

Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика



uDraw(Graph)

Курсов проект по Софтуерни архитектури

Михаил Николов (ФН: 80527)
студент в 4-ти курс, Компютърни науки,
ФМИ към СУ „Св. Климент Охридски“

10 февруари 2014г.

uDraw(Graph)

uDraw(Graph) е инструмент улесняващ създаването на графи, диаграми, графики и други структури. С цел автоматизиране на процеса и минимизиране на работата се използват модели (layouts). Това прави работата с uDraw(Graph) доста по-бърза.

uDraw(Graph) поддържа и свой API, който позволява лесното вграждане и използване на инструмента с цел визуализация на данни в новоразработени програми.

uDraw(Graph) е използван от хиляди клиенти и водещи компании в света от 1992 година насам. Системата е разработка на Университета на Бремен и е достъпна за свободно ползване. Снимка на uDraw(Graph) можете да видите в *[Виж приложение 1]*.

uDraw(Graph) има поддръжка освен за Windows NT / 2000 / XP и Mac OS (10.3.8 или по-нова) и за всички съвременни Linux-базирани дистрибуции, Solaris и FreeBSD. Възможно е програмата да работи и на версии различни от споменатите, но за тях няма официална поддръжка. Към момента uDraw(Graph) е налична само за x86 архитектура.

Хардуерните изисквания са съвсем минимални. За стартирането на програмата са необходими поне 100 MHz процесорна мощ, 64 MB RAM и 20 MB свободно дисково пространство. За стабилната работа на софтуера са необходими поне 300 MHz процесор, 128 MB RAM и 40 MB дисково пространство.

uDraw(Graph) не зависи от други програми и за използването му не е необходимо инсталирането на допълнителен софтуер. Въпреки това, за да е възможно коректното принтиране на създадените диаграми и графики, под Windows е нужно да има инсталиран PostScript® printer или друг софтуер, който може да принтира PostScript®.

uDraw(Graph) е софтуер, позволяващ лесното създаване и визуализиране с високо качество на насочени графи. Графите са често използвани за представяне на обекти, наричани още „възли“, и отношенията между тях, представяни от „ребра“ между възлите. Използват се за визуализиране на йерархии, зависимости в структури, мрежи, конфигурации, потоци от данни и др. Въпреки широкото им използване, визуализацията на графи не е сред основните свойства на модерния компютърен софтуер. Така че, дори и днес потребителите често се налага да се сблъскват с опростени текстови представяния на тези структури или съвсем прости графики, които са по-трудни за разчитане и трудно „хващат окото“. Причината за това е, че създаването на висококачествени графови модели и тяхната визуализация е трудна задача.

uDraw(Graph) осигурява напълно автоматизирано създаване на графи с богат набор от стилове, икони и фигури за представяне на възлите и ребрата. Създадените графи се съхраняват в специален файлов формат или могат директно да бъдат изпратени на външни програми посредством API.

Но uDraw(Graph) може да предложи повече от обикновените библиотеки за визуализация. С интуитивния си потребителски интерфейс и богатата функционалност, позволява лесното и бързо представяне дори на графи с много на брой и сложни връзки между възлите. Цялата функционалност е достъпна както през потребителския интерфейс, така и пред предоставения API. Това прави особено лесно използването на uDraw(Graph) във външни приложения, без да е нежно използването на алгоритми или познания по компютърна графика. С uDraw(Graph) потребителите могат да контролират свързаните си приложения посредством интуитивна визуализация на графац *[Виж приложение 2]*.

Концепции използвани в uDraw(Graph)

Графи

Представянето на графи е основна функционалност на uDraw(Graph). Графите са структури, състоящи се от обекти (възли) и отношения (ребра) между тях. При насочените графи всички ребра имат посока и съответно всяко ребро си има родител (възела, от който произлиза) и дете (възела, към който сочи). Представянето на насочените графи в uDraw(Graph) става чрез разпределянето на възлите на хоризонтални нива, така че всички родители са над техните деца, и всички ребра сочат надолу. Ребрата обикновено се изобразяват като стрелки, сочещи към възела дете. Този вид представяне се нарича йерархична визуализация на насочен граф [Виж приложение 3].

