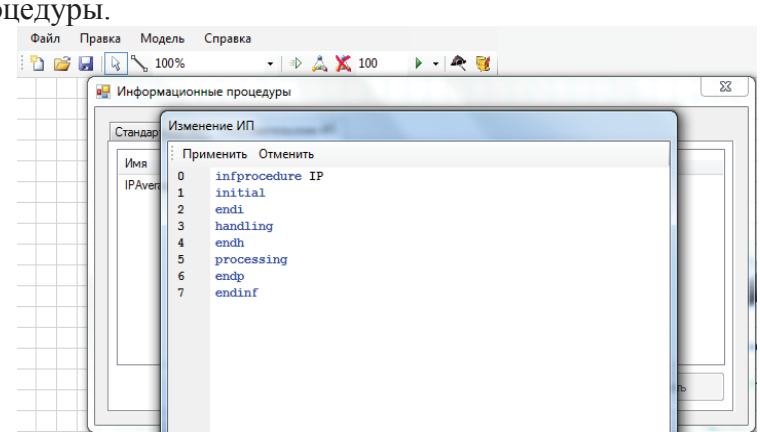

Рис. 5. Форма для редактирования информационной процедуры

Рассмотрим более подробно какие ошибки в функционировании имитационной модели можно обнаружить с помощью информационных процедур:

- неверные задержки времени,
- неверная передача данных,
- семантическая некорректность в преобразовании сигналов,
- семантическая некорректность обмена данными,
- семантическая некорректность изменения состояний имитационной модели и т.д.,

Приведем примеры информационных процедур, с помощью которых можно определить некорректность поведения имитационной модели вовремя имитационного прогона. Пусть нам необходимо убедиться, что сообщение, посланное с одного вычислительного узла должно за определенный промежуток времени быть доставлен на другой вычислительный узел. Значение сообщения должно иметь определенную величину и должен поступить на определенный входной полюс.

Информационная процедура *more_interval (t)[e1, e2]*, к примеру, фиксирует время между двумя событиями e1 (может быть, событие, которое соответствует посылке сообщения) и e2 (событие, связанное с доставкой сообщения).

Неверные временные задержки могут быть определены информационными процедурами *all_events(e)* (процедуры определяют начало и завершение события), *all_changes (var)* (процедура определяет моменты времени, когда указанная переменная изменяет свое значение).

С помощью информационной процедуры Investigator may use information procedure *schedule_event (e)* (the names of all registered events which preceded the indicated one); *inspect_change (e)* (the values of all variables which were changed before the event *e* occurrence, event *e* is an actual argument of procedure).

Перечисленные выше информационные процедуры являются стандартными. Система моделирования Triad располагает лингвистическими и программными средствами для разработки уникальных информационных процедур. Рассмотрим пример информационной процедуры,

которая определяет была ли выполнена последовательность событий во время функционирования имитационной модели:

```
information procedure event_sequence (in ref event E1,E2,E3;out Boolean arrived)
initial interlock (E2,E3); Arrived := false;
case of e1:available(e2);
        e2:available( E3):
e3:ARRIVED:=true;
endc
endinf
```

Приведенная выше процедура определяет, была ли выполнена последовательность событий E1→E2→E3. Оператор *interlock* блокирует входной параметр (event E1). Это означает, что информационная процедура не следит за этим параметром. Оператор *available* вновь устанавливает наблюдение за параметром.

Данные, полученные с помощью информационных процедур, следует сравнить с данными, которые предоставляет исследователь (заказчик имитационной модели). Данные заказчика хранятся в онтологии, которая может быть получена в результате интервьюирования.

Заключение

В статье представлены лингвистические и программные средства системы автоматизированного проектирования и имитационного моделирования Triad, которые позволяют выполнять валидацию и верификацию имитационной модели. Для структурной валидации предлагается использовать онтологический подход, который заключается в построении онтологий исследователем (заказчиком) и разработчиком имитационной модели (с помощью определенных инструментальных средств, в нашем случае, это система моделирования Triad) с последующим определением их семантической близости. Онтологию исследователя (заказчика) предполагается строить по результатам интервьюирования с помощью метода MASK.

Для операционной валидности необходимо использовать механизмы системы имитации Triad, а именно, информационные процедуры. Информационные процедуры фиксируют моменты времени выполнения событий, изменения переменных и т.д. Результаты выполнения информационных процедур сравниваются с онтологиями, построенными заказчиком.

Предложенный подход позволяет автоматизировать процедуры валидации и верификации имитационных моделей.

Благодарности

Работа выполнена при финансовой поддержке гранта РФФИ№ 19-47-230003 и Администрации Краснодарского края.

Литература

1. Кельтон В. Д., Лоу А. М. Имитационное моделирование. Классика CS. 3-е изд.СПб.:БХВ-Питер, 2004.-887 с.
2. Соколов Б.В., Юсупов Р.М.. Концептуальные и методические основы квалиметрии моделей и полимодельных комплексов //Труды СПИИРАН. Институт информатики и автоматизации. СПб.: Наука, 2004. С.10-35.
3. Электронный ресурс: Яцкив И. В. Проблема валидации имитационной модели и ее возможные решения. http://gpss.ru/immod'03/043.html
4. Sargent R.G. An Introductory Tutorial On Verification And Validation Of Simulation Models // Proceedings of the 2015 Winter Simulation Conference L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, M. D. Rossetti, eds. 2015. P.1729-1740.
5. Sargent R.G. Verification and Validation of Simulation Models» // Proceedings of the 2007 Winter Simulation Conference S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds. 2007. P. 124–137.
6. Balci O. Quality Assessment, Verification, And Validation Of Modeling And Simulation Applications // Proceedings of the 2004 Winter Simulation Conference R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds. 2004. P. 246-249.

7. Zhiyong X., Yiming H. SBARC: A Supernode Based Peer-to-Peer File Sharing System» // Department of Electrical & Computer Engineering and Computer Science University of Cincinnati. Cincinnati. 2008. P. 1-12
8. Zamyatina E., Maltcev G., Mikov A. Simulation of Routing Algorithm for SDN and SON Networks, in: 2017 IEEE 11th International Conference on Application of Information and Communication Technologies Vol. 2. Institute of Electrical and Electoronics Engineers, 2017. pp. 79-84.
9. Matta N., Ermine J.L., Aubertin G., Trivin J.Y. Knowledge Capitalization with a Knowledge Engineering Approach: The Mask Method // Knowledge Management and Organizational Memories. Springer, Boston, MA. 2002. P. 17-28. doi: 10.1007/978-1-4615-0947-9_2.
10. Ngom A., Kamara-Sangare F.Assessing Similarity Value between Two Ontologies» // IC3K 2018. Proceedings of the 10th International Joint Conference on Knowledge Discovery, 2018. P. 343–350.