

ЯЗЫК SLX – НОВЫЙ ЭТАП ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

А. Г. Варжапетян (Санкт-Петербург), Д. Хенриксен (США)

Введение

Язык SLX (Simulation Language with Extensibility) относится к новому поколению ПП Wolverine Corp. и основывается на широких возможностях ЯИМ GPSS/H. Традиционный подход версий GPSS, основанный на построении модели из блоков обладает целым рядом преимуществ: позволяет достаточно просто и быстро создавать МФ, применять ЯИМ для широкого класса разнообразных систем, обеспечивать единую основу для математического представления. Вместе с тем, используя традиционный подход, приходится сталкиваться с рядом проблем.

1. Набор операторов ЯИМ имеет тенденцию к разрастанию, по мере того как усложняются моделируемые системы и возрастает их номенклатура. Простое добавление новых операторов в известной мере начинает разрушать концептуальную целостность начальной идеи ЯИМ.

2. Не вполне очевидно, как выбирать наиболее оптимальную комбинацию операторов, способную отобразить новые возникающие ситуации, отображение которых не вполне отвечает начальным возможностям операторов.

3. Для многих практических пользователей уровень абстракции выше уровня его понимания, так, например, производственный мастер с большой для себя пользой мог бы применять ИМ, но не знает, как описать модель.

Учитывая эти и ряд других обстоятельств Wolverine Corp. на протяжении ряда лет вела разработку языка развивающего возможности традиционного ЯИМ, построенного на базе операторов. При разработке языка SLX решались три глобальных проблемы: введение послойного построения (в оригинале layer – слой, пласт), создания механизма организации связей и управления между слоями и создания внешнего интерфейса, обеспечивающего доступ к слоям.

Основная часть

Для решения этих проблем пришлось решать следующие технические задачи:

- построить такую иерархию хорошо организованных слоёв, которая позволяла бы пользователю SLX большую часть времени уделять решению практических задач на верхних слоях иерархии, но при необходимости осуществлять простой переход на нижние слои.

- создать механизм перехода вверх и вниз между слоями.
 - создать интерфейс, ориентированный на ОС Windows, который:
 - 1) был бы скрыт на высших слоях,
 - 2) немедленно был готов к использованию при запросе,
 - 3) мог осуществлять эффективную работу на нижнем слое.
- На рис. 1 представлена пирамида SLX

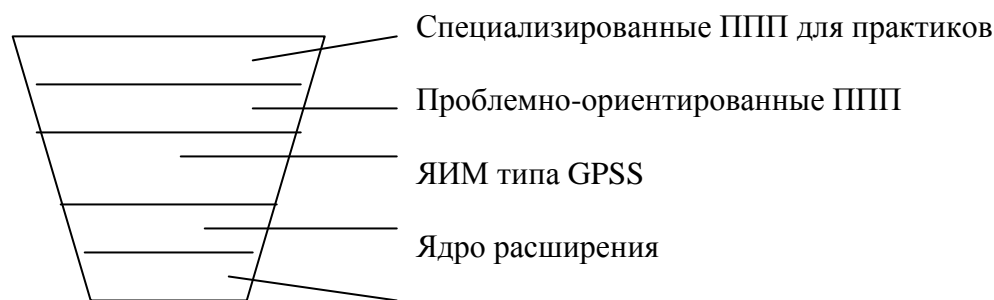


Рис. 1. Пирамида SLX

Язык обеспечивает большие возможности для моделирования современных систем, описываемых СИ-подобными языками. Язык представляет собой многоуровневую (послойную) структуру с ядром SLX в основании пирамиды, далее в середине традиционный язык имитационного моделирования (типа GPSS/H) и на вершине специализированные и прикладные диалекты языка.

Наиболее важной характеристикой языка является его послойная (многоуровневая) архитектура. Эффективность применения такой структуры зависит от ряда обстоятельств:

А. Уровни должны быть хорошо продуманы и чётко выражены. В каждом слое содержится минимально необходимое число функций. Развитию языка способствовала многолетняя успешная практика работы с GPSS/H, который и является основой нового языка. В некоторых случаях коды начального файла GPSS/H, могут быть перенесены для использования в SLX. В других случаях можно сократить, упростить или модифицировать алгоритмы GPSS/H или даже устранить дефекты программы.

В. Уровни не должны находиться далеко друг от друга. Во многих языках программирования используется многоуровневая структура, но обычно между уровнями существует большой разрыв. Например, язык может обеспечивать в качестве начальной концепции объектно-ориентированное построение, но не учитывать его в процедурно-ориентированном описании, как это допускают языки типа ФОРТРАНА или СИ. Проблема состоит в том, что в этом случае рассматриваются лишь два уровня далеко отстоящие друг от друга. На одном рассматривается большое число деталей реализации СИ, которые могут быть использованы в ЯИМ и расширить возможности СИ. С другой стороны, система контроля ошибок ЯИМ не может быть использована в СИ. Ядро SLX, написанное на СИ – подобном языке, позволяет пользователю не отбрасывать детали более низкого уровня, делая язык более мощным. Следовательно, SLX включает возможность комплексной проверки возможного возникновения ошибок, таких как ссылки в конце массива данных или использование неверных величин. Переход с уровня на уровень производится весьма просто.