Въпреки, че йерархичното представяне е възможно само за ациклични графи (графи, чиито ребра не образуват цикъл, т.е. следвайки ребрата не можем да стигнем до възела, от който сме тръгнали), uDraw(Graph) може да визуализира и циклични графи, при които всеки цикъл може да бъде нарушен чрез обръщането на посоката на точно едно ребро. По този начин графът се превръща в ацикличен и може да бъде обработен от алгоритъма за йерархично представяне.

Графите се зареждат в uDraw(Graph) чрез формата наречен „term representation“. Това представлява текстово (посредством ASCII символи) представяне на графа. То се съхранява във файл с разширение „udg“. Този формат поддържа всички видове насочени графи: циклични и ациклични, празни графи, графи със само едно ниво (т.е. списък с възли без никакви ребра), графи с повече от едно ребро между два възела и дори ребра свързани със само един възел. Тъй като term representation формата е всъщност текстово представяне на графа, то udg файловете могат да бъдат създавани и чрез обикновен текстови редактор. Но обикновено те се генерират автоматично от програмата или от приложение свързано чрез API.

Приложения използващи API

Приложението може да бъде свързано с API през менюто „Файл“ в графичния интерфейс на програмата или през командния ред. На първо време редакторът добавя няколко менюта и бутона към потребителския интерфейс на uDraw(Graph) и позволява добавянето или премахването на възли и ребра, както и промяна на някои техни атрибути. API изпраща събитията за потребителска активност директно на приложението. Например, ако потребителя избере възел и кликне на бутона „Изтрий възел“ в менюто, тогата редактора ще разбере за тези две събития и ще ги обработи. Редакторът държи данните за структурата на графа и отразява промените направени от потребителя, когато получи информация за някое събитие. След това, редакторът изпраща актуализирания граф обратно до uDraw(Graph), за да обнови визуализацията му.

Този принцип може да бъде използван от всяко приложение свързано с API на uDraw(Graph). Приложението има цялата отговорност за контролирането структурата на графа. uDraw(Graph) няма правомощия върху нея.

Представяне на графа

Графът се визуализира в основния прозорец след зареждане на файла съдържащ структурата му. В йерархичното представяне, възлите от един йерархичен слой се подреждат хоризонтално на едно ниво, така че всички ребра да сочат надолу. Ребрата свързващи възли в отдалечени нива се представят като няколко ребра с фалшиви възли между тях на всяко едно

пропуснато от оригиналното ребро ниво. Фалшивите възли не са отразени във файла съдържащ структурата на графа, а се въвеждат автоматично. Обикновено те са невидими, но се отбелязват като малки точки, когато въпросното ребро бъде избрано или когато бъдат посочени с мишката.

От менюто Layout/Orientation в потребителския интерфейс (както и през API) може да се избере ориентацията на графа при визуализирането му. По подразбиране тя е отгоре-надолу, което означава, че в най-горния слой са възлите без родители и всички ребра сочат надолу, към техните деца. Другите възможности са за отляво-надясно, отдясно-наляво и отдолу-нагоре. Принципът с подредбата на възлите в слоеве е същият.

След изчисляване ориентацията и представянето на графа, той се визуализира в основния прозорец. Всички възможни опции на представянето на графа, като цветове и размери на възлите и ребрата, шрифт и т.н. се задават от техните атрибути, записани в term representation файла. Например, атрибутът „object“ приема за стойност стринг, който се поставя във всеки възел. Размерите на възела се адаптират автоматично към текста, така че той да се събира в тях. Също така се поддържа и текст на няколко реда. По подобен начин могат да се променят много от параметрите по подразбиране, като например разстоянието между възлите в даден слой и разстоянието между слоевете.

