

РАЗРАБОТКА НА ПРИЛОЖЕНИЯ В СРЕДА НА AUTODESK-ПРОДУКТИ

Мариана Трифонова¹, Елена Николова², Светлана Тончева³

¹ Минно-геоложки университет "Св. Иван Рилски", 1700 София, trifonova.m@gmail.com

² Минно-геоложки университет "Св. Иван Рилски", 1700 София, helan@mail.bg

³ Филиал на Минно-геоложки университет „Св. Иван Рилски” – София, 6600 Кърджали, svetla.23@gmail.com

РЕЗЮМЕ. В почти всички продукти на Autodesk има възможност за създаване на допълнителни команди за улесняване и автоматизиране работата на потребителя. В днешно време това може да стане на няколко езика. Изборът на програмен език зависи от вида на задачата и от операциите, които трябва да се извършат за нейното решаване. В настоящата работа се прави сравнителен анализ на различните програмни езици, използвани в продуктите на Autodesk с цел избор на подходяща програмна среда за конкретен проблем. Описан е и начин на предаване на информация между модули, разработени на различни програмни езици.

DEVELOPMENT OF APPLICATIONS IN AUTODESK PRODUCTS ENVIRONMENT

Mariana Trifonova¹, Elena Nikolova², Svetlana Toncheva³

¹ University of Mining and Geology "St. Ivan Rilski", 1700 Sofia, trifonova.m@gmail.com

² University of Mining and Geology "St. Ivan Rilski", 1700 Sofia, helan@mail.bg

³ University of Mining and Geology "St. Ivan Rilski" - Sofia, Branch of Kardzhali, 6600 Kardzhali, svetla.23@gmail.com

ABSTRACT. In almost all of Autodesk products there is opportunity for creation of new supplementary commands aimed at facilitating and automation of user's work. Nowadays this could be realized using a number of languages. Programming language's choice depends on the kind of the task and on the operations that are to be done for its solving. In the present work is performed a comparative analysis of various programming languages in Autodesk products aiming at choice of appropriate programming environment for concrete problem. A way for exchange of information between modules developed through various programming languages is described, too.

Въведение

CAD-системите на фирма Autodesk са едни от най-използваните в света. Една от причините за това е възможността за надграждане на допълнителна функционалност в тях за решаване на специфични потребителски задачи от една страна и за настройване на интерфейса по максимално удобен за потребителя начин – от друга. За целта фирма Autodesk предоставя разнообразни инструменти.

Възможности за разширяване функционалността на Autodesk-продуктите

Тъй като продуктът, с който стартира фирмата Autodesk е AutoCAD, то по-нататък в изложението за краткост „Autodesk-продукти” ще го заменим с AutoCAD – най-популярната CAD-система с общо предназначение в света. Всичко, което е изложено в настоящата статия, е валидно не само за AutoCAD, но и за останалите продукти на Autodesk, които допускат програмиране. В олекотените версии на продуктите, т.е. при „LT” версиите отсъстват всякакви средства за програмиране с изключение на езика DIESEL.

Основните начини за разширяване функционалността на AutoCAD са:

1. Създаване на скрипт-файлове. Представяват външни текстови файлове с разширение scg, които се създават с произволен текстов редактор (например Notepad) и представляват набор от вградени команди на AutoCAD (по една на ред). Могат да се изпълняват автоматично както при стартиране на системата, така и по средата на сеанса на работа чрез използване на специална команда. Това е елементарно средство за автоматизиране на повтарящи се, често използвани команди на AutoCAD. Основният им недостатък е, че не допускат никакви междинни изчисления, както и вход от потребителя, т.е. всичко е статично.
2. Използване на меню-макроси (да не се бърка с макроси на Visual Basic for Applications – VBA). Създават се от потребителя. В тях могат да се включват както вградени команди на системата, така и елементарни изрази на AutoLISP, а така също и изрази, написани на езика DIESEL (Direct Interpretatively Evaluated String Expression Language). Тези макроси могат да се „прикачат” към потребителски бутони, разположени в някоя от съществуващите или в нова лента с инструменти (Toolbar),

