

Верификация имитационной модели методом оценки трудоемкости алгоритма

А. И. ЯКИМОВ

Введение

Методология имитационного моделирования ERP-системы управления промышленным предприятием [1] предполагает комплексное использование современных информационных технологий, включает в себя метод, представляющий собой адаптацию общей методики создания и использования имитационных моделей сложных систем применительно к производственно-экономической деятельности промышленного предприятия. В основу метода положены следующие идеи: создание и использование базовой имитационной модели (БИМП) промышленного предприятия [2]; представление промышленного предприятия в виде многоуровневой иерархической системы с активными элементами принятия решений на каждом из уровней иерархии; в) использование технологии UML и проведение функционального анализа на основе технологии IDEF0 для построения вербальной и концептуальной модели объекта исследования; г) применение объектно-ориентированного анализа и проектирования для составления программы имитационной модели на основе процессного способа имитации; д) использование технологии XML для ввода данных в модель из комплексной информационной системы; е) выбор варианта решения на соответствующем уровне управления в условиях неопределенности и риска; ж) использование иерархических уровней адаптации системы управления для приспособления ее к специфическим свойствам объекта и окружающей среды.

Одним из важных вопросов при использовании имитационного моделирования в ERP-системе управления является верификация БИМП [3].

Процесс «Планирование производства» в БИМП

Деятельность служб предприятия по составлению плана-графика производства включает непосредственно составление плана-графика производства; планирование потребности в ресурсах; план-график запуска продукции в производство. Планирование производства в модели представлено процессом *CMasterProductionScheduling* (рисунок 1) в программной реализации модели на языке C++. Процесс «Планирование производства» отражает функциональные действия по составлению плана-графика производства. Планирование осуществляется на будущий период τ_{mps} по интервалам τ_{mpsu} . Поскольку история планирования сохраняется, то первой выполняется подготовка структур данных плана производства: добавляется новая ячейка и очищаются предыдущие данные за период $(\tau_{mps} - \tau_{mpsu})$. Далее формируется новый план производства каждого вида продукции, исходя из объемов запланированных отгрузок в каждом интервале планирования. Полученный план производства корректируется в меньшую сторону на величину запасов готовой продукции на складе и определяется потребность в ресурсах для его выполнения в каждом интервале τ_{mpsu} . Последним выполняется оператор синхронизации $WAIT(\tau_{mpsu})$, по которому данному процессу назначается момент следующей активизации по окончании ожидания длительностью τ_{mpsu} , соответствующей текущему интервалу планирования. План корректируется, исходя из запасов продукции на складе. Затем формируется план запуска производства продукции, исходя из длительности производственного цикла. При дальнейшем планировании определяется объем продукции, который должен быть в производстве и который необходимо произвести в ближайших периодах планирования в пределах длительности производственного цикла. Разница между фактиче-

ским и требуемым объемом незавершенного производства распределяется по периодам планирования. На величину избытка последовательно уменьшается планируемый запуск в ближайших периодах, начиная с текущего. В дальнейшем определяются потребности в ресурсах. Полагается, что все ресурсы должны быть в наличии на момент запуска в производство.

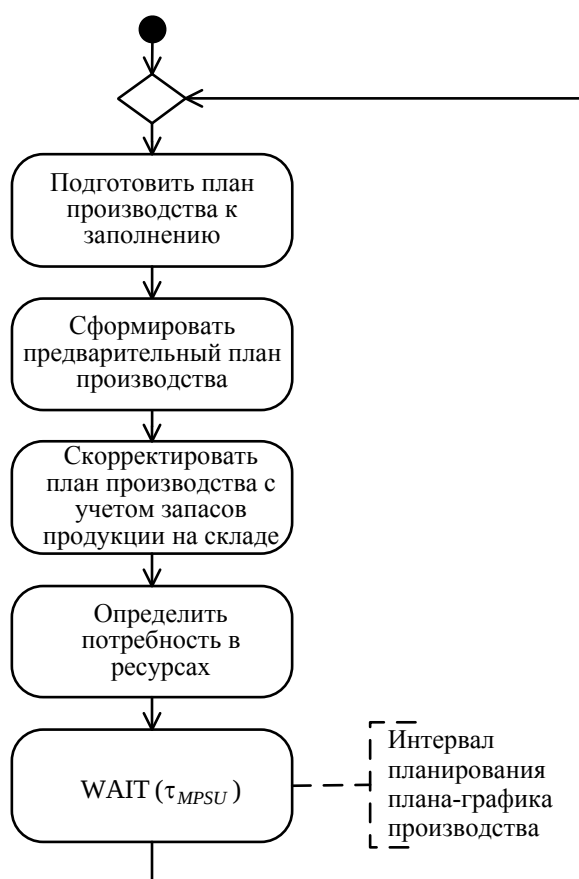


Рисунок 1 – Схема алгоритма процесса «Планирование производства»

В алгоритме процесса *CMasterProductionScheduling* (планирование производства) используются следующие переменные:

MPSTimeUnit – интервал планирования основного плана-графика производства;
MPSPanningHorizon – количество интервалов в основном плане-графике производства;
MPS[][] – основной план-график производства;
launchingSchedule[][] – график запуска в производство;
resourceSchedule[][] – график потребности в ресурсах;
numberOfProductTypes – количество видов продукции;
numberOfResourceTypes – количество видов ресурсов;
shipmentVolumeDistributions[] – параметры распределения объемов продукции в отгрузке;
shipmentIntervalDistributions – параметры распределения интервала между отгрузками продукции;
specificResourceConsumption[][] – нормы расходов ресурсов на единицу продукции;
shipments[] – список отгрузок продукции;
stock[] – запасы продукции на складе.