Ако потребителят избере възел от графа с мишката, то той ще получи сянка, което прави разпознаването на селектираните възли по-лесно. Чрез задържане на бутона Shift могат да бъдат избирани по няколко възела едновременно. Ребрата също могат да бъдат селектирани, но само по едно в даден момент. Тази функция е удобна за проследяване на пътя между възли при по-дълги ребра.

Визуализация

Правилата за визуализация важат за цели групи, а не по единично за отделни компоненти от графа. Това позволява лесния контрол на визуалното представяне на цели групи от възли и ребра, като това става централизирано. Например, можем да направим всички възли от даден тип (типът е атрибут, който се задава на всеки възел) с различна форма и цвят от останалите възли в графа. Ако решим на по-късен етап да сменим цвета, ще бъде достатъчно да го направим от едно място, където сме дефинирали този стил без да е нужно да променяме нищо в структурата на графа. Тази промяна ще се отрази на всички графи, съдържащи възли от съответния тип.

Има и възможност за презаписване на тези настройки. Можем да задаваме така наречените „локални атрибути“, които да задават различни настройки на въпросния възел, въпреки, че той може да принадлежи към група с вече зададени общи настройки. Локалните промени ще се приложат само върху въпросния възел.

За да можем да боравим с тези възможности на uDraw(Graph) свободно, е необходимо добре да познаваме работата на приложението и в частност точно в какъв ред се оценяват дадените настройки кои от тях в крайна сметка се прилагат за всеки един компонент в графа. Когато нямаме налични правила за визуализация, се прилагат стиловете (цветове, форми и т.н. на възлите и ребрата) посочени от индивидуалните атрибути на възлите и ребрата. Когато и те не са налични, предвид се вземат настройките по подразбиране. Но когато имаме зададени правила за визуализация, реда в който се оценяват атрибутите се променя. В този случай първо се разглеждат стойностите на индивидуалните атрибути за всеки възел или ребро. В случай, че няма такива, се прилагат правилата за групата, в която попада, въпросният елемент.

Фини настройки на моделите (layouts)

Автоматично генерираните графи никога не са перфектни. За да се преодолее този недостатък, uDraw(Graph) предлага възможност за фина настройки на графа. Тя може да се

прави по всяко време. Фните настройки се използват най-често за премахване на кръстосани ребра, които не могат да бъдат премахнати от алгоритъма на модела. След фината настройка, алгоритъмът може да се стартира отново, за да премахне други кръстосвания на ребрата, които могат да се получат. Следващите редове описват как протича фината настройка в модели с ориентация отгоре-надолу. За да приложите стъпките и към друга ориентация е достатъчно да размените термините „хоризонтално“ и „вертикално“.

Фините настройки работят чрез селектиране на възел от графа с левия бутон на мишката. Необходимо е бутонът да се задържи натиснат, докато влачим възела към новата му позиция. Истинските възли могат да бъдат местени както по хоризонтала, така и по вертикал, докато фалшивите (създадени от алгоритъма за визуализация) могат да бъдат местени само по хоризонтала. Фините настройки в uDraw(Graph) са повече от преместване на възли в друга позиция, защото преместването на възел автоматично коригира положението на съседните възли в реда, за да се запази йерархичната структура на графа. Например, ако даден възел бъде преместен по хоризонтала и достигне съседен възел, двата възела трябва да разменят позициите си или съседният да се отмести настрана, в зависимост от текущите настройки. Освен това, ако възел бъде преместен в следващия или предишния слой, тогава подграфите образувани от децата му и родителите му също трябва да се изместят, за да се запази йерархията, така че всички ребра да сочат надолу.

При преместването на възел по хоризонтала (наляво или надясно) има три режима на работа, които могат да бъдат определяни от менюто „Layout Algorithms“ или чрез три икони, намиращи се в лявата част на основния прозорец. Режимът на фини настройки дефинира поведението, когато преместения възел докосне своя ляв или десен съсед в своето ниво. Съседният възел се счита за докоснат, когато се достигне минималното определено разстояние между възлите. В режима по подразбиране „Preserve Node Order“ докоснатият възел се отмества с толкова, че да се запази реда на възлите в слоя, докато това е възможно. В режима „Modify Node Order“ преместеният и докоснатият възел разменят позициите си, когато разстоянието между тях стане прекалено малко. Този режим е полезен за преназдравяване на възлите в даден слой, например, за да бъде премахнато кръстосване на ребра. Третият режим е „Fixed Neighbour Nodes“, при който фините настройки се преустановяват, докато възлите не се докоснат напълно. Това е полезно, когато искаме да запазим позицията на съседните възли.

