

СОФТУЕРНИ РЕШЕНИЯ ЗА ВИЗУАЛИЗАЦИЯ В ПОМОЩ НА ОБУЧЕНИЕТО

Ивайло Буров, Иванка Бурова

SOFTWARE VISUALIZATION SOLUTIONS IN SUPPORT OF EDUCATION

Ivailo Burov, Ivanka Burova

ABSTRACT: This article reflects upon current questions and answers arising in the process of a wider study on cognitive processes, mechanisms of their formation, opportunities for optimization and its application in the learning process in terms of the usage of modern information and communication technologies in education. The findings and conclusions expressed in this article are of informative character and show the current stage of this study, thus they are therefore open to discussion, further research and development.

KEY WORDS: software, visualization, education, knowledge, technologies, UML, Mind Maps, Concept Maps.

Изследването е финансирано по проект № РД-08-88/03.02.2017 г. от параграф на фонд „Научни изследвания” на ШУ „Епископ Константин Преславски”.

1. Парадигми, пресечни точки на парадигмите

Модерното обучение е бързоразвиващ се динамичен процес. В процеса на развитието на педагогическите науки възникват парадигми, представящи различни модели на обучение. Някои от тия модели са преобладавали за определени периоди, но с развитието на обществото възникват нови условия, което се явява причина тези модели да не съответстват напълно на появилите се от това нови изисквания към обучението. Това води до смяна на парадигмата. За да бъде възможен и успешен такъв преход е необходимо плавно преминаване от едни утвърдени вече модели към нови. Същевременно, полезните и адекватни на съвременните условия достижения от предходни модели трябва да могат да бъдат използвани и в новата парадигма. Това дава опора на много изследователи да търсят пресечна точка на парадигмите.

Парадигмите на образованието са модели или набор от предположения, концепции, ценности и практики, които представляват различни форми за представяне на основните им елементи в областта на образованието и обучението. Понастоящем парадигмите се тълкуват така, че се търси противоречие между тях, но едновременно с това се отчита, че всяка една има основания и не може напълно да бъде пренебрегната за сметка на друга (Радев, П., Традиции и съвременност в образованието) [1].

Развитието на информационните и комуникационни технологии и навлизането им в обучението поражда изискването за интердисциплинарност, което е израз на пресечна точка между постмодернистичната и предмето-центрираната (учебно-дисциплинарна) парадигма.

2. Интердисциплинарен подход

Изследването на тези процеси в непедагогическите научни направления води до аналогични резултати. Наблюдават се различни парадигми, а също така техни пресечни точки. Ярък пример за това са информационните технологии. Различните парадигми в това направление са повече свързани с разработването на такива технологии, отколкото с приложението им, но се наблюдават полезни приложни решения, които са нагледен пример за реализация на една или друга парадигма в това направление. Познанието, че такова решение е израз на съществуваща известна парадигма, спомага за по-бързо усвояване на работата с такъв приложен продукт. В повечето случаи такива интердисциплинарни знания отсъстват не само при ползвателя на приложни продукти, а дори и при тесни специалисти разработчици, което води до частични познания и пропуски. Поради тази причина в съвременния процес на обучение са отправени нови изисквания към преподавателя. Изготвянето на лекционни и

семинарни курсове в тези условия изисква дълбоко познание на тези процеси, а също така и представянето им на обучаваните в достъпен формат.

3. Фактори, обуславящи необходимостта от визуализация

От проведените авторски наблюдения при организиране на лекционни курсове и семинарни занятия със студенти проличава един от възпрепятстващите обучението фактори – широко застъпена чуждоезична терминология. Въпреки употребата на такава терминология даже в ежедневието може да се отчете определено ниво на изгубено семантично значение в приложението на тази терминология. В такъв случай голяма част от езиковите конструкции и дефиниции (определения) остават неразбрани или водят до погрешно построяване на класификационни модели при обучавания. В едно допълнително изследване за причините, водещи до такива нежелани ефекти (което излиза извън обема на настоящата статия), се достига до някои от тях: различен начин на словообразуването, синтаксиса и семантичния строеж в езиковите конструкции на естествените езици. Точно същите нежелани ефекти се срещат при интердисциплинарно обучение, а дори и само при комуникация между специалисти от различни научни направления, боравещи с утвърдена терминология в съответното направление. Това поражда изискването за създаване на универсална конвенция при комуникацията. В някои области, където работят специалисти от множество научни направления като военно-космическата промишленост са направени такива опити, в резултат на които възникват езици за визуално програмиране, улесняващи работата на участниците в екипа по определен проект.

