

ИНТЕЛИГЕНТНИ КОМПОНЕНТИ В РАЗВОЙНАТА СРЕДА ЗА ЕЛЕКТРОННО ОБУЧЕНИЕ SELBO 2

докт. Дамян Митев¹, гл.ас. д-р Олга Рахнева², докт. Иван Минов¹

¹) Пловдивски университет „П. Хилендарски“, ²) УХТ - Пловдив

INTELLIGENT COMPONENTS IN DEVELOPMENT ENVIRONMENT FOR E-LEARNING SELBO 2

Damyan Mitev¹, Olga Rahneva², Ivan Minov¹

¹) University of Plovdiv, ²) UFT - Plovdiv

Abstract

Electronic education needs intelligent development environments, which not only have tools for manipulation of electronic content, but also aid developers in creating this content. Development environment Selbo 2 uses intelligent components – combination of visual component and intelligent agent – to provide necessary flexibility and adaptability of intelligent working place for electronic content developer.

1. Въведение

Selbo 2 е развойна среда за електронно обучение, която предоставя на създателите на електронни уроци интелигентно работно място за създаване на електронно съдържание. Средата разполага с различни специализирани редактори за много от популярните електронни формати, както и възможност за използване на предварително създадени ресурси от различни области, записани в специализирани онтологии. Също така към разработваните материали (електронни уроци) могат да се прилагат шаблони с педагогически правила, които да представят дидактическите цели на автора.

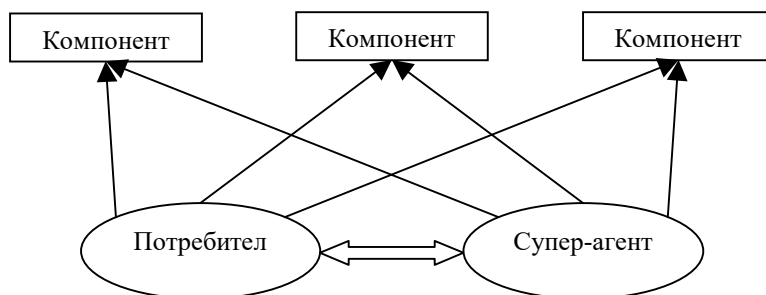
Основна характеристика на средата е наличието на персонални асистенти, които помагат на автора при създаването на електронното съдържание. Те са реализирани като интелигентни агенти-помощници, защото само агентно-ориентираната парадигма осигурява гъвкавостта и адаптивността, нужни за създаването на тези персонални асистенти.

2. Подходи за реализиране на агенти-помощници

Очертаха се три подхода за реализирането на агентите-помощници: създаване на един супер агент, който да работи с визуалния интерфейс наравно с автора; реализиране на самите компоненти на визуалния интерфейс чрез агенти; и създаване на интелигентни компоненти – комбинация от визуални компоненти и интелигентни агенти.

2.1. Подход Супер-агент

Първият подход предполага създаването на един агент (супер-агент), който да помага на автора през целия процес на създаване или редактиране на електронно съдържание (Фиг. 1).



Фиг. 1 Подход Супер-агент

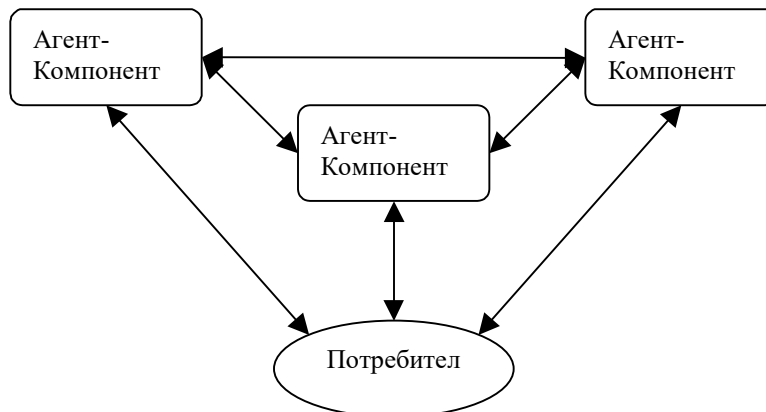
Той трябва да оказва помощ при използване на отделните специализирани редактори (като например коригира грешки), да търси подходяща информация в хранилищата от преизползваеми ресурси, която автора да използва наготово, да следи за логическата последователност на електронния урок.