Вертикалното преместване на възел между слоевете също е възможно. Това може да се прави само с истинските възли. Фалшивите, създадени от алгоритъма за визуализация не могат да се местят вертикално. По време на местенето на възела, още преди той да бъде пуснат в някое ниво, динамично се преизчислява и променя структурата на графа, така че да се запази йерархичната му структура и всички ребра да сочат надолу.

Status формат

Graph status е алтернативен формат за графи в uDraw(Graph). Другият поддържан от uDraw(Graph) формат е така нареченият „term representation“, който вече разгледахме. Status файла трябва да завършва с разширение *.status*. Освен чистата структура на графа, която се съдържа в „term representation“ представянето, статус файла съдържа информация за модела на графа и всички текущи настройки на потребителския интерфейс. По този начин потребителят може да възстанови предишна сесия, като зареди статус файла. Файлът съдържа информация само за един граф.

Трябва да се има предвид, че статус файла съдържа информация специфична за определена версия на uDraw(Graph). Обикновено новите възможности за съхранение на данни, които предлагат следващите версии могат да се запазят в статус файла само след като той бъде надграден. Това трябва да става за всяка нова версия. Това означава, че дадена версия на uDraw(Graph) не може да прочете статус файл предназначен за по-нова версия. Но старите статус файлове, генерирани от по-стари версии могат да бъдат четени, с уговорката,

че не можем да се възползваме от възможностите на по-новата версия. Лесно може да се провери даден статус файл за коя версия е, като просто файла се отвори с произволен редактор. Подобно на „term representation“ и статус файловете са в текстови формат лесен за четене и редакция „на ръка“.

Взаимодействия

uDraw(Graph) разполага с някои много полезни интерактивни операции за работа с визуализацията на графа. Някои от тях вече разгледахме, като така наречените фини настройки.

Една друга възможност е „Scaling“. Това е опция предлагаща възможността за преглед на граф, който не се събира целия на екрана. Може да му бъде зададен мащаб между 1% и 100%, с уговорката, че при мащаб по-малък от 100% някои подробности няма да се показват, като например текста във възлите, ако има такъв. Но за сметка на това, всички операции за работа с графа са налични.

Multi-View и Multi-Graph

uDraw(Graph) има вградена поддръжка на multi-view и multi-graph. Това означава, че можете да работите по един граф в различни прозорци (views) или да заредите няколко графа в отделни прозорци.

Потребителят може да отвори същият граф в нов прозорец. Промените наложени в един от прозорците ще се отрази на визуализацията на графа и в останалите прозорци.

В същото време в същият основен прозорец може да бъде отворен втори граф.

Максималният брой отворени прозорци, който се поддържат от uDraw(Graph) е 64.