От друга страна при разглеждане на теориите за формиране на познанието и приложението им в обучението, ясно се различава класификационният подход, но в редица случаи таксономията (наука за принципите на класификация и систематизация на знанията) е подлагана на критики, поради известни ограничения, възпрепятстващи възможностите за реализация на определен кръг задачи. Например при работа с обектно-ориентирани програмни езици предварително се създава класификация, като класовете не могат да бъдат дефинирани в процеса на изпълнение на програмата, а само при нейната разработка. Това може да ограничи възможностите за промяна на поведението на програмата при постъпване на нови данни или знания в системата. При системите с изкуствен интелект съществуват методики за обучение с учител или без учител. Обучението с учител е свързано с класификация, докато това без учител се приближава повече до клъстеризацията. Това може да бъде съпоставено с придобитите знания без учител при обучението на студенти и ученици и обучението им с учител. В съвременния процес на обучение, обучаваните изучават дадено научно направление с известен запас от придобити знания без учител, като впоследствие прибавят към тях знания, придобити с учител. Встъпването с предварителен тезаурус в процеса на обучение може да катализира познавателния процес в едни случаи или да го възпрепятства в други. Възниква въпросът може ли да бъде проверено състоянието на това предварително познание. Някои методики използват встъпителни тестове за определяне на равнището на предварително познание, но това са по-скоро количествени методи на измерване. Не се отразяват взаимовръзките на изградените вече понятия и техните отношения. Визуализацията на този процес би спомогнала за качествено нов поглед към предварително придобитите знания на студента/ученика.

4. Утвърдили се в практиката решения за визуализация

Съществуват множество решения, доказващи предимствата на визуалния подход при изразяването на зависимости или тенденции. Достатъчно е само бегъл поглед към визуално представена графична тенденция на един процес във времеви период, за да се придобие бърза и цялостна представа за този процес. Визуализацията се използва широко за изследване на динамични процеси, които са с невизуален произход или не подлежат на сетивна регистрация. Примерно при софтуерни решения за обработка на звук фонограмите са визуалното представяне на записан аудио фрагмент. В този пример може да се наблюдава цялостната представа за записан аудио фрагмента, без необходимост от последователното му прослушване. Подобни решения се прилагат и при продуктите за видео обработка, като динамично

променящият се видео и аудио сигнал се разполагат в паралелни видео и аудио пътеки. По този начин се придобива цялостна представа за наличното съдържание и неговата подредба.

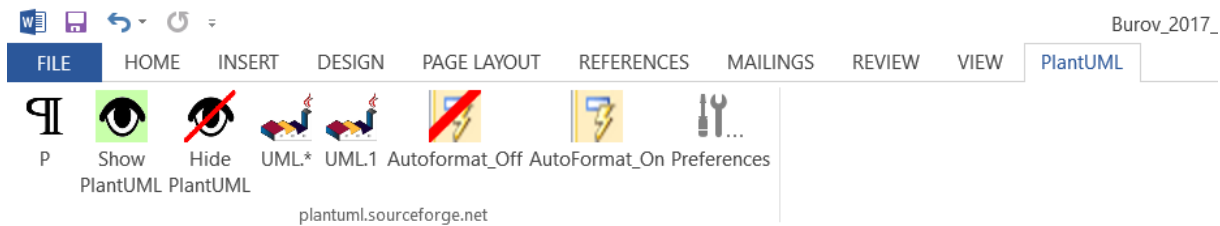
a. UML решения

UML (от англ. *Unified Modeling Language*) Унифициран език за моделиране е графичен език за визуализиране използван в софтуерното инженерство, при създаването на който се цели осигуряването на универсален стандартен път за визуализация и дизайн на системи. Обединява допреди това съществуващите стандарти, като по-късно ги доразвива. UML е развит и стандартизиран от индустриалния консорциум ObjectManagement Group (OMG) под ръководството на Ръмбах, Бууч и Джакобсън. В OMG участват водещи в информационните технологии компании като Hewlett-Packard, IBM, Sun Microsystems, Apple Computer, American Airlines и Data General. В UML са заложени възможности за визуализация на база на следните диаграми:

