

В.Ю. ОСИПОВ, В.В. КУЗЕННЫЙ
**ОПТИМИЗАЦИЯ ВИЗУАЛЬНОЙ СРЕДЫ
ПРОСТРАНСТВЕННОГО МОДЕЛИРОВАНИЯ**

Осипов В.Ю., Кузенный В.В. **Оптимизация визуальной среды пространственного моделирования.**

Аннотация. Рассматривается метод оптимизации визуальной среды пространственного моделирования с учетом накопленного опыта разработчика. Метод доведен до практической реализации. Его предлагается использовать в интеллектуальной подсистеме визуальной среды пространственного моделирования для повышения эффективности труда разработчика моделей.

Ключевые слова: оптимизация, среда моделирования, интеллектуальная подсистема.

Osipov V.Yu., Kuzenny V.V. **Optimization of visual environment for spatial modeling.**

Abstract. Technique for optimization of visual environment for spatial modeling is considered. It was implemented. The use of the technique in intelligent subsystem of visual environment for spatial modeling is proposed.

Keywords: optimization, modeling environment, intelligent subsystem.

1. Введение. Одной из актуальных научно-практических задач в области компьютерного моделирования пространственных процессов выступает поиск путей совершенствования прикладных программных систем. В качестве таковых могут выступать различные современные визуальные среды имитационного моделирования [1, 2].

Известны подходы по совершенствованию подобных систем. В основном они сводятся к периодическому обновлению версий среды и предоставлению пользователям возможности сохранять и использовать типовые шаблоны, создавать пользовательские блоки. При этом не в полной мере учитывается опыт разработки моделей конкретными специалистами.

Предлагается подход к оптимизации визуальной среды пространственного моделирования в реальном масштабе времени под индивидуальных разработчиков с учетом накопленного опыта.

2. Постановка задачи. Рассмотрим ситуацию, когда разработчик прикладных моделей накопил определенный опыт работы в некоторой визуальной среде моделирования пространственных процессов. В качестве такой среды моделирования может выступать, например, визуальная среда DroomsTab [3]. Общими свойствами таких систем являются: наличие визуального редактора моделей, позволяющего создавать схемы моделей, состоящие из блоков моделирования; библиотек стандартных блоков; репозитария моделей; возможность редакци-

рования и создания пользовательских блоков и библиотек и другие. При разработке модели эксперт выбирает блок из библиотеки, используя визуальную палитру, в которой каждому блоку поставлен в соответствие индивидуальный графический элемент, и переносит его на схему модели. Затем он соединяет его стрелками с другими блоками на схеме и в отдельной форме задает входные параметры. Пример схемы модели в среде DroolsTab приведен на рис. 1.