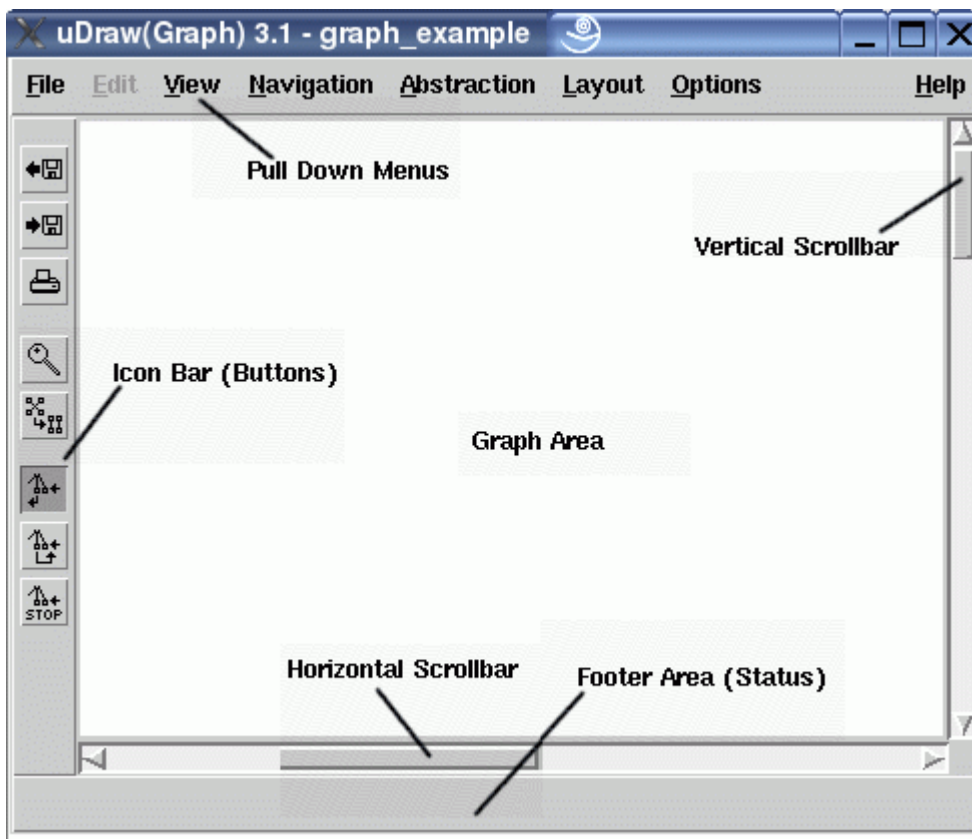
Drag & drop

Програмният интерфейс на uDraw(Graph) позволява drag & drop на обекти от графа за по-добра поддръжка. Потребителят може да създаде ново ребро само с влачене на мишката от единия възел до другия. По същият начин се създава и нов възел-дете. За целта трябва да влачите мишката от възела до някое празно пространство. Така се създава ново ребро, което сочи към новия възел. Нов възел може да се създаде и само с клик на празно пространство, като първоначално няма да бъде свързан с останалата част от графа чрез никакво ребро.

Разглеждайки в детайли функционалността на uDraw(Graph) видяхме и основните причини, които правят този софтуерен продукт предпочитан от множество хора и компании. Лесното създаване на графовите структури, още по-лесното им променяне и визуализация, както и наличието на API са безспорни предимства.