Верификация процесса «Планирование производства»

Д. А. Садовски (D. A. Sadowski, 2004) [4, с. 61–66] отмечает, что к провалу проектирования имитационной модели приводит недооценка трех основных задач: сбор данных; проверка модели; анализ прогонов модели. К. М. Оверстрит (C. M. Overstreet, 2002) [5, с. 642]

обращает внимание на общую проблему тестирования – обнаружение непредвиденных взаимодействий между компонентами. В еще большей степени это относится и к БИМПП, поскольку функционирование модели состоит из последовательности выполнения компонентов, чей порядок определен разработчиком модели, программной реализацией имитационной модели и случайными числами. До настоящего времени не существует единого метода для сбора и представления взаимодействия элементов в ходе имитационного моделирования. Например, Т. Ф. Брэди (Т. F. Brady, 2005) [6, с. 285-289] предлагает проводить интроспективный анализ. При этом исследуются объекты и ресурсы модели, заданные ключевыми словами, с помощью программного анализатора частоты, который определяет распределение частоты ключевых слов и корреляцию между ними.

Для решения задачи верификации и валидации модели предлагается метод на основе оценки трудоемкости алгоритма процесса модели, рассмотренный на примере процесса «Планирование производства».

Под трудоемкостью алгоритма $F_A(N)$ для данного конкретного входа N понимают количество «элементарных» операций, совершаемых алгоритмом для решения конкретной проблемы в данной формальной системе [7]. Пусть D_A – множество конкретных проблем данной задачи, заданное в формальной системе. В общем случае существует собственное подмножество D_N множества D_A , включающее все конкретные проблемы, имеющие мощность N :

$$D_N = \{D \in D_A \mid |D| = N\}.$$

Тогда содержательно данный алгоритм, решая различные задачи размерности N , будет выполнять в каком-то случае наибольшее количество операций, а в каком-то случае наименьшее количество операций. Введены следующие обозначения:

– $\hat{F}_A(N)$ – худший случай на D_N – наибольшее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N :

$$\hat{F}_A(N) = \max_{D \in D_N} (F_A(D)); \quad (1)$$

– $\check{F}_A(N)$ – лучший случай на D_N – наименьшее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N :

$$\check{F}_A(N) = \min_{D \in D_N} (F_A(D)); \quad (2)$$

– $F_d(N)$ – средний случай на D_N – среднее количество операций, совершаемых алгоритмом A для решения конкретных проблем размерностью N :

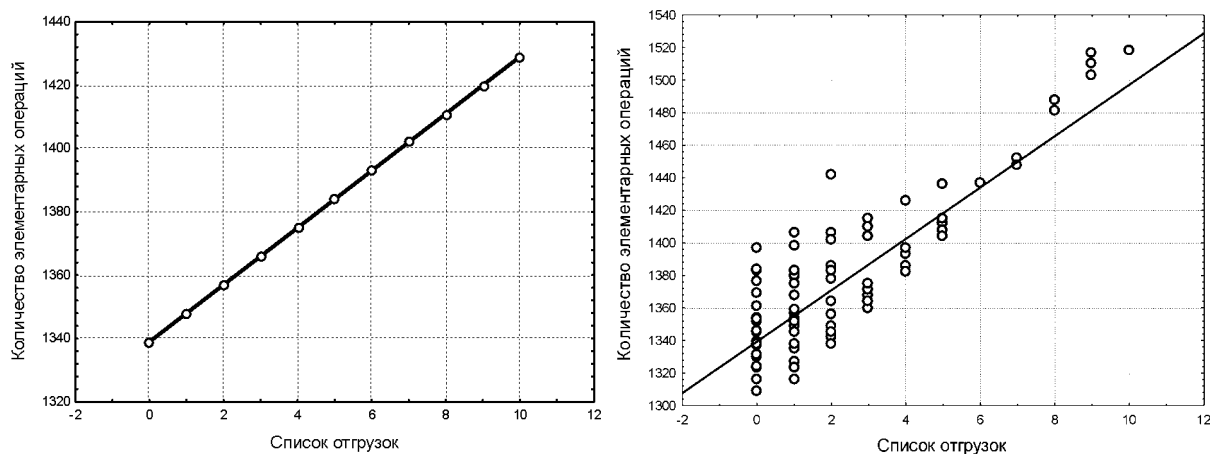
$$\overline{F}_A(N) = \frac{1}{|D_N|} \sum_{D \in D_N} F_A(D). \quad (3)$$

