

## **ПРИНЦИПИ ПРИ ИЗГРАЖДАНЕ НА ИНТЕЛИГЕНТЕН SCORM РЕДАКТОР**

**Дамян Митев**

***Резюме.** В тази статия са разгледани принципите, залегнали в основата на изграждането на средата за създаване и редактиране на SCORM съвместимо електронно съдържание Selbo 2. Дискутирано е как тези принципи са повлияли на архитектурата на средата и как са се отразили на отделните компоненти на системата.*

### **1. ВЪВЕДЕНИЕ**

Selbo 2 [2] е среда за разработване на SCORM 2004 [1] съвместимо електронно съдържание, което да се използва за нуждите на проекта Distributed eLearning Center (DeLC) [4] във Факултета по математика и информатика на Пловдивски университет „Паисий Хилендарски“. Целта на проекта е да се създаде гъвкава и адаптивна инфраструктура, която ефективно да поддържа система за персонализирано електронно и дистанционно обучение [5, 6].

В основата на разработката на тази инфраструктура е залегнал стандарта SCORM 2004. SCORM е референтен модел, който дефинира принципите на изграждане, структуриране, интерпретиране и визуализация на електронно съдържание, като позволява многократно използване на съществуващи материали, голяма гъвкавост при създаване на нови електронни уроци и провеждане на персонализиран (от гледна точка на обучаващият се) обучителен процес.

Една от задачите на проекта DeLC е създаването и поддържането на материали по дисциплината Софтуерни технологии [3]. За създаването на тези материали сме използвали различни съществуващи SCORM редактори – SCORM Editor на Reload [8] и Trident [9]. С опознавателна цел сме разглеждали и други SCORM съвместими средства [10].

Опитът ни показва, че използването на SCORM редактори с общо предназначение е трудно, дори за специалисти, добре запознати със стандарта. В резултат на това се роди и идеята да разработим среда, която да подпомага създателя на електронно съдържание (в частност при създаването на материали по Софтуерни технологии) и да крие спецификите на референтния модел като позволява работа чрез използване на интуитивни команди.

В тази статия са представени принципите и идеите, които стоят зад реализацията на средата и е дискутирано как те влияят върху архитектурата на приложението и разработката на отделните компоненти на системата.

## **2. ПРИНЦИПИ ПРИ РАЗРАБОТКАТА НА СРЕДАТА SELBO 2**

Принципите са идеи, възгледи и цели, които се прилагат при реализирането на даден проект. Те обуславят функционалността на системата и начина на работа на потребителите с нея.

Няколко основни принципа са залегнали при изграждането на средата Selbo 2. Те са определящи не само за архитектурата на средата, но и за имплементацията на отделните компоненти в нея:

- цялостна среда за създаване и редактиране на SCORM съвместимо електронно съдържание;
- поддръжка на педагогически, домейн и потребителски модели;
- интелигентно поведение на средата – проактивност, адаптивност, абстракция над техническата реализация на SCORM;
- модулност – лесно добавяне на нови компоненти към средата;
- интуитивна работа със средата – стандартен и лесен за използване потребителски интерфейс.

## **3. ЦЯЛОСТНА СРЕДА ЗА SCORM СЪДЪРЖАНИЕ**

В SCORM има много ясно разграничение между електронните ресурси и организацията на тези ресурси (със съответните правила за навигация между тях) за постигане на конкретна педагогическа цел. За ресурс може да се използва всяка електронна единица, която носи информация на обучаемия. Примери за ресурси са текстови файлове, изображения, презентации и колекция от HTML страници.

В класическото електронно обучение често една учебна единица (урок) се състои само от един вид ресурс. Например, една лекция е записана в една презентация или в един документ. На всички обучаеми се предлага едно и също съдържание, което е статично и с предварително дефинирана организация.

Според SCORM учебната единица се състои от множество атомарни използвани ресурси, които имат завършен смисъл. Такъв пример е лекция, която се състои от няколко глави, като всяка глава представлява отделен документ. В специален манифестен файл се задава логическата структура на урока като конкретна йерархична организация на ресурсите. Тази йерархична структура представлява дърво, чиято дълбочина и смисъл на отделните възли се дефинират от автора на съдържанието. Най-голямото предимство на SCORM е това, че позволява дефиниране на правилата за последователност и навигация между възлите в дървото. Използвайки тези правила и прогреса на

всеки обучаващ се при използването на материала, обучителната система може да предостави персонализирано взаимодействие с потребителите си. Резултатната SCORM обучителна единица е пакет, съдържащ манифестния файл и всички ресурси, които той референцира.