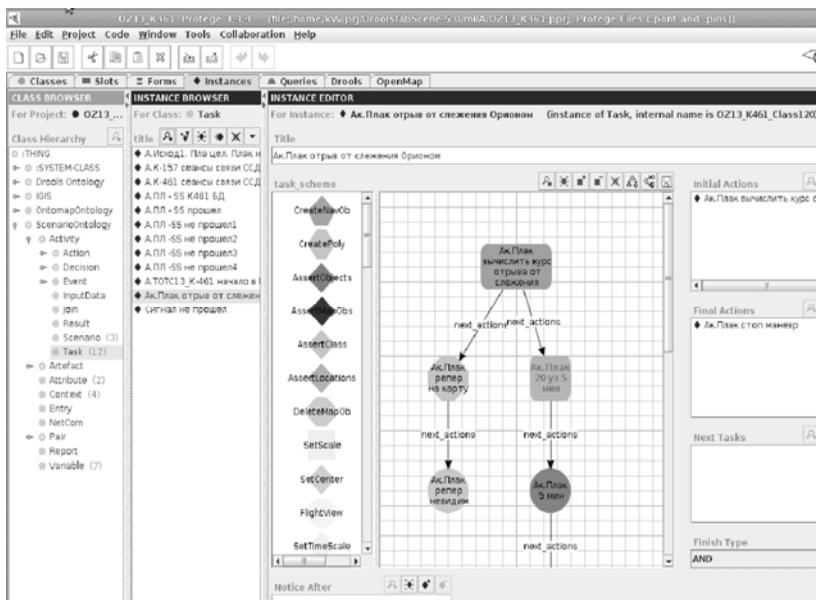


Рис. 1. Схема модели в среде DroolsTab

Результатом накопленного опыта выступает база уже разработанных в этой среде моделей, отражающих закономерности, свойственные моделированию процессов в заданной предметной области. Эти закономерности могут быть представлены в виде характеристик наиболее часто используемых шаблонов. Включение этих шаблонов в базовую библиотеку в качестве отдельных блоков, с одной стороны, позволяет сократить время на разработку новых моделей. С другой стороны, это включение увеличивает затраты времени на поиск необходимого блока и его добавление в модель. Однако не очевидно, какая библиотека блоков с таким подходом может обеспечить наибольшую

эффективность разработки новых моделей в заданной предметной области.

Для разрешения этой ситуации в состав моделирующей системы предлагается включить интеллектуальную подсистему, обеспечивающую анализ накопленного опыта, выработку и реализацию управляющих решений по совершенствованию используемой оператором среды моделирования. Обобщенная структура такой интеллектуальной моделирующей системы приведена на рис. 2.

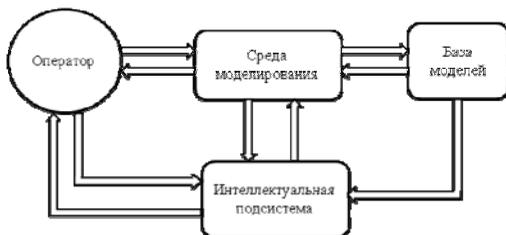


Рис. 2. Структура интеллектуальной системы моделирования

Согласно рис. 2, интеллектуальная подсистема периодически анализирует базу готовых моделей, контролирует текущее состояние самой среды моделирования и оценивает возможности по ее совершенствованию. В случае наличия таких возможностей, например, за счет расширения или сужения библиотеки блоков, она может выдавать рекомендации оператору по совершенствованию среды моделирования. Полагается, что в общем случае оператор может напрямую контролировать состояние интеллектуальной подсистемы и изменять ее параметры.

С учетом этих обстоятельств требуется разработать метод оптимизации визуальной среды моделирования, учитывающий накопленный опыт и позволяющий повысить эффективность разработки моделей.

2. Математическая формулировка. Пусть имеется стандартная или начальная библиотека базовых блоков $\bar{B} = \{b_1, b_2, \dots, b_n\}$, из которых были разработаны Q моделей в заданной предметной области. Они хранятся в базе данных. Каждой модели M_q ставится в соответствие пара $M_q = (\bar{s}_q, A_q)$, где $\bar{s}_q = \{s_{q1}, s_{q2}, \dots, s_{qk}, \dots, s_{qd_q}\}$, $s_{qk} \in \bar{B}$, $k = \overline{1..d_q}$ $q = \overline{1..Q}$ $A_q = \|a_{ij}\|_q$ — матрица смежности графа для q -й

модели размером $d_q \times d_q$. В этой матрице $a_{ijq} = 1$, если i -й блок связан с j -м блоком в q -й модели; в противном случае $a_{ijq} = 0$.

Каждому новому блоку b_r , за счет которого можно расширить \bar{B} и сформировать некоторую новую библиотеку \bar{B}_e , также ставится в соответствие пара $b_r = (\bar{s}_r, A_r)$, где $\bar{s}_r = \{s_{r1}, s_{r2}, \dots, s_{rk}, \dots, s_{rm_r}\}$, $s_{rk} \in \bar{B}$, $k = \overline{1..m_r}$, $r = \overline{1..z}$; A_r — матрица смежности графа блока, определяемая аналогично матрице A_q ; z — число вводимых в \bar{B} новых блоков.

Каждой новой библиотеке \bar{B}_e свойственно свое время $T_1(N_e)$ извлечения из нее необходимого блока. Чем шире эта библиотека, тем больше это время.

Всем имеющимся и новым блокам характерно среднее время $T_2(b_f)$ введения f -го блока в разрабатываемую модель, включающее в себя время задания его параметров. Чем больше размер блока, тем меньше это время.

Путем анализа базы имеющихся моделей для блоков b_f можно определить частоты $P(b_f)$ их использования при моделировании процессов в заданной предметной области. Введение нового блока в библиотеку допустимо, если частота $P(b_f)$ его использования не ниже заданного уровня P_0 .

