ЗАО «АЙТИ»

УТВЕРЖДАЮ

Генеральный директор
ЗАО «АйТи».

_____ Бакиев О.Р.

"__" _____ 2011 г.

Создание высокотехнологичного производства комплексных решений в области предметно-ориентированных облачных вычислений для нужд науки, промышленности, бизнеса и социальной сферы

ПРОГРАММНЫЙ КОМПОНЕНТ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЯ ПРИ РАЗРАБОТКЕ СКРИПТОВ НА EASYFLOW CLAVIRE/UI/EASYFLOWEDITORSERVICES

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 62

2011

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

ЗАО «АЙТИ»

УТВЕРЖДЕН
RU.СНАБ.80066-06 13 62-ЛУ

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРОГРАММНЫЙ КОМПОНЕНТ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЯ ПРИ РАЗРАБОТКЕ СКРИПТОВ НА EASYFLOW CLAVIRE/UI/EASYFLOWEDITORSERVICES

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 62

ЛИСТОВ 15

2011

Инв.№ подл.	Подп. и дата	Взам.инв.№	Инв.№ дубл.	Подп. и дата

АННОТАЦИЯ

Документ содержит описание программного компонента CLAVIRE/UI/EasyFlowEditorServices RU.СНАБ.80066-06 01 62 поддержки пользователя при разработке скриптов на EasyFlow, реализующего функции проверки синтаксических и семантических ошибок, а также функции контекстно-зависимых подсказок возможных вариантов ввода при написании текстов в редакторе скриптов на языке EasyFlow в процессе описания композитных приложений для МИТП CLAVIRE. Программный компонент поддержки пользователя при разработке скриптов на EasyFlow разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

Компонент поддержки пользователя при разработке скриптов на EasyFlow CLAVIRE/UI/EasyFlowEditorServices RU.СНАБ.80066-06 01 23 предназначен для проверки синтаксических и семантических ошибок, а также для вывода контекстно-зависимых подсказок возможных вариантов ввода при написании текстов в редакторе скриптов на языке EasyFlow в процессе описания композитных приложений для МИТП CLAVIRE. Данный компонент разработан на языке C# с применением технологии Silverlight.

Компонент функционирует на аппаратных системах с архитектурой процессора x86, x86_64 и IA64. Для его работы необходимо следующее системное программное обеспечение: ОС семейства Windows NT (версии старше Windows 2000), .NET 4.0, Microsoft Internet Information Services (с версией старше 6.0), наличие компонента SyntaxEditor из библиотеки Actipro Silverlight Studio.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Компонент поддержки пользователя при разработке скриптов на EasyFlow предназначен для выполнения следующих функций:

1) проверка синтаксиса на этапе редактирования скрипта EasyFlow;
2) проверка семантики скрипта на языке EasyFlow;
3) проверка отсутствия циклов в графе workflow, построенного на основе скрипта EasyFlow;
4) предоставление контекстно-зависимых подсказок возможных вариантов ввода на этапе редактирования скрипта EasyFlow.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Принципы функционирования

Разрабатывая скрипт на языке EasyFlow, пользователь редактирует его текст с помощью компонента SyntaxEditor. Для того чтобы этот редактор мог поддерживать функции подсветки синтаксиса, обнаружения и выделения синтаксических и семантических ошибок, предоставления контекстно-зависимых подсказок, необходимо

реализовать языковые сервисы в соответствии с интерфейсами редактора. Эти сервисы регистрируются в инфраструктуре редактора и могут быть вызваны как из кода редактора, так и из модуля Ginger.

Когда пользователь делает паузу в наборе текста, редактор автоматически выполняет синтаксический анализ текста, вызывая сервис EasyFlowParser. Анализ выполняется асинхронно в фоновом потоке. После выполнения анализа вызывается обработчик события OnEditorUserInterfaceUpdate в классе FlowEditControl, который отображает синтаксические ошибки, если они есть; иначе – строит граф зависимостей и запускает семантический анализ в фоновом потоке. По завершении анализа результаты отображаются в списке ошибок, а соответствующие ошибкам места в тексте подчеркиваются.

