**6.1. Компонент построения сценария исполнения**

Прототип программного компонента построения сценария исполнения предназначен для обеспечения лексического, синтаксического и семантического анализа скрипта на прототипе языка EasyFlow (специализированном языке представления цепочек заданий (Workflow, далее WF), разрабатываемом в рамках проекта) с целью получения его внутреннего представления в виде объектного дерева.

Основное функциональное назначение компонента состоит в обеспечении возможности работы других компонентов (например, компонента управления исполнением и компонента диалогового интерфейса) с представлением абстрактного WF, заданного в форме скрипта на прототипе языка EasyFlow.

В функциональность компонента входят следующие возможности:

* получение скрипта EasyFlow в виде файла, потока или строки;
* лексический анализ полученного скрипта и обнаружение ошибок на уровне лексем, а также представление скрипта в виде потока лексем;
* синтаксический анализ на основе полученного потока лексем, состоящий в построении внутреннего представления, и обнаружение синтаксических ошибок с сообщением развернутой информации о них;
* семантический анализ скрипта с целью построения связей между различными блоками WF, а также с целью выявления логических ошибок;
* сбор информации о соответствии отдельных элементов внутреннего представления WF с его текстовым представлением (номера строки и столбца в файле и участки текста, соответствующие тем или иным элементам внутреннего представления);
* предоставление в результате разбора внутреннего представления WF в виде объектного дерева, пригодного для дальнейшей обработки другими компонентами.

Таким образом, компонент разбора скрипта EasyFlow представляет собой основу для реализации возможности задания пользователем абстрактного WF в виде декларативно-императивного.

Общая функционально-логическая схема компонента представлена на рисунке 6.1.1.

*Компонент построения сценария исполнения*

Семантический

анализатор

Лексический анализатор

Синтаксический

анализатор

Классовая структура

внутреннего

представления

Рисунок 6.1.1 - Структурная схема прототипа компонента построения сценария исполнения

Как видно из рисунка, компонент построения сценария исполнения состоит из трех анализаторов кода (лексического, синтаксического и семантического). Лексический анализатор предназначен для разбиения исходного текста скрипта EasyFlow на набор лексем. Синтаксический анализатор на основе этого набора анализирует язык на основе формального описания его грамматики, а семантический — производит действия, связанные с определением сути описанных в скрипте действий. Компонент интерпретации WF на основе формального представления WF, полученного от блока анализа кода, производит запуск WF с помощью подсистемы исполнения.

Взаимосвязь функциональной схемы работы компонента разбора скрипта EasyFlow с другими компонентами системы интеллектуальной поддержки представлена на рисунке 6.1.2. На нем видно, что обработка скрипта представляет собой процесс последовательных трансформаций и попутной обработки с построением внутреннего представления.

*Подсистема разбора*

Лексический

анализатор

Семантический

анализатор

Синтаксический

анализатор

*Компонент*

*управления*

*динамической*

*Базой знаний*

*Компонент*

*управления исполнением*

Знания о пакетах

,

параметрах

,

входных

и выходных данных

*Компонент*

*диалогового*

*интерфейса*

Графический вид

,

подсветка кода

,

контекстные подсказки

Внутреннее

представление

AWF

Информация об

ошибках

Скрипт

EasyFlow

Набор лексем

Информация об ошибках

Рисунок 6.1.2 - Функциональная схема работы компонента построения сценариев исполнения

Скрипт поступает в компонент построения сценариев исполнения от компонента диалогового интерфейса в текстовом виде для первичного лексического анализа. На этом этапе на основе формального определения грамматики языка EasyFlow происходит разбиение исходного текста на набор лексем, соответствующих основным понятиям языка (идентификатор, оператор, строка, число, комментарий и пр.) посредством лексического анализатора. При этом удаляются комментарии, пробелы, символы табуляции и возврата каретки. На выходе получается поток лексем с информацией об их типе и положении в тексте. На этом этапе происходит также первичная обработка ошибок, связанная с нахождением в тексте неприемлемых символов и лексем, не соответствующих формальной грамматике языка.