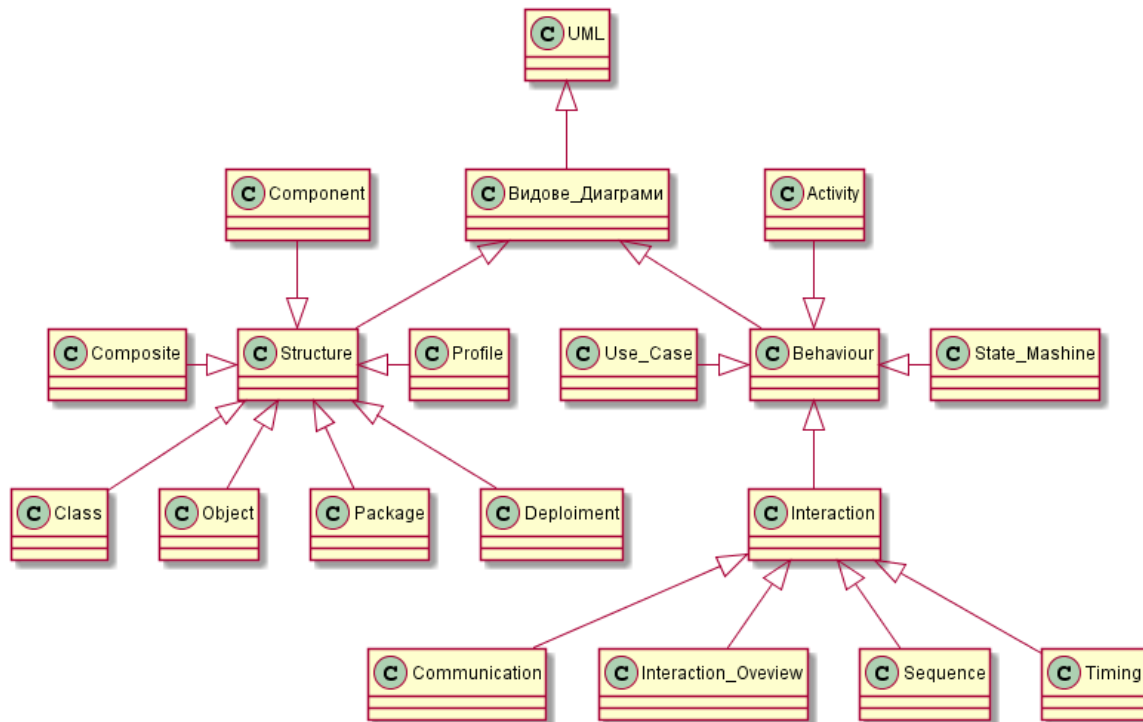
- Classdiagram (класова диаграма)
- Componentdiagram (компонентна диаграма)
- Compositestructurediagram (диаграма на съставна структура)
- Deploymentdiagram (диаграма на разгръщане)
- Objectdiagram (обектна диаграма)
- Packagediagram (диаграма на пакетите)
- Activitydiagram (диаграма на дейност)
- State Machinediagram (диаграма на машина на състоянията)
- Usecasediagram (диаграма на типичните случаи на употреба)
- Communicationdiagram (комуникационна диаграма)
- Interactionoverviewdiagram (UML 2.0) (диаграма за преглед на взаимодействие)
- Sequencediagram (диаграма на последователност)
- UML TimingDiagram (UML 2.0) (времева диаграма)
- UML ProfileDiagram (профилна диаграма)

UML предоставя множество възможности за дизайн, моделиране и визуализация на диаграми. Това включва визуализация на архитектурата на системи, техните компоненти, връзките между тях, дейности, взаимодействия между компоненти и много други. Създадени са софтуерни продукти, които на база на UML предоставят възможност за автоматизация на процесите по създаване на различни системи, визуално програмиране, работа с бази данни и др. Предоставени са методи за визуално генериране на документация и представянето им в организирана класификационна форма. При по-задълбочено изследване може да се забележи, че UML покрива възможностите за визуализация на по-голямата част от парадигмите за програмиране. Онлайн ресурсите предлагат голям обем структурирана информация, където може да се направи подробна справка за UML [2].