Друго предимство на стандарта е възможността за дефиниране на интерактивни ресурси и разграничаването между пасивно и интерактивно съдържание. Пасивните ресурси не комуникират с обучителната система, след като бъдат предоставени на обучаемия. Пример за такива ресурси са изображение, текстови документи и статични HTML страници. Интерактивните ресурси, от своя страна, могат да комуникират с обучителната система по време на изпълнението си. По този начин те могат да следят много фино прогреса на обучаемия и да предоставят на системата сложни структури от данни, които след това да бъдат използвани в правилата за последователност и навигация. Това прави възможно изграждането на тестове.

SCORM се фокусира само върху съдържанието на манифестният файл, като създаването, редактирането и намирането на източници с информация за създаването на индивидуалните ресурси са извън неговата сфера. Задача на създателя на съдържанието е да намери готови преизползваеми ресурси или сам да създаде такива, използвайки различен софтуер за редактиране на различните избрани от него файлови формати и намиране на информация за включване в тези ресурси. Същият път следват и съществуващите SCORM съвместими редактори. Те предоставят средства единствено за създаване и редактиране на възлите на манифестния файл и техните стойности. Често тези редактори не проверяват за коректност или консистентност на въведените данни, което позволява запис на грешен или безсмислен манифестен файл.

Selbo 2 се стреми да затвори цикъла на създаването на електронно съдържание, като предоставя средства за манипулиране на организацията на ресурсите и SCORM правилата за последователност и навигация (SCORM sequencing and navigation), позволява редактирането на самите ресурси чрез набор от вградени в средата редактори и съдържа материали от различни дисциплини, които могат да се използват при създаване или редактиране на ресурси.

### **3.1. МАНИПУЛИРАНЕ НА ДЪРВОТО НА СЪДЪРЖАНИЕТО**

В архитектурата на Selbo2 организацията на ресурсите се реализира чрез дърво на съдържанието. Това дърво определя структурата на електронния урок. Компонентът, който визуализира и манипулира дървото, е ресурсният организатор (resource organizer).

### **3.2. РЕДАКТОРИ НА РЕСУРСИ**

Компонентите, които отговарят за манипулирането на ресурсите в Selbo 2, са редакторите на съдържание (content editors). Средата предвижда

включването на различни редактори за различни типове файлови формати на ресурсите. SCORM, в ролята си на референтен модел, не задава ограничения относно това какви типове ресурси могат да се използват. Разработваната от нас система за обучение, както и повечето популярни продукти от този тип, използва Web среда за предоставяне на учебните материали на обучаващите се. Поради тази причина ресурсите трябва да могат да се визуализират в стандартен Web browser. Това наложи основният редактор на съдържание в Selbo 2 да бъде HTML редактор и основният ресурс, поддържан от средата, да бъдат стандартни HTML страници.

### **3.3. ОНТОЛОГИИ**

В Selbo 2 са включени онтологии за различни дисциплини. Онтологията [18] представлява формално описание на знания като йерархия от концепции в определена област, описвайки характеристиките и взаимовръзките между тези концепции. Онтологиите [7], използвани в средата Selbo 2, съдържат в себе си понятия, термини, концепции и примери. Всеки елемент от онтологията съдържа семантични връзки към други елементи, както и описание, което може да бъде изцяло или частично използвано от автора при създаване на материалите.

Онтологиите не са хранилища с готови преизползваеми ресурси. Те трябва да се разглеждат по-скоро като речници, които описват конкретна дисциплина и предоставят обзор на използваните в нея ключови понятия.

### **3.4. ВИЗУАЛИЗАТОР НА ГЕНЕРИРАНИЯ ПАКЕТ**

Средата също така съдържа средство за визуализиране на готовия SCORM пакет, което улеснява тестването и откриването на грешки и несъответствия между желаното поведение на пакета и зададените правила за последователност и навигация.