Този подход е неудачен поради ред причини. На първо място този супер-агент трябва да „знае“ как да работи с различните компоненти (специализирани редактори) в системата. Освен това трябва и да може да се обучава да работи с нови компоненти, които могат да бъдат включени в средата благодарение на нейната система за приставки (plug-in subsystem). Също така този агент ще трябва да може да работи с педагогическите правила в различни домейни, в които може да бъде използвана средата. Накратко супер-агентът трябва да притежава интелигентност в много сфери.

Проблемът се изразява в това, че изискванията за интелигентност в един домейн са много различни от изискванията за интелигентност в друг[1].

2.2. Подход Агенти-компоненти

Вторият подход представлява използване на агенти като компоненти на визуалния интерфейс (Фиг. 2).



Фиг. 2 *Подход агенти-компоненти*

Така значително се намаляват изискванията за интелигентност на отделните агенти [2]. Всеки агент-компонент има ограничена функционалност и поема отговорност да помага, само когато потребителя работи с него. Освен това той има пълен контрол върху вътрешните си състояния и може да взема адекватни решения на тази база или да изменя тези състояния съобразно целите си. По този начин средата се превръща в мулти-агентна система и цялостното ѝ поведение се обуславя от колаборацията между агентите в нея.

Въпреки своите предимства, този подход има и редица недостатъци. Основният минус е смесването на две много различаващи се по идеология парадигми – компонентната и агентно-ориентираната.

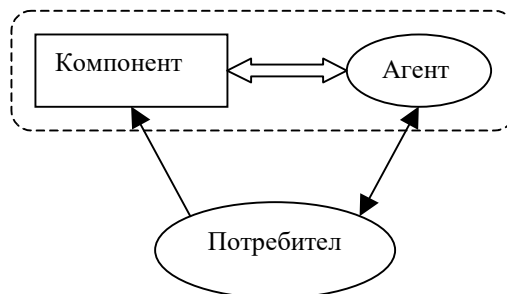
Архитектурата на компонента му позволява да бъде многократно използван в различни системи. Компонентите нямат вътрешно самоуправление, нямат ментални състояния, нямат автономност, цели и поведения. Техният жизнен цикъл се управлява от средата, в която са поставени. Функционалността им се реализира от методи, които се изпълняват веднага при извикването им, реагира на момента, най-вече на действията от страна на потребителя. Архитектура на компонента го прави идеален за преизползване във визуална среда, но неудачен за извършване на автономни действия.

Агентите, от своя страна, са проектирани за изпълнение на строго специфични задачи [3]. Те сами управляват жизнения си цикъл, могат да вземат самостоятелни решения и сами да променят вътрешните си състояния без външна намеса. Могат да преследват предварително зададени цели и да променят плановете си. За да могат да бъдат използвани като компоненти на визуалния интерфейс, към тях трябва да допълнително да се добави функционалност за обработка на визуални съобщения, за реагиране на командите на потребителя, за обработка на съобщенията на визуалната среда и взаимодействие с другите компоненти на системата. Архитектурата на агента го прави идеален за автономна работа, но е неудачна за вграждането му във визуален интерфейс.

2.3. Подход Интелигентни компоненти

Нашият подход за решаването на гореспоменатите проблеми се основава на използването на интелигентните компоненти [4]. Виждането ни за интелигентен компонент представлява комбинация от компонент на визуалния

интерфейс и интелигентен агент, при която агента може да се разглежда като контролер на компонента (Фиг. 3).