Въпреки че UML диаграмите по-често се свързват с дизайна и моделирането на компютърно-ориентирани системи, не съществуват пречки за прилагането им в процеса на обучение, а също така и при изготвяне на лекционни курсове и материали за семинарни занятия. Една от причините за непознаването на тези възможности за приложение от преподаватели, несвързани с ИТ сектора, е непознаването на удобен неспециализиран софтуер, свързан с такова приложение. За генерираната Диагр 1, по-долу ще бъде използван един друг подход към създаване на UML визуализация, базиран на свободния софтуер PlantUML, в средата на Microsoft Office продукта MSWord. Това решение разширява възможностите на познатия на повечето преподаватели офис продукт, но има едно съществено различие спрямо повечето съществуващи решения – диаграмите се създават динамично на база на краткото им структурирано описание в текстовия редактор (Фиг.1).



Фиг. 1



UML Клас диаграма, визуализираща UML диаграмната йерархия. За генерация е използван plantUML средата на MSWord

Диagr. 1

Така изглежда текстовото описание, генериращо UML Диагр. 1:

```

UML<|-- Видове_Дијаграми
Видове_Дијаграми<|-- Behaviour
Видове_Дијаграми<|-- Structure
Behaviour<|--up Activity
Behaviour<|-- Interaction
Behaviour<|--right State_Mashine
Behaviour<|--left Use_Case
Structure <|-- Class
Structure <|--up Component
Structure <|-- Object
Structure <|-- Package
Structure <|--left Composite
Structure <|-- Deployment
Structure <|--right Profile
Interaction <|-- Communication
Interaction <|-- Interaction_Oveview
Interaction <|-- Sequence
Interaction <|-- Timing
    
```

На база на достъпно и логично представено текстово описание се генерира изображение, визуализиращо текстовата структура. За разлика от многото съществуващи инструменти за графична визуализация, като вградените в MSWord, преподавателят не е ангажиран с допълнително разчертаване и подредба на графични елементи по екрана. По този начин той остава концентриран върху основната идея на текущата си работа. За автори, предпочитачи обектно-ориентирания подход на описание, PlantUML предоставя алтернативни възможности за класово-йерархично или комбинирано описание на диаграмата. В следващия пример се използва този подход.

| | |
|--|----------------------|
| <p>Текстовото описание за генерация</p> <pre> Геом_фигура : +Вид Геом_фигура : +Цвят class Многоъгълник extends Геом_фигура { +върхове +ъгли +страни } class Триъгълник extends Многоъгълник { +брой страни = 3 } class Правоъгълник extends Многоъгълник { +брой страни = 4 +вид на ъгъла = прав } class Квадрат extends Многоъгълник { +брой страни = 4 +вид на ъгъла = прав +равна дължина на страните } class Кръг extends Геом_фигура { +Радиус } </pre> | <p>UML диаграма:</p> |
|--|----------------------|

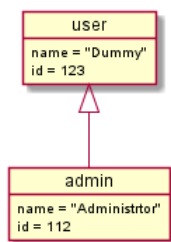
PlantUml поддържа следните диаграми:

| Вид UML диаграма | Примерна визуализация |
|--|-----------------------|
| <p>Sequencediagram, (диаграма на последователност)</p> <p>Показва как обектите взаимодействат един с друг. Отражава и последователността на тези взаимодействия</p> | |
| <p>Usecasediagram (диаграма на типичните случаи на употреба)</p> <p>Използва се за представяне на потребителското взаимодействие със системата като показва различни типични случаи на употреба</p> | |