С. Механизм перехода с уровня на уровень весьма эффективен и базируется на абстрактном представлении. Более высокие уровни обеспечивают более абстрактное описание, чем нижние. Детали описания нижних уровней скрыты в описании более высоких уровней. Язык обеспечивает как представление данных, так и абстрактных процедур. Подобно СИ, SLX обеспечивает возможность определения новых типов данных и агрегировать их в создаваемые объекты. Механизм процедурной абстракции весьма эффективен, включает макросы и возможность введения новых определений, позволяя создавать петли, дополнительные аргументы, расширение логики и т. п.

Д. Язык SLX позволяет осуществлять связь с другими языками и программами. Если у пользователя возникает необходимость обратиться к коллекции функций СИ/СИ++ непосредственно из SLX, то необходимо поместить эти функции в динамическую библиотеку (DLL) и обеспечить аргументами, отвечающими на вызов. SLX автоматически генерирует хедер-файлы СИ/СИ++(.h), которые определяют объекты SLX в синтаксисе СИ.

Особенности языка SLX

В компиляторах традиционных языков программирования *модули* программы преобразуются в объектный код, который непосредственно исполняется компьютером или преобразуется интерпретатором.

В SLX логика работы компилятора расширена. В процессе компиляции расширение осуществляется за счёт внутренних возможностей компилятора. На рис. 2 представлен пример расширения модуля С, за счёт расширения модуля В, причём этот процесс может повторяться неоднократно.

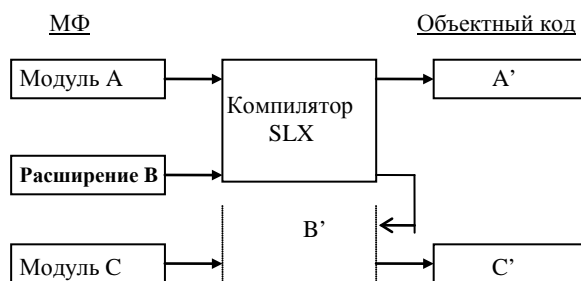


Рис. 2. Архитектура компилятора SLX

Высказав эти предварительные соображения, рассмотрим некоторые особенности построения языка.

А. Особенности построения

А1. Уровни (слои) пирамиды SLX, Рис.1 достаточно хорошо представляет разделение слоёв SLX. Ядро SLX достаточно мало, но содержит все необходимые элементы ИМ. В результате тщательного отбора разработчики выделили из GPSS/H наиболее важные, корневые операторы. Ядро имеет следующие основные характеристики:

- число потребных операторов весьма мало, благодаря их малому количеству базовые операторы могут комбинироваться таким образом, чтобы описать наибольшее число возникающих ситуаций;

- ядро имеет непосредственный доступ. В традиционном построении GPSS операторы не допускают прямого доступа, до тех пор, пока они не будут переведены в надлежащее состояние (логические переключатели, очереди и т. п.). Такая функция как `wait until` сложна для реализации, поэтому не удивительно, что она очень редко используется в других ЯИМ. Парадоксально, что команды, трудные для реализации, наиболее легко понимаются пользователями. Эти сложности исключены с появлением в SLX директивы `wait until`;

- доступ к основным элементам ядра гарантирует точность и адекватность модельного представления. В пирамиде SLX выполняется правило: «чем ниже вы опускаетесь, тем более точная модель получается». На уровне ядра точность виртуально не ограничена;

– ядро обеспечивает вычислительные возможности при работе с СИ-подобными программами. Большинство ЯИМ меньше внимания уделяют вычислительным возможностям, а больше обращают внимание на управление параллелизмом процесса ИМ, чем на вычислительные возможности;

– все элементы ядра, включая взятые из ядра СИ, активно контролируются, возникновение ошибок типа неполного запоминания массивов, неверного синтаксиса диагностируются в процессе исполнения и приводят к его прерыванию.

Обратимся ещё раз к рис. 1, каждый более высокий слой шире нижнего. Это означает что более высокие слои включают в себя все особенности нижних слоёв. И каждая комбинация приспособлена для выполнения частных требований. Сравним, например, использование ОБ SEIZE в GPSS/H, который моделирует поведение одноканального сервера и SEIZE в SLX, который является подпрограммой 9-ти линейного исполняемого кода. Наиболее важные функции этого кода исполнение команды *wait until*, реализующей ожидание освобождения ОБ и сбора статистики между слоями GPSS/H и ядра SLX. Размер кода продиктован соображениями уменьшения величины и возможности контроля в процессе исполнения.