## **4. ПОДДРЪЖКА НА ДОМЕЙН, ПЕДАГОГИЧЕСКИ И ПОТРЕБИТЕЛСКИ МОДЕЛИ**

Вторият принцип е поддръжката на педагогически, домейн и потребителски модели. Моделите са набор от специализирани знания за изучаваната дисциплина (домейн модел), знания за използвания педагогически подход (педагогически модел) и знания за индивидуалните предпочитания на учениците и учителите (потребителски модел). Тези знания се използват като насоки в дизайна или във функционирането на компонентите на средата.

### **4.1. ДОМЕЙН МОДЕЛ**

Домейн е предметната област или дисциплината, за която се разработва електронно съдържание. Домейн моделът се поддържа от всички компоненти,

които са специфични за дадена предметна област. В Selbo 2 това са специализирани редактори и вече споменатите онтологии.

Специализираните редактори работят с файлови формати, които са специфични за дадена предметна област или близки една до друга области. Специализиран редактор може да е отделен компонент, който редактира или генерира конкретен файлов формат. Пример за такъв специализиран редактор за UML диаграми, който генерира изображения и намира приложение при изготвяне на електронно съдържание по Софтуерни технологии. Специализираните редактори могат да бъдат и plug-in компоненти, които предоставят разширени или допълнителни функционалности към съществуващи редактори. Такъв пример е редактор на математически формули, който разширява функционалността на стандартния HTML редактор и позволява вграждането на математически символи и формули в стандартна HTML страница. Този редактор има приложение основно в обучението по математика.

## 4.2. ПЕДАГОГИЧЕСКИ МОДЕЛ

Педагогическият модел включва в себе си елементите от системата, отговарящи за задаване на педагогически цели, които трябва да бъдат удовлетворени от обучаемия при използване на електронната лекция. Педагогическите цели са набор от логически правила, които обучителната система спазва при взаимодействието си с обучаемите.

Пример за педагогическа цел е обучаващият се да усвои поне 80% от нов материал. За дефинирането на тази цел е необходим пакет с два главни възела – първият, съдържащ материала с нови знания, вторият – тест, който проверява тези знания. Логическите правила биха били задължителен прочит на материала, след това задължително държане на теста и ако резултата от него е по-малък от 80%, то на обучаемият не се позволява да продължи напред, а задължително се връща за повторен прочит.

В класическото обучение педагогическите правила са вплетени в самите обучителни ресурси и са интегрална и неотделима част от тях. В Selbo 2 педагогическият модел е изведен като основна принципна единица с явна поддръжка от компонентите на средата.

Педагогическият модел се поддържа от ресурсния организатор, който е и основния компонент в системата. Чрез него се манипулира дървото на SCORM съдържанието – създаване, премахване, местене и редактиране на възли в дървото, задаване на правила за последователност и навигация между възлите и свързване на крайните възли с конкретни ресурси. Върху това дърво се дефинират педагогическите цели, които авторът иска да бъдат удовлетворени. Логическите правила, дефинирани от целите, се записват като конкретни правила за SCORM последователност и навигация (Sequencing and Navigation), като едно логическо правило често се разбива на няколко SN правила, зададени върху различни възли.

Създателят на материалите решава по какъв начин да групира ресурсите, колко дълбока да е йерархичната структура на организацията и какъв смисъл имат различните нива. За улесняване на процеса по автоматична обработка на дървото на съдържанието са разработени специални онтологии, които описват йерархичната структура и формално задават имена и смисъл на всяко ниво от елементи. Тези онтологии се използват от абстрактните правила, описани в следващата точка. Такава онтология може да дефинира корена на дървото като урок, второто ниво като точка и третото – като подточка и да забрани създаване на елементи на четвърто ниво.

Друг елемент от средата, поддържащ този модел, са педагогическите шаблони. Целта им е да улеснят създаването на ново учебно съдържание. Шаблонът отговаря за конкретна педагогическа цел, като съдържа предварително дефинирана структура на съдържанието заедно с правилата за удовлетворяване на тази цел. Шаблоните се прилагат върху ресурсния организатор при създаване на нов учебен материал.

### **4.3. ПОТРЕБИТЕЛСКИ МОДЕЛ**

Потребителският модел е свързан с персонализацията на системата към своите потребители и средствата, които ги подпомагат при работата им с нея. В тази група се включват всички визуализатори и редактори, които използва потребителя.