- както и в Ribbon-лентата. Езикът DIESEL разполага с 28 функции предимно за работа с низове. Позволява да се формират низове с променливо съдържание, зависещо от някакви условия. Резултатът е низ, който се интерпретира от AutoCAD като команда. Езикът за първи път се появи в AutoCAD R12.
3. Action-макроси (*автоматично* генерирани макроси за разлика от горните, които се разработват от потребителя). Това са последователности от команди, които се създават с помощта на инструмента Action Recorder . Тази възможност за първи път се появи в AutoCAD 2009. Така създадените макроси могат да се редактират, а в следствие и да се пренасят в някой панел с инструменти или в специално меню.
 4. AutoLISP. Това е интерпретатор, притежаващ основните конструкции на езика Lisp, допълнени с възможности за използване на командите на AutoCAD. AutoLISP е най-старият от вътрешните езици за програмиране в AutoCAD. За първи път се появи в 1986г. в AutoCAD 2.18. С негова помощ е създадено богато програмно осигуряване под формата на допълнителни команди за AutoCAD от хиляди потребители и фенове на AutoCAD по света. Част от тези допълнителни команди се разпространяват от Autodesk, обособени в самостоятелен модул (напр. Express Tools или под някакво друго име, зависи от версията на продукта). От AutoCAD R12 към AutoLISP е добавен и DCL (Dialog Control Language) за създаване на диалогови прозорци за приложения, написани на AutoLISP. Трябва да се отбележи, обаче, че DCL не предоставя визуални среди и възможностите му за създаване на диалогови прозорци са силно ограничени.
 5. Visual LISP. Въпреки името, Visual LISP не предоставя среда за визуално програмиране. Най-общо, това е AutoLISP, разширен с технологията ActiveX. С Visual LISP могат да се създават функции, *компилирани до някакъв междинен код* (разширение fas), за разлика от AutoLISP, който е класически интерпретатор. Visual LISP беше добавен към AutoCAD в 2000г.
 6. VBA (Visual Basic for Applications). VBA беше вграден в AutoCAD R14. За разлика от Visual LISP, VBA представлява изцяло визуална среда за програмиране с пълна поддръжка на ActiveX-технологията и възможности за използване на dll-библиотеки. Но от версия AutoCAD 2010 насам VBA-средата не се включва по подразбиране. Autodesk постепенно се отказват от поддържането на VBA, отдавайки приоритет на .NET.
 7. ObjectARX. Това е допълнение към Microsoft Visual Studio. Съдържа специални библиотеки, header-файлове и спомагателни инструменти, предназначени за създаване на програмно осигуряване на C++, функциониращо в среда на Autodesk-продукти под Windows. ObjectARX замести ADS (Advanced Development System) – библиотеки, които се използваха в операционна среда DOS. Както ADS, така и

ObjectARX не се инсталират с инсталацията на AutoCAD, а е необходимо допълнително да се изтеглят от официалния сайт на Autodesk.

8. Microsoft .NET Framework. Хронологически погледнато, това е последната възможност, предоставена от Autodesk, за разработване на надстройки за техните продукти. Може да се използва за версия 2005 и следващи.

Първите три от гореизложените начини за обогатяване функционалността на AutoCAD са сравнително примитивни и не дават възможност за създаване на по-сложна допълнителна команда. За сметка на това не изискват познания върху език за програмиране с изключение евентуално за езика DIESEL. DIESEL е силно различен от класическите процедурни и функционални езици за програмиране, сравнително елементарен и с ограничен набор от функции, което улеснява прилагането му. Така, че подходите, изложени в първите три точки биха се използвали основно в два случая:

- работа с олекотени версии на продуктите (напр. AutoCAD LT) поради невъзможност за програмиране на друг език;
- използване от потребители-непрограмисти.

VBA не само не се развива вече от Autodesk, но и от 2010г. не се включва по default при инсталациите на AutoCAD. Поради тази причина няма да разглеждаме по-надолу в изложението възможността за създаване на допълнителни команди в AutoCAD чрез VBA. Въпреки това, програмно осигуряване, разработено на VBA, може да се изпълнява в среда AutoCAD.

Сравнение AutoLISP/Visual LISP и ObjectARX/.NET

AutoLISP и Visual LISP също вече не се развиват от Autodesk. Но те се предоставят по подразбиране с инсталацията на AutoCAD и могат да се използват непосредствено във всеки момент без допълнително инсталиране на други модули.