А2. Механизм связей между слоями. Создание слоёв, это только решение первой проблемы. Вторая, не менее важная – это создание механизма связей, который позволял бы расширять существующие слои или создавать новые поверх старых. Более высокие слои обеспечивают больший уровень абстракции, чем нижние, поэтому детали нижних слоёв скрыты при переходе на верхний уровень. SLX обеспечивает представление как данных, так и процедур. Подобно СИ, SLX обеспечивает определение новых типов данных и строит *объекты*, которые включают в себя эти данные. В SLX используются объекты двух видов. Первый вид – пассивные объекты, используемые для моделирования единиц, подобных геометрическим построениям в Proof, т. е. когда поведение объекта не меняется, или, например, операторы Facilites, Queues, Storages ЯИМ GPSS/H, неизменяемые ресурсы типа автостоянки и т. п. Второй вид – активные объекты, поведение которых меняется в процессе исследования.

А3. Возможности интерфейса SLX. Третьей проблемой было создание внешнего интерфейса, позволяющего проводить отладку и контроль процесса моделирования. Возможности ОС Windows позволяют наблюдать в процессе исполнения за данными модели, кодами МФ, списком событий и т. д.

А4. Запись команд и директив SLX. Одной из наиболее применяемых форм расширения возможностей SLX является использование новых команд и директив, являющихся по сути дела макросами (в понимании традиционных языков). Отличие состоит в том, что команды чаще работают на уровне утверждений, чем на уровне определений.

Б. Особенности ядра SLX

Б1. Определение пака (pack). Pack (эльф, озорной) является программным элементом SLX, определяющим время задержки или ожидания освобождения сервера. Манипуляция с паками позволяет осуществлять в процессе ИМ квазипараллелизм и является основным механизмом моделирования появления событий. В SLX функциональность транзактов скрыта на нижнем уровне и в ядре нет транзактов, как таковых. Роль транзактов, в данных определяемых пользователем, играет класс объектов (*object class*). Носителем задаваемых данных в SLX является пак (*pack*). Каждый объект SLX является членом какого то класса и имеет копию в данных объекта.

Б2. Внешний и внутренний параллелизм. В SLX параллелизм реализуется двумя путями: осуществление связей между объектами (внешний параллелизм) и многократное обеспечение действия самого объекта (внутренний параллелизм). Внешний параллелизм, при котором поровну 1:1 осуществляются связи между объектами и паками,

что функционально эквивалентно потоку транзактов. Внутренний параллелизм достигается определением более одного пака на активный объект.

В. Особенности расширения

SLX представляет собой платформу, на базе которой могут быть построены другие средства имитации. Механизм расширения позволяет придать известным средствам новые особенности. Особенно очевидно это проявляется в поисковых системах. Обычно, в большинстве языков программирования, для укрупнения операций используют макросы, структура которых ограничена или отличается по синтаксису от языка к языку. В этом смысле SLX сам является макросом, содержащим небольшое число основных определений и позволяющим чтение файла даже в процессе выполнения предписания.

Выводы

Язык SLX никоим образом не отменяет и не заменяет ЯИМ GPSS/H, скорее наоборот, за счёт реализации всех его преимуществ исследователи получают мощный, эффективный, интегрированный ПП, который естественным образом сочетает преимущества универсальных СИ-подобных языков с эффективным и быстродействующим ЯИМ GPSS/H. Рассмотрим вкратце преимущества языка.

- Язык SLX представляет собой многоуровневую (многослойную- layer) структуру, включающую в себя на нижнем уровне СИ- подобные конструкции (функции, подпрограммы), на среднем уровне ЯИМ типа GPSS и на верхних уровнях проблемно – ориентированные или специализированные пакеты ПП. Таким образом, возможность расширения решается достаточно просто.

- Слои SLX, в отличие от других языков, хорошо организованы и позволяют с удивительной легкостью осуществлять передвижение от нижних слоёв к верхним и наоборот. Функции нижних слоёв скрыты на верхних, но к ним всегда можно обратиться.

- Уровень абстрагирования возрастает снизу вверх, и пользователям специализированных пакетов верхнего слоя можно не разбираться с деталями построения программ нижних слоёв.

- Язык SLX обладает эффективными средствами диагностики и отладки.

- Введенные в языке директивы легко позволяют решать проблемы возникающих ожиданий, создания внутреннего и внешнего параллелизма, органично соединяя возможности макросов универсальных языков с простотой управления ими внутри языка SLX.

- Язык SLX хорошо сочетается с универсальными языками и может пользоваться файлами динамической библиотеки ОС Windows.

- Необходимо отметить, что язык достаточно просто интегрируется с ПП Proof Animation, обеспечивая качественную анимацию.

Литература

1. **Варжапетян А.Г.** Имитационное моделирование на GPSS/H М. Вузовская книга 2004
2. **Henriksen J.** An Introduction to SLX Proceedings of the 1997 Winter Simulation Conference IEEE 1997 p. 593–599