Средата Selbo 2 позволява създаването на различни потребителски профили. Профилът съдържа личните предпочитания на конкретен потребител. Такива примери са позициониране на различните редактори, избор на предпочитан редактор за конкретен вид ресурс (ако в системата са инсталирани няколко), списък с последно използвани онтологии и избор на основен домейн. Този домейн се поддържа и от интелигентните помощници в средата – агенти и магьосници (wizards). Те са дискутирани в следващата точка.

## **5. ИНТЕЛИГЕНТНО ПОВЕДЕНИЕ НА СРЕДАТА**

В контекста на средата Selbo 2 интелигентността е термин, обхващащ поведението на множество компоненти, което цели да автоматизира или улесни автора при създаването на образователните единици.

### **5.1. ПРОВЕРКИ ЗА КОНСИСТЕНТНОСТ НА ДАННИТЕ**

В основата си Selbo 2 е редактор на SCORM съвместимо съдържание и предлага средства за директна манипулация на структурите от данни на дървото на съдържанието. Компонентите, които предоставят интерфейса за визуализиране и промяна на тези данни, също така извършват и проверка за

консистентност и смисленост на въведените стойност. Например, средата няма да позволи използване на objective в правилата за натрупване, ако то не е било дефинирано в секцията с валидни objectives. По подобен начин няма да е възможно задаване на стойности в два взаимноизключващи се възела.

## 5.2. АБСТРАКЦИЯ НАД SCORM

Често за постигането на елементарен ефект (например, задължително препрочитане на материал, ако теста след него е издържан с по-малко от 80%) трябва да се използват множество SCORM правила, разпръснати между няколко възела в организацията на материала. При създаване на по-сложен учебен курс броят на тези правила значително нараства и задаването им се превръща в трудоемка задача. Selbo 2 се стреми да предостави ниво на абстракция над заложените в SCORM правила за последователност и навигация. На това ниво авторите на съдържание могат да задават интуитивни или лесни за разбиране правила, които задават педагогически цели в урока. Задаването и декомпозирането на тези правила се извършва от ресурсния организатор.

Абстракцията над SCORM се допълва от опростен модел на дървото на съдържанието, в който средата автоматично управлява метаданните на възлите от организацията. Средата се грижи и за безпроблемното свързване на елементи от дървото на съдържанието с ресурси и автоматичното описване и управление на ресурсните дефиниции в манифестния файл. По този начин се размива границата между ресурсите и организацията им и прави създаването на електронно съдържание по-интуитивно.

Интерактивните преизползваеми ресурси, дефинирани в термините на SCORM, позволяват обратна връзка от ресурса към обучителната система по време на изпълнението си. Данните, които се предават, са множество от променливи, наречени objectives. На базата на тези променливи могат да се зададат правила за последователност и навигация. Абстракцията, поддържана от ресурсния организатор, позволява тези променливи да се дефинират на високо логическо ниво като параметри на педагогическите правила.

## 5.3. ПОМОЩНИЦИ

Интелигентното поведение на Selbo 2 се допълва от специализираните помощници – агенти и магьосници (wizards). Помощниците играят важна роля в архитектурата на средата, защото комуникират с много от компонентите на средата, за да улеснят създателя на електронното съдържание.

Магьосниците са пасивни помощници. Те се извикват изрично от потребителя и жизненият им цикъл е кратък. Тяхната роля е да предоставят интуитивен интерфейс и да предлагат добри практики при извършване на специфични задачи. Пример за такава задача е създаване на нов SCORM

пакет – магьосникът позволява избор на дисциплина (домейн) и на педагогически шаблон, като автоматично създава начално дърво на съдържанието с копирани от шаблона педагогически правила и зарежда съответните за домейна онтологии. Магьосникът отговаря и за добавяне на нови педагогически правила към дървото на съдържанието, като следи за неговата валидност и за конфликти с целите в избрания шаблон или други ръчно добавени правила. Магьосниците може да са и домейн-зависими компоненти, които да са налични само при работа с конкретна дисциплина.