Когда пользователь в процессе редактирования нажимает клавиши Ctrl+Space, вызывая окно контекстно-зависимых подсказок, происходит обращение к методу RequestSession класса EasyFlowCompletionProvider. Этот класс получает контекст редактирования с помощью EasyFlowContextFactory и формирует список возможных вариантов ввода с учетом информации о пакетах, загруженной из компонента PackageBase классом PackageInfoLoader.

3.2. Программная архитектура

Диаграмма классов компонента поддержки пользователя при разработке скриптов на EasyFlow приведена на рис. 3.1. Ядром является класс EasyFlowSyntaxLanguage – он предоставляет редактору все языковые сервисы, реализованные специально для EasyFlow:

1) лексический анализатор (EasyFlowLexer) – необходим для подсветки синтаксиса и работы парсера;
2) синтаксический анализатор (EasyFlowParser) – необходим для индикации синтаксических ошибок и определения контекста редактирования. Результатом работы парсера является EasyFlowParseData, содержащий атрибутированное синтаксическое дерево (AST) разобранного фрагмента.

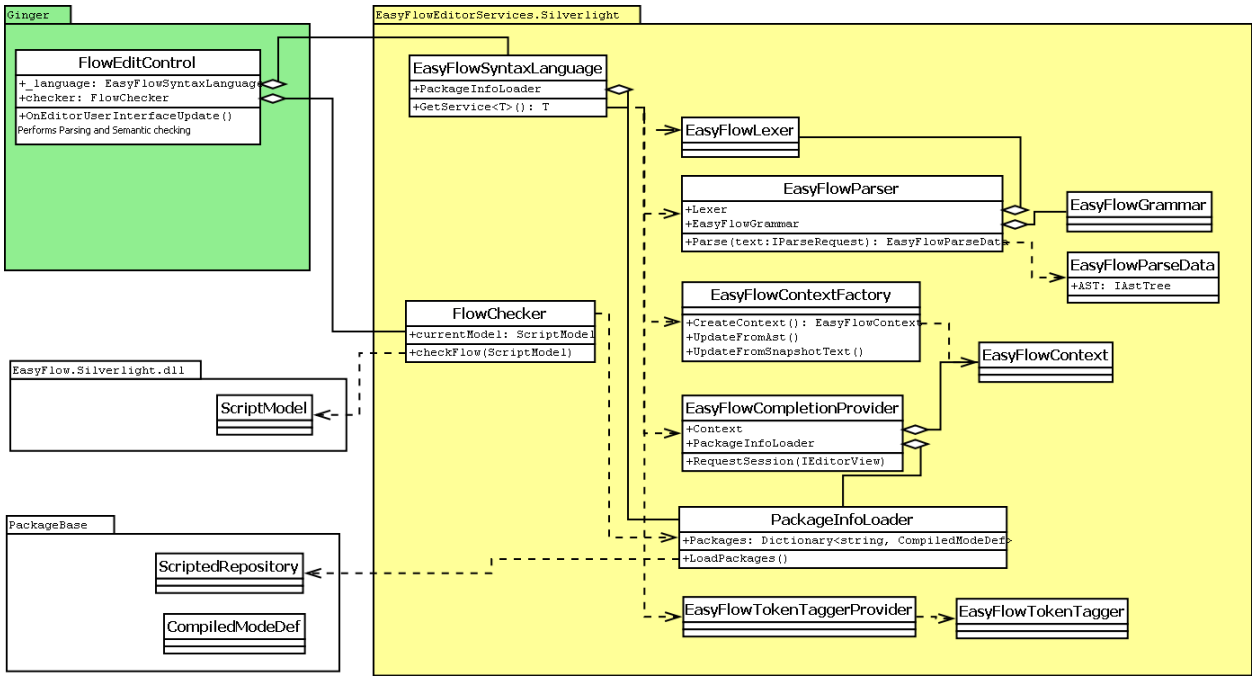

Рисунок 3.1 – Диаграмма классов компонента поддержки пользователя

Компонент поддержки пользователя при разработке скриптов на EasyFlow реализован в виде библиотеки классов платформы .NET 4.0 на языке C# с использованием технологии Silverlight и платформы Actipro LL Parser Framework.