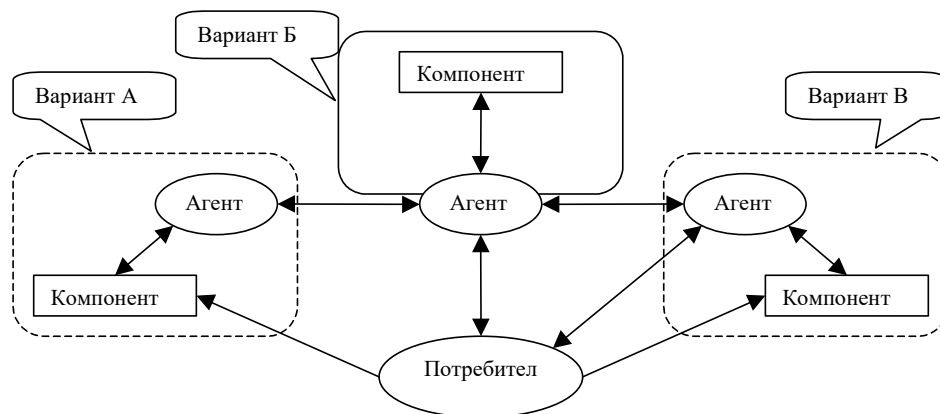
Фиг. 3 *Подход интелигентни компоненти*

Комбинацията представлява слаба свързаност между две самостоятелни единици. Компонента представлява леко изменен компонент от визуалния интерфейс, което му осигурява безпроблемна интеграция в средата. Агента, от своя страна, си запазва агентно-ориентираната архитектура и възможностите за дефиниране на поведения и цели и автономна работа. Свързаността между двата елемента се изразява в това, че те са проектирани за работят един с друг. Агента е проектиран за познава и да може да работи с интерфейса на компонента. Да следи вътрешните му състояния и да ги променя при необходимост. От своя страна компонента е предвиден да работи с агента и да му съобщава чрез специализиран протокол за промените, настъпили в резултат на действията на потребителя, евентуално искайки разрешение дали да приложи тези промени.

Едно от основните предимства на този подход, е че единият елемент от двойката може да бъде заменен лесно, без да се променя другия. Например може агент, който е специализиран за търсене на информация в онтология, може да бъде лесно заменен с друг „по-интелигентен” агент, който разполага с по-ефективни алгоритми за търсене, без да е нужно да се сменя компонента, отговарящ за визуализирането на намерената информация. Със същата лекота, в тази двойка може безпроблемно да се замени визуализатора на онтологията с друг компонент, който представя намерените данни по по-удобен, красив и интуитивен начин, без да е нужно да се сменя агента.

Друго предимство е факта, че агентите не са длъжни да познават интерфейса на „чуждите” компоненти в системата. Ако даден агент иска да направи промяна в състоянието на „чужд” компонент, той трябва само да знае как да се обърне към агента на въпросния компонент и да му предаде заявката си. От този момент другият агент има задължението да обработи тази заявка като приложи исканите промени върху компонента или я отхвърли.

Взаимодействието между компонента и агента и взаимодействието между интелигентният компонент като цяло и потребителя може да се класифицира между два крайни варианта: при единия вариант потребителя работи единствено с компонента, а при другия – единствено с агента (Фиг. 4).



Фиг. 4 Варианти на комбинация между компонент и агент

Първият краен вариант (Фиг. 4, Вариант А) предвижда агента да работи на заден план, обслужвайки компонента. Създателят на електронното съдържание вижда и работи само с компонента, променяйки вътрешните му състояния. През това време агента следи тези промени и извършва действия, в съответствие с целите си, без да изисква или да представя директно информация на потребителя. Пример за такъв вариант е комбинация от текстов редактор и агент за търсене на ключови думи. Докато потребителят пише, агента следи за изписването на ключови думи (термини) от предварително зададен домейн, и ако намери такива, ги предоставя на агент за търсене в онтология, който съответно може да намери полезна информация (примери, определения) за тези термини в онтологията, която авторът да използва наготово.