Агентите са непрекъснато активни помощници. Те работят автономно, като взаимодействат с компоненти и други агенти в средата. В резултат на това взаимодействие и техните зададени цели те могат да вземат решения и извършват действия, отнасящи се към работата на потребителя [15, 16]. За целите на средата, агент е софтуерна единица, която има собствен механизъм на контрол, може да проявява различни поведения и може да взаимодейства с други агенти и компоненти от средата [13, 14]. Типично агентите са свързани с един конкретен компонент от средата и следят взаимодействието на потребителя с него. Тази комбинация от компонент и агент се нарича интелигентен компонент [17, 19, 20]. По време на това взаимодействие те могат да извършват корекции, да дават предложение или да предупреждават за евентуални грешки или лоши практики. Един пример е агент, проверяващ за правописни грешки при използване на текстов редактор. Друг пример е агент, който следи за ключови думи в текста и ги търси като термини в заредената онтология. Това може да подскаже на създателя на съдържанието да включи допълнителни материали в учебното съдържание.

## **6. МОДУЛНОСТ И АДАПТИВНОСТ**

Принципите за адаптивност и модулност са много близки един на друг, като различията между тях са логическото ниво, на което са дефинирани.

Модулността се определя на техническото ниво на изграждане на системата. Тя дефинира наличието на plug-in система в средата [12], чрез която да могат лесно да се добавят нови компоненти, като визуализатори, редактори, магьосници, агенти, онтологии, шаблони и списъци с библиотеки от предварително подготвени ресурси.

Адаптивността разглежда логическата структура на средата и функционалността на елементите ѝ. За да бъде системата адаптивна, тя трябва да може лесно да дефинира нов домейн и да добавя нужните за домейна редактори, онтологии, помощници и педагогически шаблонни. Също да може да променя съществуващи шаблони и да може да добавя нови педагогически модели. Не на последно място средата трябва да може да се адаптира към персоналните предпочитания на потребителя, което свойство се покрива от потребителския модел, заложен в архитектурата на системата.



## 7. ИНТУИТИВНА РАБОТА СЪС СРЕДАТА

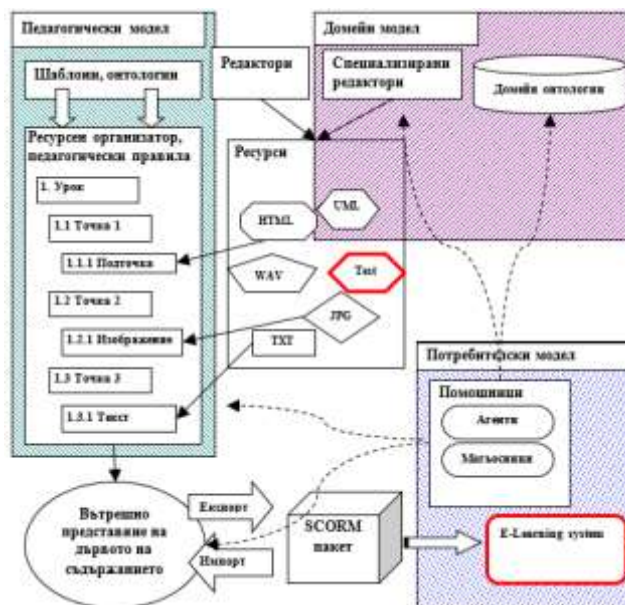
Принципа за лесна и интуитивна работа със средата оказва влияние на всички слоеве от имплементацията ѝ.

Системата предоставя стандартен и естетически издържан графичен потребителски интерфейс, удобно разположение на компонентите и възможност за тяхното местене и оразмеряване. Също така реализира адекватна и атрактивна визуализация на помощниците.

Абстракцията над спецификата на SCORM и използването на помощници допълнително улеснява потребителя и му помага да се фокусира върху логиката на създаване на материала, като скрива тривиалните детайли по реализацията на желаното поведение.

## 8. АРХИТЕКТУРА НА SELBO2

Архитектурата на средата Selbo 2 е показана на Фигура 1. Моделите, които се реализират от различните компоненти, са маркирани в различни щрихи.