1. Среда за разработка. AutoCAD има вградена IDE-среда за създаване и тестване на код на AutoLISP/Visual LISP. IDE за .NET и ObjectARX в AutoCAD няма. Тя се инсталира отделно – обикновено това е MS Visual Studio.
2. Бързодействие. Приложенията на .NET и ObjectARX като правило работят по-бързо, отколкото тези на AutoLISP/Visual LISP. Това в особена сила важи за функции, в които преобладават аритметичните операции. Но трябва да се отбележи, че в някои случаи на изчертаване на графични примитиви и извличане на информация от графичната база данни скоростта на изпълнение е съизмерима.

3. Графичен интерфейс. При използване на C++ и особено на .NET-език диалогови прозорци се създават много лесно, като има възможност за използване на разнообразни контроли. В същото време при AutoLISP/Visual LISP възможностите за графичен интерфейс с потребителя са силно ограничени поради ограничените възможности на DCL. Независимо от факта, че в последно време AutoLISP/Visual LISP допускат и създаване на диалогови прозорци чрез езиците ObjectDCL или OpenDCL, то последните не са вградени в средата на AutoCAD, платени са и нямат мощта на визуалните среди на MS Visual Studio. Но пък има възможност на някой .NET-език (напр. C# или VB.NET) да се създаде GUI (графичен потребителски интерфейс) за използване от функции на AutoLISP и VisualLISP.
4. Взаимодействие с Windows-приложения. Такава възможност в AutoLISP напълно липсва, във Visual LISP е ограничена, докато при .NET и ObjectARX е напълно развита.
5. Преминаване към нова версия на AutoCAD. При преминаване към нова версия на Autodesk-продуктите функции, написани на AutoLISP/Visual LISP на практика не се налага да се променят. Ако изобщо се направи промяна, тя обикновено е за оптимизиране на някоя функция чрез използване на нова възможност на езика, т.е. промяната не е наложителна. За код, написан на .NET и ObjectARX често се налага прекомпилиране, а в някои случаи и промяна на програмния код. Това се дължи на факта, че Autodesk постоянно развива .NET API за разлика от AutoLisp и VisualLisp.

И четирите подхода: AutoLISP, Visual LISP, ObjectARX и .NET-езиците за програмиране могат да се прилагат за създаване на допълнителни команди към продуктите с цел изпълнение на специфични задачи. Но всеки един от тези програмни езици е по-удобен за решаване на определени специфични задачи, като:

- AutoLISP и Visual LISP са по-удобни за създаване на нови команди с елементарен графически интерфейс, в които няма много аритметични изчисления, а преобладава използването на вградени команди на

AutoCAD или извличане на информация от графичната база данни на системата. Трябва да се има предвид, че AutoLISP е интерпретатор, докато функциите, написани на Visual LISP се компилират в междинен код (разширение fas) и затова тяхното изпълнение е по-бързо. Използването на команди на AutoLISP в диалогов режим на AutoCAD в някои случаи е много удобно и полезно.

- C++ чрез ObjectARX и .NET-езиците – за създаване на нови обекти, класове, методи, пространства, интерфейси, както и модули за обработване на различни системни събития и драйвери за външни устройства.

Заклучение:

1. Използвайки съвременни езици за програмиране може съществено да се разшири функционалността на Autodesk-продуктите.
2. Детайлното познаване на възможностите на езиците за програмиране в среда на Autodesk позволява да се оптимизира процеса на създаване на допълнителни надстройки (plug-ins).

Литература:

<http://www.autodesk/Download.htm>
<http://www.caddsoftsolutions.com/Download.htm>
http://docs.autodesk.com/ACDMAC/2014/ENU/Mac_Sandstone_Dev_Help/Developer%20Guide/index.html
 OBJECTARX DEVELOPER'S GUIDE
<http://www.afralisp.net/visual-lisp/>
<http://www.cadtutor.net/forum/forumdisplay.php?21-AutoLISP-Visual-LISP-amp-DCL>
<http://www.afralisp.net/visual-lisp/tutorials/reactors-part-1.php>
<http://lispexpert.blogspot.com/p/chapter-15.html>

Статията е препоръчана за публикуване от кат. „Информатика“.