| | |
|--|--|
| <p>Classdiagram (Клас диаграма)</p> <p>Клас диаграмата се използва при обектно-ориентираното моделиране. Описва статическа структура на класовете, техните атрибути и методи, както и взаимовръзките между тях. Тази диаграма се използва също и при моделирането на данни.</p> | <pre> classDiagram class AbstractCollection class AbstractList class Collection class List class ArrayList class TimeUnit class SuppressWarnings AbstractCollection < -- Collection AbstractList < -- List AbstractList < -- ArrayList Collection < -- List ArrayList ..> AbstractList : Object[] elementData ArrayList ..> AbstractList : size() TimeUnit ..> TimeUnit : DAYS, HOURS, MINUTES SuppressWarnings ..> SuppressWarnings </pre> |
| <p>Activitydiagram (Диаграма на дейност)</p> <p>Представя графично работни процеси и постъпково изпълняване на дейности. Дава възможност за описанието на изчислителни и организационни процеси.</p> | <pre> graph TD Start(()) --> Test{Some Test} Test -- true --> Act1(activity 1) Act1 --> D1{ } D1 -- Other test --> Act3(activity 3) D1 --> Act5(activity 5) D1 --> Act6(activity 6) Act3 --> D2{ } D2 -- last test --> Act8(activity 8) D2 --> Act7(activity 7) Test -- false --> Act2(activity 2) </pre> |
| <p>Componentdiagram (Компонентна диаграма)</p> <p>Визуализира взаимовръзките между компонентите на една система. Може да бъде използвана и за представянето на структурата на сложна система.</p> | <pre> classDiagram class SomeGroup class FirstComponent class AnotherComponent class OtherGroups class SecondComponent class Example1 class MySQL class ThisIsMyFolder class Folder3 class Frame4 SomeGroup -- FirstComponent SomeGroup -- AnotherComponent FirstComponent ..> AnotherComponent : HTTP OtherGroups -- SecondComponent SecondComponent ..> OtherGroups : FTP Example1 ..> Example1 MySQL -- ThisIsMyFolder ThisIsMyFolder ..> Folder3 Folder3 ..> Frame4 </pre> |
| <p>State diagram (Диаграма на състоянието)</p> <p>Използва се в компютърните науки и свързаните с тях области за визуализация на поведението на системи. Диаграмата е абстрактното описание на поведението на системата посредством краен брой състояния.</p> | <pre> stateDiagram-v2 [*] --> NotShootingState state NotShootingState state IdleMode as Idle mode state ConfiguringMode as Configuring mode NotShootingState --> IdleMode NotShootingState --> ConfiguringMode IdleMode --> ConfiguringMode : EvConfig ConfiguringMode --> IdleMode : EvConfig </pre> |

Objectdiagram (Обектна диаграма)

Използва се за описание на пълното или частично състояние на системата в определено време



Макар и създадени за целите на ИТ индустрията, UML диаграмите могат да бъдат използвани и в други области, където е необходимо визуалното представяне на йерархии, класове и обекти, връзките между тях, взаимовръзките между процеси и др.

4.1. Mind Map

Мисловните карти визуализират диаграми, които представят мисли, цели, задачи, идеи, които са свързани с определена главна идея. Могат да помогнат при структуриране на информацията, нагледно представяне на идеи, да бъдат използвани като помощно средство в обучението, при решаването на определен проблем и др. Базираны са около една централна идея, от която се разклоняват асоциирани с нея идеи и категории. Всяка конкретна идея се представя посредством ключова дума и визуални елементи (изображения, цветове, схеми, форми). Отделните категории представляват семантични връзки между части от информацията.

Произходът на мисловната карта е свързан с работата на британския психолог Тони Бюзан, който изследва методи, спомагащи за по-бързо запаметяване и усвояване на знания. Тези изследвания започват още през 60-те години на 20-ти век, а понастоящем са софтуерно реализирани в авторския му продукт iMindMap(Фиг. 2):