Фигура 1. Архитектура на Selbo 2

### 8.1. РЕСУРСЕН ОРГАНИЗАТОР

Ресурсният организатор манипулира дървото на SCORM съдържанието. Той групира отделните ресурси в структура, представляваща една образователна единица, и позволява създаване, премахване, местене и редактиране на възли

в дървото. Чрез него се задават правилата за последователност и навигация между възлите и върху него се дефинират педагогическите цели.

## **8.2. ПЕДАГОГИЧЕСКИ ШАБЛОНИ**

Шаблоните служат за скелет при изграждане на структурата на ново учебно съдържание. Шаблонът съдържа конкретни педагогическа цели, началната структура и организацията на структурата на съдържанието. Шаблоните се прилагат върху ресурсния организатор при създаване на нов учебен материал.

## **8.3. РЕДАКТОРИ НА РЕСУРСИ**

Чрез редакторите се създават и манипулират същинските ресурси, които съставят учебния материал. Ресурс е всеки файл, който може да бъде визуализиран на обучаемите. Модерните обучителни системи работят в Web среда, затова и основния файлов формат, поддържан от средата, е HTML. Поради това главният редактор е редакторът на HTML страници.

Редакторите могат да бъдат разделени на две групи, в зависимост от своята домейн-ориентираност. Първата група се състои от редактори на стандартни формати, които не зависят от предметната област. Във втората група попадат редакторите, които манипулират специфични за конкретен домейн файлови формати. Те, от своя страна, могат да бъдат разделени на 2 типа – самостоятелни редактори или plug-ins към съществуващи, като разширяват техните възможности.

В Selbo 2 редакторите са реализирани като интелигентни компоненти. Интелигентните компоненти са комбинация между компонент на графичния интерфейс и специализиран агент, създаден да взаимодейства с компонента.

## **8.4. ДОМЕЙН ОНТОЛОГИИ**

В Selbo 2 се използват специално създадени онтологии, всяка от които съдържа формално описание на знания от определена дисциплина. Термините, концепциите и понятията, съставляващи онтологията, съдържат в себе си детайлни описания както и семантични взаимовръзки между елементите. Описанията могат да се използват от автора на съдържание като насоки или като готов материал за включване в електронните лекции.

Онтологиите използват една и съща вътрешна структура и дефинират единен набор от характеристики, които могат да бъдат задавани за всеки възел. Това прави възможно създаване на един специален компонент – визуализатор на онтологии, чрез който потребителят да достъпва съдържанието на всички домейн-зависими онтологии. Това способства и за създаване на агенти, които да търсят нужната информация в зависимост от избраните педагогически цели на урока.

## 8.5. ПОМОЩНИЦИ

Помощниците са софтуерни компоненти, чиято цел е да улеснят автора при създаване на съдържанието. Тяхната роля е да автоматизират техническите процеси при създаване на SCORM съвместим пакет, да проверяват за технически и доколкото е възможно за смислови грешки, да предлагат най-добри практики при работа със системата и да търсят информация в зависимост от действията на потребителя. Това се постига чрез колаборация между помощника и няколко компонента на системата чрез предварително дефиниран протокол за комуникация.

Selbo 2 използва два типа помощници – магьосници и агенти. Магьосниците са пасивни помощници, стартират се изрично от потребителя. Те се използват основно при създаване на нови елементи от дървото на съдържанието, за да инициализират структурите на SCORM манифеста според предварително избран педагогически шаблон.

Агентите са постоянно активни помощници. Те оперират автономно, като комуникират с други компоненти и агенти или следят действията на потребителя. В резултат на това взаимодействие и поставените цели, агентът може да взема решения и да извършва действия, свързани с работата на потребителя и контекста, в който оперира. Всеки агент е разработен съобразно спецификата на задачите, които изпълнява, и компонентите, с които взаимодейства. За улеснение на това взаимодействие и комуникацията между агентите се използва вътрешен протокол, надграждащ ACL [21].