Формирование новой библиотеки \bar{B}_e блоков, учитывающей накопленный опыт, должно осуществляться за время $T_e(\bar{B}_e)$, не превышающее допустимого значения T_0 .

С учетом этих обстоятельств, требуется найти целесообразную библиотеку блоков \bar{B}_o , при которой обеспечивается минимум временных затрат $W(\bar{B}_o)$ на разработку типового набора моделей в заданной предметной области:

$$W(\bar{B}_o) = \min_{e \in E} \sum_{f=1}^{N_e} P(b_{ef})(T_1(N_e) + T_2(b_{ef})) \quad (1)$$

при условиях:

$$P(b_{ef}) \geq P_0; \quad (2)$$

$$b_{ef} \in B_e; \quad (3)$$

$$T_e(B_e) \leq T_0; \quad (4)$$

$$f = \overline{1..N_e}; \quad e = \overline{1..E};$$

В (1)–(4) N_e — число блоков в библиотеке \overline{B}_e ; E — число всех возможных альтернативных библиотек.

Особенность этой задачи состоит в поиске наиболее часто используемых оператором шаблонов, частот $P(b_f)$ их применения, а также в формировании альтернативных библиотек и расчете параметров $T_1(N_e)$, $T_2(b_f)$.

Для решения оптимизационной задачи (1)–(4) предлагается использовать стратегию «жадного» выбора [4]. Ниже приводится алгоритм решения, реализующий эту стратегию.

3. Алгоритм решения. Алгоритм включает в себя следующие шаги.

1. Загрузить текущую базу моделей в интеллектуальную подсистему (см. рис. 2).

2. Определить частоту применения $P(b_f)$ для каждого блока b_f из начальной библиотеки $\overline{B} = \{b_1, b_2, \dots, b_n\}$.

3. Рассчитать $W(\overline{B}) = \sum_{f=1}^N P(b_f)(T_1(N) + T_2(b_f))$ для начальной библиотеки.

4. Положить $W_{\min} = W(\overline{B})$. Принять N равным числу блоков в начальной библиотеке.

5. Определить множество шаблонов G , являющихся потенциально новыми блоками, для которых выполняется условие (2).

6. Для всех шаблонов из G рассчитать выигрыш $C(b_r)$ — период сокращаемого времени при использовании данного шаблона как отдельного блока для разработки моделей.

7. Удалить из G шаблоны, для которых значение выигрыша $C(b_r) = 0$.

8. Если $G \neq \emptyset$ то упорядочить G по выигрышу $C(b_r)$, иначе имеет место конец решения задачи.

9. Выбрать $b_r \in G$ с максимальным значением выигрыша $C(b_r)$.

10. Удалить выбранный элемент b_f из множества шаблонов G и добавить его в библиотеку \bar{B} . Положить $N = N + 1$.

11. Пересчитать частоты $P(b_f)$ применения блоков библиотеки \bar{B} , входящих в новый блок b_r .

12. Удалить все $b_f \in \bar{B}$, для которых $P(b_f) = 0$.

13. Рассчитать $W(\bar{B}) = \sum_{f=1}^N P(b_f)(T_1(N) + T_2(b_f))$.

14. Если $W(\bar{B}) < W_{\min}$, то $W_{\min} = W(\bar{B})$, то переходим к шагу 8.

Принимая во внимание то, что схемы моделей, состоящие из блоков, соединенных связями, можно представить графами, для выполнения шага 5 (определения множества потенциальных новых блоков G и их частот $P(b_f)$) предлагается использовать известный алгоритм *gSpan (graph-based Substructure pattern mining)* поиска частых подграфов во множестве графов [5]. Алгоритм позволяет находить во множестве графов подграфы с частотами, не меньшими заданного минимального значения.

Для оценки времени $T_1(N)$ поиска блока в библиотеке предлагается использовать известную формулу [6]:

$$T_1(N) = \frac{N+1}{2} t_{\phi}, \quad (5)$$

где t_{ϕ} — время фиксации взгляда, которое в зависимости от условий восприятия составляет 0,025–0,65 с и более.

Для оценки времени $T_2(b_f)$ выполнения оператором добавления блока в модель и введения параметров применима известная математическая модель KLM оценки временных параметров интерфейсов [7]. Она позволяет рассчитать время выполнения какой-либо задачи пользователем, используя в качестве исходных данных спецификацию данной задачи и ее интерфейса.