Программы, входящие в состав библиотеки классов поддержки пользователя, написаны на языке C# и поставляются в форме стандартного проекта, созданного с использованием интегрированной среды разработки VisualStudio 2010. Основные классы, входящие в библиотеку модуля поддержки пользователя при разработке скриптов на EasyFlow, описаны в файлах с расширением *.cs.

3.3. Основные классы

Ниже приводятся сокращенные описания структуры и методов основных классов компонента поддержки пользователя при разработке скриптов на EasyFlow.

3.3.1. Класс EasyFlowSyntaxLanguage

Основной класс библиотеки . Реализует интерфейс ISyntaxLanguage. Предоставляет редактору (SyntaxEditor) все языковые сервисы, реализованные специально для EasyFlow. Конструктор регистрирует все сервисы.

Свойства

- PackageInfoLoader (тип PackageInfoLoader) – ссылка на загрузчик информации о пакетах.

Открытые методы

- GetService<T> – получение сервиса по его типу
    a) параметризованный тип T – тип возвращаемого сервиса;
    b) возвращаемое значение типа T.

3.3.2. Класс EasyFlowLexer

Лексический анализатор – сгенерирован автоматически с помощью утилиты Language Designer, входящей в состав Actipro Silverlight Studio, позволяет получать последовательность лексем типа EasyFlowTokenId (enum).

Вместо непосредственного обращения к этому классу рекомендуется использовать ITextSnapshotReader. Например,

```
ITextSnapshotReader reader = snapshotOffset.Snapshot.GetReader(snapshotOffset.Offset);
IToken nextToken = reader.ReadToken();
```

3.3.3. Класс EasyFlowParser

Синтаксический анализатор. Получает на вход лексемы от EasyFlowLexer и возвращает EasyFlowParseData, содержащий атрибутированное синтаксическое дерево разобранного фрагмента, а также список ошибок разбора. Наследуется от LLParserBase.

Открытые методы

- Parse – разбор текста.

Закрытые методы

- CreateTokenReader – создание экземпляра класса EasyFlowTokenReader, обеспечивающего связь с лексическим анализатором.

3.3.4. Класс EasyFlowGrammar

Содержит описание грамматики EasyFlow конструкциями из классов библиотеки Actipro LL Parsing Framework, заданными непосредственно на языке C#. Описываются логика восстановления после ошибок, а также структура AST.

3.3.5. Класс EasyFlowContextFactory

Фабрика контекстов. По заданному снимку рабочей области редактора языка EasyFlow определяет соответствующий контекст. Производит двухступенчатый анализ

снимка рабочей области с использованием абстрактного синтаксического дерева документа и текста, предшествующего текущему положению курсора.

Открытые методы

- CreateContext – создание контекста
    a) входной параметр snapshotOffset (тип: TextSnapshotOffset) – снимок рабочей области редактора языка EasyFlow;
    b) возвращает созданный контекст.

3.3.6. Класс EasyFlowContext

Содержит информацию о контексте редактирования – главным образом, перечисление EasyFlowContextType, а также дополнительную информацию.

Свойства

- Type (тип: EasyFlowContextType) – контекст.
- StepName (тип: String) – имя шага, в описании которого находится курсор.
- StepPackageName (тип: String) – имя пакета, выполняемого данным шагом.
- StepPackageFullName (тип: String) – полное имя пакета, включающее в себя все идентификаторы через точку.
- ExprStepName (тип: String) – имя шага в выражении справа от «=», когда параметру присваивается результат другого шага.