Другият краен вариант (Фиг. 4, Вариант Б) е агентът да се превърне в абсолютен контролер на компонента. В този случай всяка заявка за промяна на компонента, независимо дали идва от потребителя или от други агенти или компоненти на системата, минава през агента. Това позволява абсолютна синхронизация между двата елемента, но по идеология се приближава до вторият описан подход „агент-компонент” и се лишава от гъвкавостта на слабото свързване. Има реален смисъл само за компоненти, които не предоставят възможност за редактиране на показваните от тях данни.

В общия случай (Фиг. 4, Вариант В) агента и компонента се допълват взаимно за да предоставят на автора на електронно съдържание необходимата гъвкавост и удобство за работа. Потребителя може да работи директно с компонента или да комуникира с агента в диалогов режим. Пример за това е интелигентния компонент за дървото на съдържанието на урока. Потребителя може да използва агента за да създаде нова точка от урока, съгласно предварително зададен шаблон с педагогически правила. След това автора може да използва компонента за да промени някои от характеристиките на създадената точка, а агента следи дали тези промени отговарят на шаблона и ако не – в диалогов режим да предложи алтернативи.

Наред с изброените предимства, този подход поражда някои сериозни предизвикателства. Основното предизвикателство се състои във факта, че развойната среда се превръща в мулти-агентна система. Трябва да се разработят методи за управление на колаборацията между отделните агенти, протокол за описание на услугите, предоставяни от агентите, и механизми за достъп до тези услуги.

Интелигентните компоненти в средата Selbo 2 са предвидени да се реализират чрез използване на стандартните визуални компоненти от библиотеката SWING и агентният контейнер JADE.

JADE (Java Agent DEvelopment Framework) е софтуерна платформа разработена от TILAB, напълно реализирана на платформата Java. Той служи за разработване на разпределени мулти-агентно ориентирани приложения базирани на peer-to-peer комуникационната архитектура. JADE използва библиотеки, чрез които се разработват софтуерни агенти и осигурява run-time среда, която се грижи за управление на жизнения им цикъл и комуникацията между тях. Платформата осигурява хомогенен слой, който скрива от агентите комплексността и сложността на приложния слой (хардуер, операционна система, Java VM).

3. Заключение

За нуждите на електронното обучение са необходими интелигентни развойни среди, които не просто да предлагат средства за създаване на електронно съдържание, но също така и да помагат на авторите при създаването на това съдържание. Средата Selbo 2 използва интелигентни компоненти, комбинация на визуален компонент и интелигентен агент, за предоставяне на необходимите гъвкавост и адаптивност на интелигентно работно място за създател на електронно съдържание.

Подхода за реализирането на интелигентни компоненти като комбинация от слабо свързани визуален компонент и интелигентен агент позволяват да се създават гъвкави и интелигентни архитектури. По този начин се предоставя възможност за промяна на визуалния интерфейс на системата без да се променя функционалността ѝ, или обратно, да се повишава интелигентността чрез заменянето на агент, без да е нужно да се променя визуалния интерфейс. В същото време използването на агенти позволява задаването на разнообразни цели и поведения и улеснява поставянето не само на синтактични, но и на семантични задачи към средата.

Тази публикация е частично финансирана от проект У-ТН-402/2006 год. на Фонд „Научни изследвания” на МОН.

4. Литература

- [1] M. Griss, G. Pour, Accelerating Development with Agent Components
- [2] G. Kaminka, M. Tambe, A Synergy of Agent Components: Social Comparison for Failure Detection
- [3] An Architecture for Embedded Intelligent Components, PC AI magazine, Jan/Feb 2001
- [4] J.P. Briot, Agents and components: can they be combined?, SEAS'05