Полученный набор лексем является исходными данными для работы следующего блока − синтаксического анализатора, который на основе формальной грамматики языка выделяет из текста отдельные конструкции и строит внутреннее представление скрипта в объектном виде. На этом этапе пользовательский интерфейс может получать информацию о графическом представлении WF, данные о возникающих ошибках и служебную информацию для подсветки синтаксиса в текстовом редакторе и осуществления контекстных подсказок).

После синтаксического анализа происходит вызов блока семантического анализа, который на основе знаний, получаемых от компонента управления динамической базой знаний, определяет состав вызываемых пакетов, их формат, набор параметров, а также проверяет возможность исполнения переданного WF и его непротиворечивость (например, отсутствие циклических зависимостей). На выходе этого этапа, в случае отсутствия критических ошибок, получается внутреннее представление WF, пригодное для передачи в компонент запуска.

Пример разбора текста скрипта во внутреннее представление WF показан на рисунке 6.1.3 (в левом сером прямоугольнике стрелками показаны зависимости данных от запусков).



Рисунок 6.1.3 - Пример разбора скрипта EasyFlow

Из рисунка видно, что семантический анализатор учитывает зависимости по данным и строит соответствующее скрипту дерево исполнения. При этом учитывается возможность параллельного исполнения нескольких блоков в случае, если они не зависят друг от друга и находятся на одном уровне (т.е. им в одно и то же время становятся доступны все необходимые входные данные).

Для организации внутреннего представления WF используется набор классов, каждый из которых соответствует конкретному элементу языка (идентификатор, описание шага и т.п.). Структура этих классов представлена на рисунке 6.1.4.

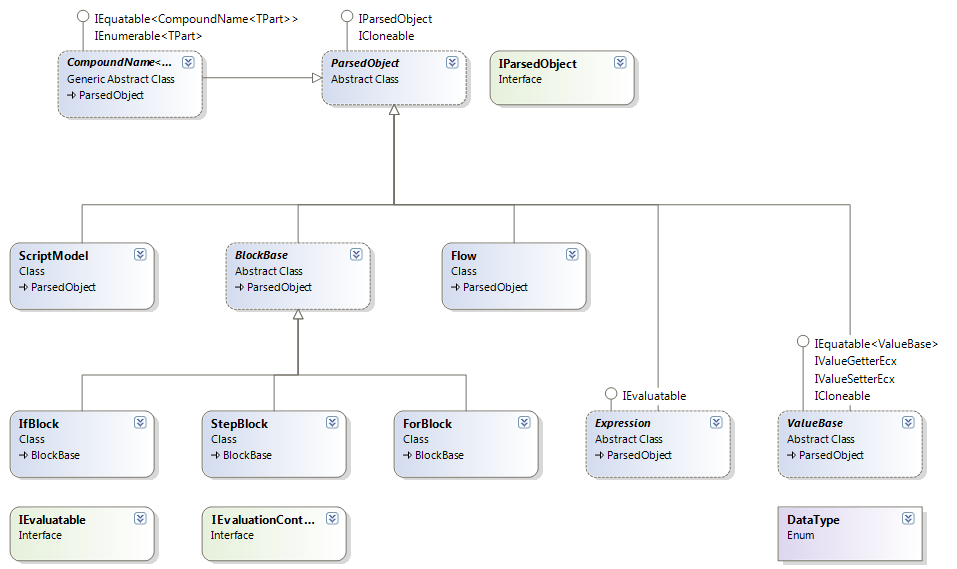


Рисунок 6.1.4. Частичная структура классов внутреннего представления WF

Входными данными для компонента является скрипт EasyFlow, который может быть передан компоненту тремя способами:

* как текстовый файл, содержащий скрипт;
* как строка (String), хранящая содержимое скрипта;
* как поток .NET (Stream), ссылающийся на любой поставщик данных (файл, строка, участок памяти, сетевое соединение и др.).