3.3.7. Перечисление EasyFlowContextType

- Global – вне описания шага или атрибута. Возможные варианты ввода: описание шага, описание атрибута.
- AfterRequire – после ключевого слова *require*.
- StepParams – внутри скобок с описанием параметров шага. Возможные варианты ввода: имена параметров пакета.
- StepHeaderAfterRuns – после ключевого слова *runs*. Возможные варианты ввода: имена пакетов.

3.3.8. Класс EasyFlowCompletionProvider

Формирует список возможных вариантов ввода в контекстно-зависимых подсказках, используя контекст и информацию о пакетах.

Открытые методы

- RequestSession – получить список вариантов ввода
    a) входной параметр view (тип: IEditorView) – ссылка на редактор кода;
    b) входной параметр canCommitWithoutPopup (тип: bool) – возможность автодополнения без отображения списка выбора возможных вариантов;
    c) возвращает значение типа bool – признак успешной обработки запроса подсказки.

3.3.9. Класс PackageInfoLoader

Загружает информацию о пакетах, используя компонент PackageBase, и позволяет получить как список пакетов, так и информацию о параметрах по заданному имени пакета.

Свойства

- Packages (тип: Dictionary<string, CompiledModeDef>) – список загруженных пакетов.

Открытые методы

- LoadPackages – загрузить список пакетов.

3.3.10. Класс FlowSemanticError

Обеспечивает базовый класс для всех классов семантических ошибок. От него наследуют классы:

- PackageError,
- PackageEngineRequireError,
- PackageNameError,
- PackageParameterError,
- PackageParameterTypeError,
- FlowAttributeError,
- FlowAttributeNameError,
- StepAttributeNameError,
- FlowAttributeValueError,
- StepAttributeValueError,
- FileRequirementError,
- StepIdentifierError,
- UnknownError,
- PackageEngineError,

- PackageEngineValidationError,
- PackageEngineEnableError,
- ParameterDependencyError.

Каждый из этих классов описывает конкретный тип возможных семантических ошибок, переопределяя основные методы базовых классов.

Свойства

- errorMessage (тип: String) – строка с описанием ошибки.
- errorPlace (тип: ParsedObject) – ветвь дерева внутреннего представления скрипта, в котором найдена ошибка.
- textRange (тип: TextPositionRange) – участок текста скрипта, в котором найдена ошибка.

Открытые методы

- addTail – добавить строку к тексту с описанием ошибки.
- constructErrorMessage – сформировать текст с описанием ошибки.
- needsRefinement – определить, нужно ли отдельно определять участок текста, к которому относится ошибка, или он совпадает с участком, занимаемым textRange.
- isTaggable – определить, нужно ли выводить описание ошибки и подсказку.
- getFullErrorDescription – вернуть полное текстовое описание ошибки с включенной позицией в тексте скрипта.

3.3.11. Класс FlowChecker

Используя внутренне представление разобранного скрипта (ScriptModel) и информацию о пакетах, валидирует семантику скрипта.

Свойства

- highlighter – экземпляр класса ErrorHighlighter, который обрабатывает найденные ошибки.
- messageCollection – список найденных семантических ошибок.

Открытые методы

- checkExpression – обработать выражение.
- checkPackageParams – проверить параметры шага (без проверки типов).
- checkRequiredParameters – проверить, указаны ли все необходимые параметры шага.
- checkPackageParamTypes – проверить типы параметров шага.
- clearMessages – очистить список ошибок.

- checkAttributes – проверить список атрибутов скрипта.
- checkPackageAttributes – проверить список атрибутов шага.
- checkFlow – провести проверку семантики в переданном внутреннем представлении скрипта.

3.3.12. Класс ErrorHighlighter

Обеспечивает подчеркивание семантических ошибок. Служит для уменьшения связности классов FlowChecker и SemaErrorTagger.

Открытые методы

- ErrorHighlighter – конструктор экземпляра класса, получает в качестве параметра данные, необходимые для уточнения мест ошибок в тексте.
- highLight – подчеркнуть в тексте переданную ошибку.
- endHighlighting – выполнить действия, необходимые после подчеркивания всех ошибок.