С использованием модели KLM для каждого стандартного блока b_f подсчитывается среднее время по всей базе, требуемое для добавления блока в модель и задание параметров. Это время и составляет $T_2(b_f)$.

Расчет $T_2(b_r)$ новых блоков выполняется по формуле:

$$T_2(b_r) = \sum_f c(b_r)T_2(b_{rf}), \quad (6)$$

где сумма вычисляется по всем стандартным блокам b_{rf} , входящим в b_r ; $c(b_r)$ — выигрыш по времени от одноразового использования нового блока, вычисляемый по формуле:

$$T_2(b_r) = \sum_f c(b_r)T_2(b_{rf}), \quad (7)$$

$$c(b_r) = n_{edge}(b_r)(2R + 2B + 2M) + (n_{node}(b_r) - 1)(T_1(N + 1) + 2M + t_{title} + 2R + 2BB) \quad (8)$$

где n_{edge} — число стандартных блоков, входящих в b_r ; n_{node} — число внутренних связей (стрелок на схеме) между стандартными блоками, входящими в b_r ; t_{title} — среднее время ввода названия блока при добавлении его в модель; R — время, необходимое для указания с помощью манипулятора типа «мышь» на необходимый участок экрана; BB — время, необходимое для нажатия, а затем отпуская кнопки на манипуляторе типа «мышь» (одиночный «клик»); M — время, необходимое пользователю для поиска решения.

Выигрыш на шаге 6 рассчитывается по формуле:

$$C(b_r) = c(b_r)P(b_r). \quad (9)$$

Разработанный алгоритм реализован программно на языке Java. Для поиска частых подграфов во множестве графов использован пакет ParSeMiS (*Parallel and Sequential Graph Mining Suite*) [8], реализующий алгоритм gSpan. База моделей переводилась в формат GraphML [9].

3. Результаты и их анализ. С использованием разработанного метода исследовался опыт разработки сценариев действий сил на море в визуальной среде пространственного моделирования DroolsTab [3]. Сценарии состояли из задач, каждая из которых представляла собой определенную модель действий. Стандартная библиотека \bar{B} содержала 32 блока ($N = 32$), предназначенных для моделирования типовых действий. База моделей содержала $Q = 2052$ различных задач.

Параметры R , BB , входящие в (7), согласно оригинальной KLM, принимались равными 1,1 и 0,2 с соответственно [13].

В результате решения задачи (1)–(4) получены зависимости времени поиска блока в библиотеке $\sum_{f=1}^N P(b_f)T_1(N)$ (рис. 3а), времени до-

добавления блока в модель и задания параметров $\sum_{f=1}^N P(b_f)T_2(b_f)$ (рис. 3б)

и суммарного времени разработки исследуемого набора моделей $W(\bar{B})$ (рис. 4) от размера N библиотеки \bar{B}

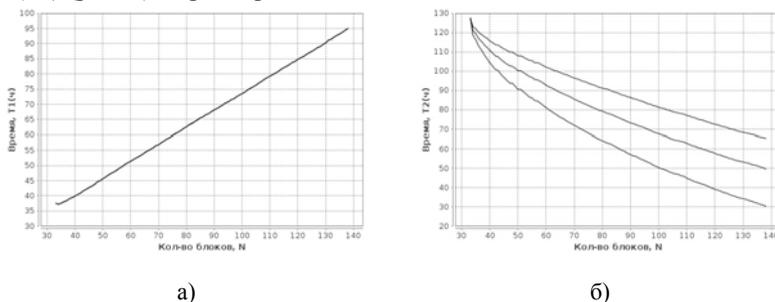


Рис. 3. Временные затраты на поиск блока в библиотеке (а) и добавление блока в модель (б)

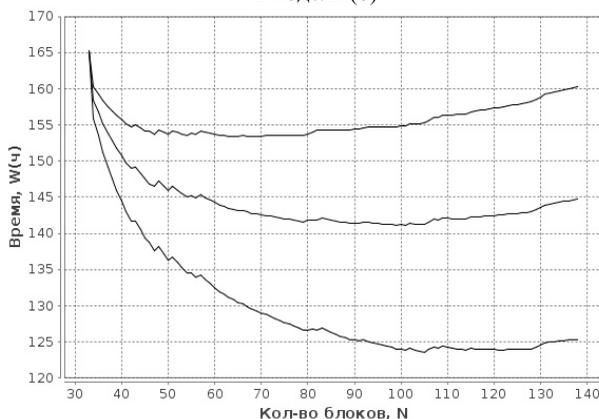


Рис. 4. Графики временных затрат на разработку моделей.