Приложения

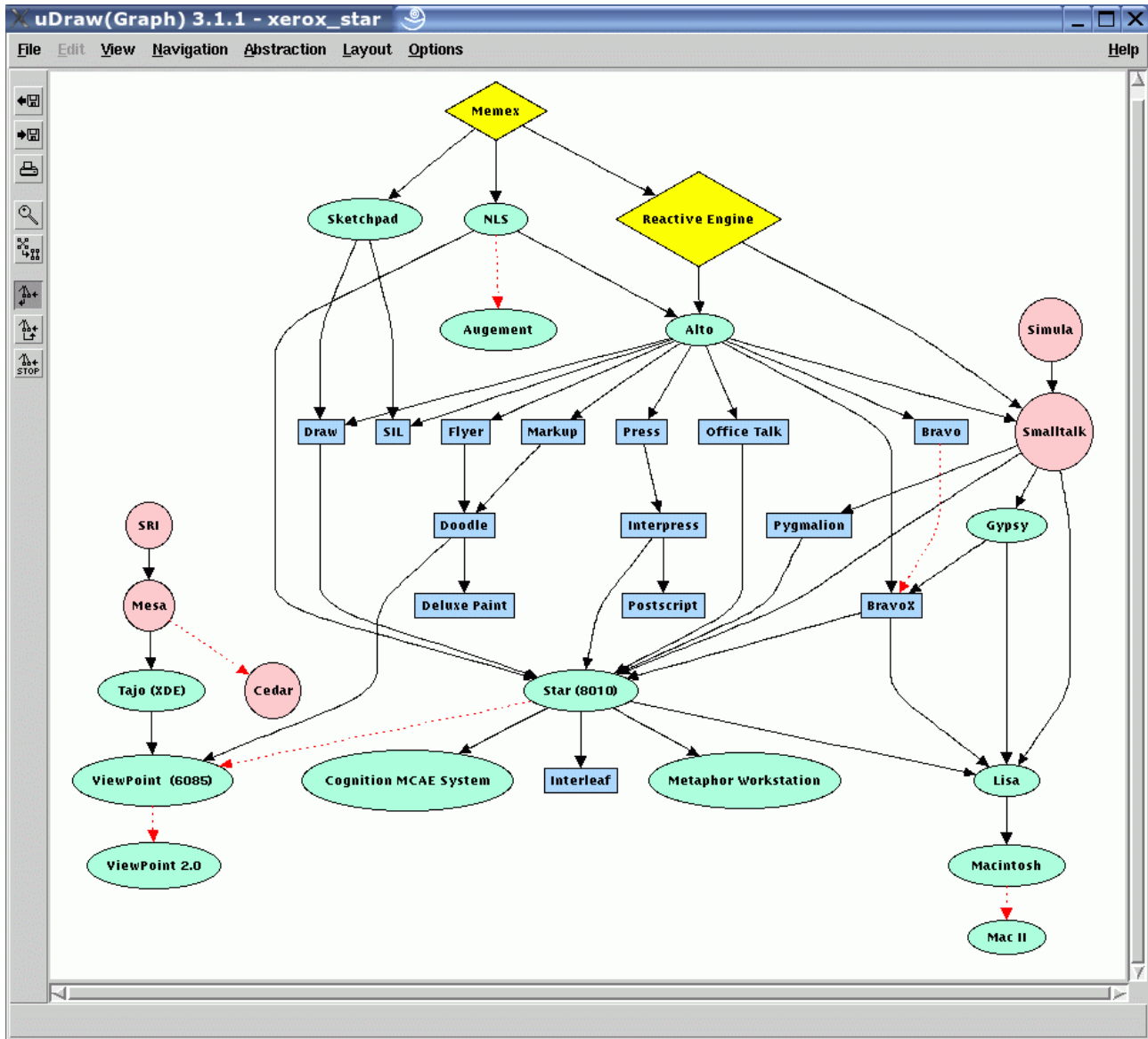
Приложение 1:



Источник:

<http://www.informatik.uni-bremen.de/uDrawGraph/en/uDrawGraph/screenshots/screenshots.html>

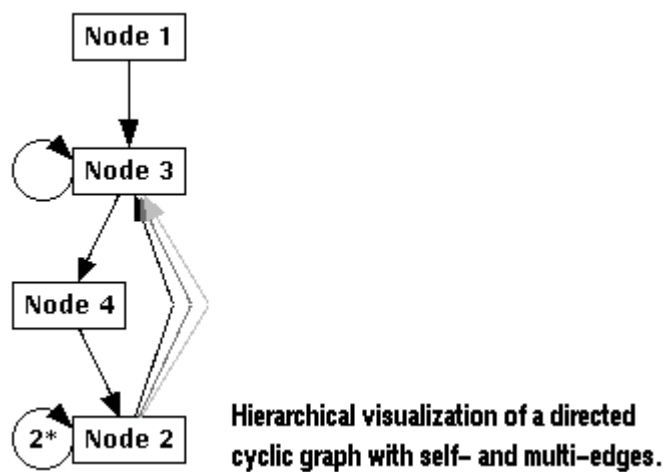
Приложение 2:



Источник:

http://www.informatik.uni-bremen.de/uDrawGraph/en/service/uDG31_doc/gui_base.html

Приложение 3



Источник:

http://www.informatik.uni-bremen.de/uDrawGraph/en/service/uDG31_doc/conc_gra.html