Фиг. 2

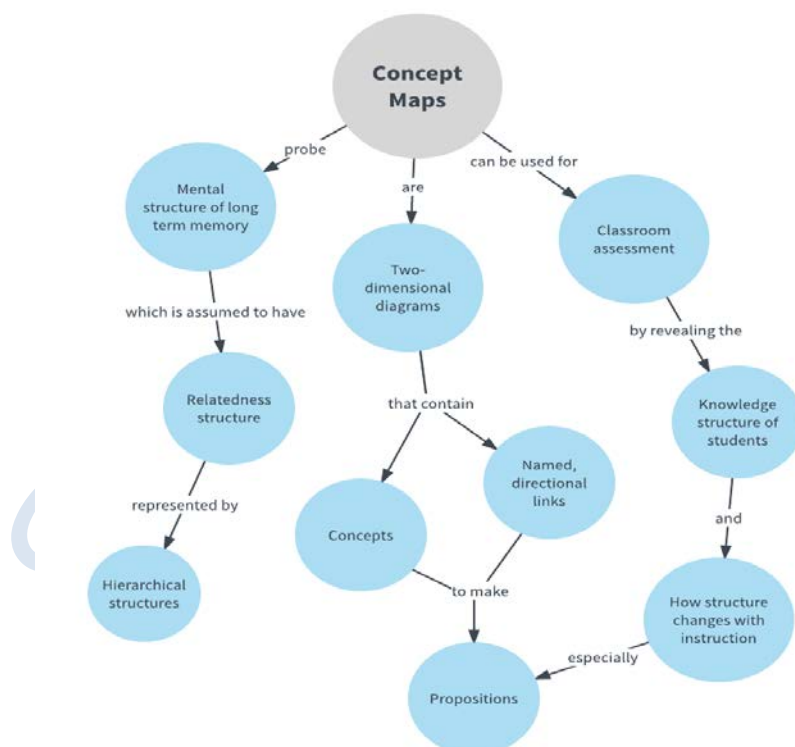
Съществуват множество Mind Map софтуерни реализации както за локално десктоп приложение, така и под формата на уеб услуга:

| Mind Map приложение | Платформа | лиценз |
|---------------------|-------------|-----------------------|
| iMindMap | Десктоп | комерсиален |
| FreeMind | Десктоп | отворен код |
| XMind | Десктоп | комерсиален, свободен |
| MindMup | Уеб, онлайн | свободен, комерсиален |
| WiseMapping | Уеб, онлайн | отворен код |
| MindMapFree | онлайн | Свободен |

Множество автори и фирми предлагат Mind Map курсове за обучение, целта на което е повишаване на ефективността при използването на тези решения. Погледнато през очите на потребителя, създаването на такива карти не изисква специализирани умения. Погледнато през очите на автор на подобен софтуер, цялата информация, онагледяваща MindMap връзките, се преобразува в структурирана йерархия от данни, които могат да бъдат обработвани в програмата и запазени на запаметяващо устройство. Това открива нови възможности като преобразуването на съществуваща вече структурирана текстова информация в Mind Map визуализация. Посредством този метод би било лесно визуализиране в MindMap карта например съдържанието на изготвен вече лекционен курс. Създателите на Mind Map приложения работят по прилагането и на тази функционалност. Комерсиалната версия на XMind, а също така и уеб услугата text2mind.com могат да послужат за запознаването с тази технологична възможност. Макар и не толкова разпространено, това решение е известно като Text To Mind. Както споменахме в началото, при провеждане на входящи тестове най-често получаваме количествена оценка на знанията. Съчетаването на една система за тест с Text To Mind може да осигури поглед върху структурата, генерирана от отговорите на обучавания.

4.1. Concept Map

Концептуалната карта е диаграма, която визуализира връзките между понятията. За разлика от мисловната карта, където връзките между елементите имат дървовидна структура и са центрирани около една главна идея, то концептуалната схема се разглежда като семантична връзка от взаимосвързани понятия по определени правила. Докато мисловната карта се разглежда като асоциативна карта, концептуалната карта може да бъде разглеждана като семантична мрежа. Описаните досега UML карти се явяват подмножество на концептуалните карти.



References:

1. Radev, P., Tradicii I savremennost v obrazovaniето, 2015

2. Object Management Group, OMG Unified Modeling Language TM (OMG UML),
<http://www.omg.org/cgi-bin/doc?formal/15-03-01.pdf>
3. PlantUml Language reference guide,
http://plantuml.com/PlantUML_Language_Reference_Guide.pdf

*Ivaylo Burov, Ivanka Burova
Department of Technological and Vocational Education,
Preschool and Primary School Education
At Konstantin Preslavsky – Univesrity of Shumen*

SOCIOBRAINS