Кривые, обозначенные на рис. 3б и 4 цифрой 1, получены при времени $M = 1,2$ с, необходимом для поиска решения, кривые 2 — при $M = 6$ с. и кривые 3 — при $M = 12$ с. Время M , входящее в (7), в данном случае определяет уровень подготовленности разработчика.

Размер оптимальной библиотеки \bar{B}_0 для рассматриваемых случаев составил соответственно $N_0 = 65$, $N_0 = 101$ и $N_0 = 105$, а выигрыш

во времени — 5,4, 14,6 и 25,2% относительно использования базовой библиотеки.

На рис. 3б видно, что точка начала полученных зависимостей соответствует начальной библиотеке с числом блоков $N = 32$ и временем разработки заданного набора моделей $W(\bar{B}) = 165,28$ ч. Заметим, что с добавлением в \bar{B} новых блоков время на разработку моделей $W(\bar{B})$ сначала быстро сокращается (см. рис. 4), стремясь к минимуму. Затем, преодолев точку, соответствующую оптимальной библиотеке \bar{B}_0 , данный период начинает расти практически по линейному закону. Это обусловлено существенным преобладанием значений $T_1(N)$ над $T_2(\bar{b})$, в зависимости от которых рассчитывается основной показатель.

Из анализа полученных результатов следует, что учет опыта позволяет существенно снизить временные затраты на разработку моделей.

4. Выводы. Предложен подход к оптимизации среды визуального моделирования по минимуму временных затрат на разработку моделей с учетом накопленного опыта. Такой подход позволит, как показали полученные оценки, сократить время на разработку моделей в заданной предметной области с применением полученной оптимальной библиотеки от 5,4 до 25,2% относительно использования базовой. Рекомендуется в состав визуальных сред пространственного моделирования вводить интеллектуальную подсистему для учета накопленного опыта и выдачи рекомендаций по совершенствованию используемой среды.

Литература

1. Simulink — Simulation and Model-Based Design: [электронный ресурс: URL: <http://www.mathworks.com/products/simulink/> (дата обращения 15.11.2009)].
2. Асафьев Г.К. Современные системы имитационного моделирования // Научный исследовательский центр СПбГУ ИТМО: [электронный ресурс: http://sdk.ifmo.ru/publications/articles/MajorRead2009/MREAD2009_AsafevGK.pdf (дата обращения 02.12.2010)].
3. DroolsTab — визуальная среда для пространственного моделирования на основе Protégé и Drools // Научно-исследовательская лаборатория объектно-ориентированных геоинформационных систем: [электронный ресурс: URL: http://oogis.ru/component/option,com_remository/Itemid,34/func,fileinfo/id,15/lang,ru/ (дата обращения 10.10.2010)].
4. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Глава 16. Жадные алгоритмы // Алгоритмы: построение и анализ = Introduction to Algorithms. / Под ред. И. В. Крайскова. М.: Вильямс, 2005. 1296 с.

5. *Yan X., Han J.* gSpan: Graph-Based Substructure Pattern Mining // Proc. 2002 of Int. Conf. on Data Mining (ICDM'02).
6. *Душков Б.А., Ломов Б.Ф., Смирнов Б.А.* Хрестоматия по инженерной психологии. М.: Высшая школа, 1991. 287 с.
7. *Kieras D.E.* Using the Keystroke-Level Model to Estimate Execution Times // University of Pittsburgh: [электронный ресурс:
URL: www.pitt.edu/~cmlewis/KSM.pdf (дата обращения 10.10.2010).
8. ParSeMiS — the Parallel and Sequential Mining Suite: [электронный ресурс:
URL: <http://www2.cs.fau.de/EN/research/ParSeMiS/index.html> (дата обращения 15.05.2010).
9. The GraphML File Format: [электронный ресурс:
URL: <http://graphml.graphdrawing.org/> (дата обращения 17.06.2010).