## 9. ЗАКЛЮЧЕНИЕ

Целта на проекта Selbo 2 е да създаде цялостна среда за създаване и редактиране на SCORM 2004 съвместимо електронно съдържание, чрез която авторите да могат да дефинират не само структурата на учебния материал, а и да редактират самото му съдържание. При имплементацията е залегнала идеята за модулност, която позволява лесно да се добавят нови компоненти в системата. Това способства за дефиниране на домейн-зависими компоненти – инструменти, специализирани за създаване на съдържание за определена дисциплина – редактори на специализирани файлови формати и онтологии, описващи домейна. Друга основна концепция е явното поддържане на педагогически модел като множество от структури, които описват педагогическите цели на материала. Тези цели се описват чрез педагогически правила. Средата се стреми да улесни работата на потребителя като предоставя абстракция над техническите детайли на SCORM и предлага лесен и интуитивен интерфейс за изграждане на структурата на материала. Важна особеност е и използването на интелигентни помощници, които подпомагат потребителя по време на създаване на електронните материали.

## ЛИТЕРАТУРА

- [1] Sharable Content Object Reference Model, URL <http://www.adlnet.gov/scorm/>
- [2] Mitev, D., S. Stoyanov and I. Popchev, Selbo 2 – an Environment for Creating of Electronic Content in Software Engineering, *CIT*, Vol. 9, No. 3, Sofia, 2009, 96–105.
- [3] Software Engineering: Computer Science Education and Research Cooperation, URL <http://www2.informatik.hu-berlin.de/swt/intkoop/daad/>
- [4] Stojanov, S., I. Ganchev, I. Popchev, M. O’Droma and R. Venkov, DeLC – Distributed eLearning Center, *Proc. of the 1<sup>st</sup> Balkan Conference in Informatics BCI’2003*, Thessaloniki, Greece, 21–23 November, 2003, 327–336, ISBN: 960-287-045-1.
- [5] Stojanov, S., M. O’Droma and I. Ganchev, A Model for Integration of Electronic Services into a Distributed eLearning Center, *Proc. CD of 14<sup>th</sup> EAEEIE International Conference*, A17, 7 pages, June, 2003, Gdansk, Poland, ISBN: 83-918622-0-8.
- [6] Stojanov, S., Virtual university – helper tool for student education, *National scientific conference “Informatics in scientific knowledge”*, 13–15.06.2002, VSU, 141–151.
- [7] Protégé ontology editor, Stanford University, URL <http://protege.stanford.edu/>
- [8] RELOAD Content Package and Metadata editor, URL <http://www.reload.ac.uk/>
- [9] Trident – SCORM IDE, URL <http://www.scormsoft.com/trident>
- [10] eXe – eLearning editor, URL <http://exelearning.org/>
- [11] Kafenio WYSWYG HTML Editor component, URL <http://editor.kafenio.org/>
- [12] Java Plugin Framework (JPF), URL <http://jpf.sourceforge.net/>
- [13] Java Agent Development platform – JADE, Telecom Italia Lab, URL <http://jade.tilab.com/>
- [14] Krzysztof Chmiel, Maciej Gawinecki, Pawel Kaczmarek, Michal Szymczak, Marcin Paprzycki, “JADE Agent Platform” (2005)
- [15] Greenwood, D., F. Bellifemine, G. Caire, *Developing Multi Agent Systems with JADE*, Apr 2007.
- [16] Bigus, J., *Constructing Intelligent Agents With Java A Programmers Guide to Smarter Applications*.
- [17] Mitev, D. and I. Minov, *Intelligent Components*, 30.11.2007, Smolyan, Bulgaria.
- [18] Brase, J. and W. Nejdl, *Ontologies and Metadata for eLearning*, S. Staab & R. Studer (Eds.) *Handbook on Ontologies*, Springer-Verlag, 2004, 555–574.
- [19] Griss, M. and G. Pour, *Accelerating Development with Agent Components*.
- [20] Briot, J., *Agents and components: can they be combined?*, SEAS’05
- [21] FIPA ACL, URL [www.fipa.org/repository/aclspecs.html](http://www.fipa.org/repository/aclspecs.html)

Факултет по математика, информатика и информационни технологии  
Пловдивски университет „Паисий Хилендарски“  
Бул „България“ 236, 4003 Пловдив, България  
[damyan\\_mitev@mail.bg](mailto:damyam_mitev@mail.bg)

## **PRINCIPLES FOR BUILDING INTELLIGENT SCORM EDITOR**

**Damyam Mitev**

***Abstract.** This paper discusses the principles that govern the creation of environment for editing of SCORM compatible electronic content – Selbo 2. It also explains how these principles affect the overall architecture of the environment and different components of the system.*