3.3.13. Класс SemaErrorTagger

Обеспечивает подчеркивание семантических ошибок. Для этого реализует интерфейс TaggerBase.

Свойства

- Instance – экземпляр класса.

Открытые методы

- packageNamesRefinement – информация, необходимая для уточнения мест ошибок в тексте скрипта.
- ClearRanges – очистить список подчеркиваемых диапазонов текста.
- refineError – уточнить место ошибки в тексте.
- addError – обработать ошибку.
- GetTags – вернуть список диапазонов текста, в которых есть ошибки. Дополнительно создать для каждой ошибки всплывающую подсказку с помощью класса SemanticErrorProvider.
- AddRange – подчеркнуть заданный диапазон текста.

3.3.14. Класс SemanticErrorProvider

Обеспечивает выдачу описания семантической ошибки при наведении пользователем курсора на текст, который ее содержит. Описание выдается в виде всплывающей подсказки.

Открытые методы

- SemanticErrorProvider – конструктор экземпляра подсказки.
- GetContent – возвращает текст подсказки.
- AddMessage – добавить строку к тексту подсказки.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Компонент поддержки пользователя при разработке скриптов на EasyFlow предназначен для функционирования на аппаратных системах с характеристиками:

- архитектура процессора – x86, x86_64, IA64;
- минимальный объем оперативной памяти – 1 ГБ;
- минимальный объем свободного пространства на жестком диске – 1 ГБ;
- минимальная тактовая частота процессора – 1 ГГц.

Компонент требует для своей работы наличия следующего системного ПО: ОС семейства Windows NT (версии старше Windows 2000), .NET 4.0, IIS (с версией старше 6.0), наличие компонента SyntaxEditor из библиотеки Actipro Silverlight Studio.

5. ВЫЗОВ И ЗАГРУЗКА

Компонент поддержки пользователя реализован в виде библиотеки классов, на которую ссылается Silverlight-приложение (Ginger), содержащее редактор текстов на языке EasyFlow. При загрузке приложения необходимо создать экземпляр класса EasyFlowSyntaxLanguage и присвоить ссылку на этот экземпляр свойству элемента управления SyntaxEditor.Document.Language. После этого парсер и другие языковые сервисы компонента станут доступны редактору. Также необходимо подписать на событие редактора EditorUserInterfaceUpdate, которое происходит всякий раз после того, как пользователь закончил ввод очередного фрагмента текста, и был произведен анализ этого фрагмента парсером компонента. В обработчике события EditorUserInterfaceUpdate

необходимо создать экземпляр класса FlowChecker и произвести проверку семантики входного текста следующим образом:

```
PackageInfoLoader loader = _language.PackageInfoLoader;
checker = new FlowChecker(loader.packageList, loader.definitionsList, loader.Packages,
loader.repository);
EasyFlowParseData parseData = _flowTextBox.Document.ParseData as EasyFlowParseData;
checker.checkFlow(parseData.ScriptModel);
```

6. ВХОДНЫЕ ДАННЫЕ

Входными данными для компонента поддержки пользователя являются файлы описания пакетов, которые могут быть считаны из локальной папки (FileSystemFileLoader) или с сервера по протоколу HTTP (HttpFileLoader). Формат файлов и способ их интерпретации описаны в компоненте PackageBase.

Интерфейсные операции языковых сервисов в качестве входных параметров принимают текст скрипта на языке EasyFlow, а также позицию курсора в этом тексте.

7. ВЫХОДНЫЕ ДАННЫЕ

Выходными данными компонента являются:

- список синтаксических и семантических ошибок во входном скрипте EasyFlow;
- информация о наличии циклов в графе workflow;
- контекстно-зависимый список возможных вариантов ввода.

Формат выходных данных определяется интерфейсами компонента Actipro SyntaxEditor.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОС	Операционная система
ПО	Программное обеспечение
МИТП	Многопрофильная инструментально-технологическая платформа

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	изме-ненных	заме-ненных	но-вых	Аннули-рован-ных					