Осипов Василий Юрьевич — д-р техн. наук, профессор; в. н. с. Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: информационное противоборство, моделирование, искусственный интеллект. Число научных публикаций — 102. osipov_vasily@mail.ru; СПИИРАН, Галерный проезд В.О., д. 5, Санкт-Петербург, 199178, РФ; п.т. 8 (812) 355-96-82.

Osipov Vasily Yurevich — Dr.Sci.Tech., the professor; the leading scientific employee of the St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: information antagonism, modelling, artificial intelligence. The number of publications — 102. osipov_vasily@mail.ru; SPIIRAS, 5, Galerniy proesd V.O., St. Petersburg, 199178, Russia; office phone 8 (812) 355-96-82.

Кузенный Виктор Викторович — аспирант Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: геоинформационные системы, моделирование, искусственный интеллект. Число научных публикаций — 2. vvkuz@oogis.ru; СПИИРАН, Галерный проезд В.О., д. 5, Санкт-Петербург, 199178, РФ; п.т. 8 (812) 355-96-82.

Kuzenny Victor Victorovich — postgraduate student of the St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: geoinformation systems, modelling, artificial intelligence. The number of publications — 2. vvkuz@oogis.ru; SPIIRAS, 5, Galerniy proesd V.O., St. Petersburg, 199178, Russia; office phone 8 (812) 355-96-82.

РЕФЕРАТ

Осинов В.Ю. Кузенный В.В. **Оптимизация визуальной среды пространственного моделирования.**

Одной из актуальных научно-практических задач в области компьютерного моделирования пространственных процессов выступает поиск путей совершенствования прикладных программных систем. В качестве таких систем могут выступать различные визуальные среды имитационного моделирования.

Известны подходы по совершенствованию подобных систем. В основном они сводятся к периодическому обновлению версий среды и предоставлению пользователям возможности сохранять и использовать типовые шаблоны, создавать пользовательские блоки. При этом не в полной мере учитывается опыт разработки моделей конкретными специалистами.

Предлагается метод оптимизации визуальной среды пространственного моделирования с учетом накопленного опыта индивидуальных разработчиков. Суть метода сводится к поиску оптимальной библиотеки блоков среды моделирования, учитывающий текущий опыт. Поиск такой библиотеки осуществляется, исходя из минимизации времени на разработку заданного набора моделей при ограничениях на частоту использования блоков, время их поиска и размер. Для решения этой оптимизационной задачи предлагается использовать стратегию жадного выбора. Разработан алгоритм решения этой задачи, доведенный до программной реализации. Проанализировано 2052 сценария, разработанных в визуальной среде пространственного моделирования DroomsTab. С учетом этого опыта получены результаты оптимизации этой среды моделирования. Показано, что своевременная оптимизация среды моделирования с учетом накопленного опыта позволяет снизить временные затраты на разработку моделей от 5,4% до 25,2% относительно исходного состояния. Рекомендуется этот метод реализовывать в составе интеллектуальных подсистем современных визуальных сред пространственного моделирования.

SUMMARY

Osipov V.Yu., Kuzenny V.V. **Optimization of visual environment for spatial modeling.**

Search of ways of enhancement of computer application systems is one of current scientific and practical problems in computer simulation of spatial processes. A variety of visual simulation environments can serve as such systems.

A variety of approaches are known to improve such systems. Basically they boil down to periodic upgrades of environments and enable users to save and use standard templates, to create custom blocks. This does not fully take into account experience of specific skills in developing models.

A new method for optimizing the visual environment for spatial modeling based on experience of individual developers is proposed. The essence of the method reduces to finding the best libraries of blocks of the simulation environment, taking into account the current experience. Search this library is carried out based on the minimization of time to develop a given set of models with constraints on the frequency of use of blocks, search time, and size. To resolve this optimization problem is proposed to use the greedy strategy of choice. An algorithm for solving this problem brought to the software implementation. The 2052 scenario developed in a visual environment for spatial modeling DroolsTab was analyzed. The optimization results were obtained based on this experience of development in that environment. It is shown that timely optimization of simulation environment based on experience helps reduce the time needed to develop models from 5,4% to 25,2% with respect to the initial state. It is recommended to implement this method in the intelligent subsystems of modern visual environment for spatial modeling.