Выходными данными для компонента является объект класса ScriptModel, содержащий следующую информацию:

* граф внутреннего представления WF в виде объекта класса Flow;
* список требуемых файлов для запуска WF, описанных в секции require.

Прототип компонент представляет собой программную библиотеку, предназначенную для исполнения в среде Microsoft .NET 4.0, разработанную на языке C#. Для обеспечения лексического и синтаксического разбора в нем используется библиотека генерации синтаксиса ANTLR.

В качестве примера использования компонента построения сценариев рассмотрим прикладной сервис, предназначенный для моделирования динамики морских объектов под воздействием внешних волновых возмущений нерегулярной природы в заданном диапазоне характеристик эксперимента (номенклатуры геометрических моделей, параметров волнения, скорости и курсового угла). Результатом работы прикладного сервиса является набор операционных диаграмм (параметры колебаний объекта в зависимости от условий проведения эксперимента), используемый для обоснования выбора оптимальных характеристик объекта в процессе исследовательского проектирования.

Прикладной сервис включает в себя следующие составляющие:

* *композитное приложение*, описывающее процесс формирования и использования соответствующих вычислительных ресурсов и пакетов под управлением платформы;
* *прикладной пакет* ShipX-DS моделирования динамики морских объектов в экстремальных условиях эксплуатации, используемый в составе композитного приложения;
* *скрипт описания пакета* ShipX-DS*,* дополнительное метаописание и привязку пакетов к платформе;
* *контрольный пример:* входные данные и настроечные параметры, необходимые для обеспечения работы пакета ShipX-DS и сервиса в целом под управлением платформы.

На рисунке 6.1.5 приведен текст композитного приложения на языке EasyFlow в окне редактирования прототипа системы интеллектуальной поддержки, соответствующий описанной структуре сервиса.

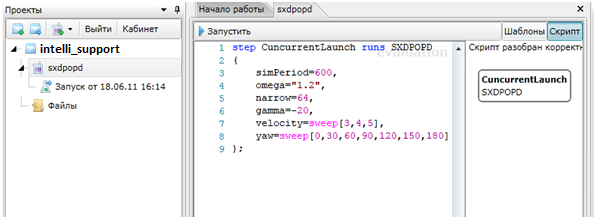


Рисунок 6.1.5 - Запуск композитного приложения на выполнение в среде прототипе среды интеллектуальной поддержки

Входные и выходные данные композитного приложения определяются следующим списком параметров запуска пакета ShipXDS (см. табл. 6.1.1).

**Таблица 6.1.1 − Список параметров запуска прикладного сервиса моделирования динамики морского объекта**

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| -ship | Модель судна. Возможные варианты: cutter, tugboat |
| -sim\_period | Продолжительность моделирования. Задается в секундах |
| -omega | Частота пика спектра морского волнения |
| -narrow | Параметр формы углового распределения морского волнения |
| -gamma | Параметр узости спектра морского волнения |
| -velocity | Удерживаемая скорость судна в м/с |
| -yaw | Удерживаемый курс судна относительно направления распространения волн в градусах в диапазоне [0;180°] |

Результатом работы композитного приложения, описанного на языке EasyFlow, является набор текстовых файлов, содержащих параметры движения судна с шагом по времени 0.25 с. В первую строку данных файлов записываются пять чисел, передаваемых в качестве входных данных: частота пика спектра, коэффициенты углового распределения и узости спектра морского волнения, удерживаемые скорость и курс судна. Каждая последующая строка содержит параметры движения судна на каждом шаге по времени: время в секундах, прошедшее с начала эксперимента, курс, угол продольного крена, угол бортового крена, три координаты центра масс судна, высота волны в точке нахождения центра масс судна, модуль линейной скорости в м/с и угловая скорость судна в рад/с. Данный набор файлов может быть использован для построения оперативной диаграммы судна (см. раздел 7.1).