Определение трудоемкости процесса «Планирование производства» обеспечивается расстановкой по тексту программы модели счетчика элементарных операций *elcounter*. Для последующего статистического анализа создан отклик «Количество элементарных операций», параметрами которого являются *elcounter* и *time* (время вызова активности), а также «Список отгрузок» с параметром *shipments.size()*. В результате такого анализа программы модели функция теоретической трудоемкости алгоритма для лучшего случая, см. формулу (2), представлена в виде:

$$\hat{F}_A = 13M^2N^2 + 4M^2N + 75MN + 7M + 8CN + 42N + 9S + 42, \quad (4)$$

где $M \equiv MPSPlanningHorizon$ – количество интервалов планирования в основном плане-графике производства, $N \equiv numberOfResourceTypes$ – количество видов продукции, $C = productionCycle/MPSUnit$ (*productionCycle* – длительность производственного цикла, *MPSUnit* – интервал планирования основного плана-графика производства), $S \equiv shipments[]$ – список отгрузок.

По формуле (4) построен график теоретической трудоемкости алгоритма модели (рисунок 2а), показывающий рост трудоемкости алгоритма от параметра «Список отгрузок». Результаты экспериментальных исследований (рисунок 2б) подтверждают правильность программы процесса «Планирование производства». Отклонения между экспериментальными и теоретическими результатами оценки трудоемкости алгоритма не превышают 5 %.



а) теоретическая трудоемкость

б) экспериментальная трудоемкость

Рисунок 2 – Зависимость теоретической (а) и экспериментальной (б) трудоемкости алгоритма от параметра «Список отгрузок» при $M=1$, $N=6$, $C=2$

Формула для вычисления времени работы алгоритма имеет вид:

$$T_A = 728,9 \cdot 10^{-9} F_A,$$

где $728,9 \cdot 10^{-9}$ с – время выполнения элементарной операции на процессоре с тактовой частотой 500 МГц.

Заключение

Верификация имитационной модели методом оценки трудоемкости алгоритма модели позволяет собрать статистику о взаимодействии активностей модели. После статистической обработки полученных результатов программными средствами, например, используя пакет Statistica данные о взаимодействии активностей модели представляются в виде графических диаграмм. При этом появляются дополнительные возможности по тестированию модели благодаря визуализации результатов. Метод реализован при разработке динамической производственно-экономической модели завода органического синтеза [8]

Abstract. The paper presents the method of verification of a simulation model based on the study of theoretical and experimental function of the algorithms laboriousness. The realization of the method is shown on the example of the process “Production Planning”. The results of the investigation are also given.

Литература

1. Якимов, А. И. Имитационное моделирование в ERP-системах управления / А. И. Якимов, С. А. Альховик. – Мн.: Бел. наука, 2005. – 198 с.: ил.
2. Альховик, С. А. Имитационная модель промышленного предприятия для ERP-системы управления / С. А. Альховик, А. И. Якимов // Вестник Могилевского государственного технического университета. – 2004. – №2(7). – С. 11-16.
3. Balci, O. Expanding our Horizons in Verification, Validation, and Accreditation Research and Practice / O. Balci, R. E. Nance, J. D. Arthur, W. F. Ormsby [Электрон. ресурс] Proceedings of 2002 Winter Simulation Conference. – 2002. – pp. 653-663. – Режим доступа : <http://www.informs-sim.org/wsc02papers/084.pdf>.

4. Sadowski, D. A. Tips for Successful Practice of Simulation / D. A. Sadowski, M. R. Grabau [Электрон. ресурс] Proceedings of 2004 Winter Simulation Conference. – 2004. – pp. 61-66. – Режим доступа : <http://www.informs-sim.org/wsc04papers/008.pdf>.
5. Overstreet, C. M. Model Testing: Is it Only a Special Case of Software Testing? / C. M. Overstreet [Электрон. ресурс] Proceedings of 2002 Winter Simulation Conference. – 2002. – pp. 641-647. – Режим доступа : <http://www.informs-sim.org/wsc02papers/082.pdf>.
6. Brady, T. F. Simulation Data Mining: A New Form of Computer Simulation Output / T. F. Brady, E. Yellig [Электрон. ресурс] Proceedings of 2005 Winter Simulation Conference. – 2005. – pp. 285-289. – Режим доступа : <http://www.informs-sim.org/wsc05papers/030.pdf>.
7. Ульянов, М. В. Математическая логика и теория алгоритмов, часть 2: теория алгоритмов / М. В. Ульянов, М. В. Шептунов. – М.: МГАПИ, 2003. – 80 с.
8. Разработка динамической производственно-экономической модели завода органического синтеза ОАО «Могилевхимволокно» : отчет о НИР (заключ.) / Бел.-Рос. ун-т; рук. А. И. Якимов ; исполн. С. А. Альховик [и др.]. – Могилев, 2007. – 166 с. – Библиогр. : с. 148-150. – № ГР 20066024. – Инв. № ХД 0663.