

МОДЕЛ НА РАБОТАТА НА WEB БАЗИРАНА ИНФОРМАЦИОННА СИСТЕМА ЧРЕЗ ОБОБЩЕНА МРЕЖА

Даниела Орозова^{1,3}, Евдокия Сотирова^{2,3}

¹Бургаски свободен университет, Бургас 8001, ул. Сан Стефано 62
e-mail: orozova@bfu.bg

²Университет “Проф. Асен Златаров”, Бургас 8000, България
e-mail: esotirova@btu.bg

³ЦЛБМИ “Проф. И. Даскалов” – БАН, София 1113, България

1. Обектен модел на Web базирана информационна система

Съвременните информационни системи се характеризират с висока степен на географска разпределеност, сложна архитектура и времеви и събитийни модели. Независимо от това всеки обмен на ресурси може да се разложи до множество базови транзакции от типа request – response. Основен подход за изграждане на многопотребителски системи е свързан с използването на компоненти. При разглеждания компонентен модел реализацията на информационната система се основава на обектно-ориентирания подход, като вниманието се насочва към по-високо ниво на абстракция на разглежданите обекти. Отделните модули на системата се дефинират като обектни типове, без ограничение на структурата на изграждащите ги компоненти. Така действието на информационната система може да се разглежда като взаимодействие на различни обекти - обект на потребителския интерфейс, обект на СУБД, обект на файлови операции за вход и изход и т.н. Тогава, в зависимост от това какви екземпляри сме създали, ще имаме различни системи. Реализацията на конкретното приложение представлява избор на желаните обектни типове и създаване на техни конкретни екземпляри. Освен това всеки конкретен обект може да бъде модифициран, заменян или настройван към конкретните нужди на приложението.

Основните предимства на такъв подход са:

- Гъвкавост - системата лесно може да бъде адаптирана към конкретните нужди на потребителя. Например, ако обектът потребителски интерфейс се замени с друг такъв, ще се получи външно различна система, но със същите функционални възможности. Ако пък се замени обектът СУБД, системата може да промени физическата организация на данните си или средствата за извод, но няма да се промени външния си вид.

-Минималност - една система функционира на базата на необходимите обекти и не създава такива, които не използва.

-Мощност - няма ограничение на вида на обектите, които ще се добавят и може да се постигне максимална ефективност.

Системите, създадени чрез такъв подход, търпят сравнително произволно разширение и развитие по отношение на: множеството на създаваните обекти; апаратното пространство - преносимост, настройване към по-мощни апаратни конфигурации.

2. ОМ модел на работата на Web базирана информационна система

Тук конструираме модел чрез обобщена мрежа (ОМ) [1, 2], показан на Фиг. 1, който предлага възможност за проследяване на работата на информационна система. Тази ОМ представя групиране на обектите от гледна точка на различните клиентски заявки, поставени към сървъра. Втора мрежа, която е подмрежа на разглежданата, може детайлно да моделира процеса на динамично генериране на отговор от страна на сървъра. Този процес в някои случаи може да бъде много сложен като включва сериозни изчисления, обработка на данни и изпълнение на SQL-заявки.

Позициите на обобщената мрежа са два типа: a - и b - позиции. Позициите a представят клиентските заявки, съобразно особеностите на разпределената система, а b -позициите, представят статуса на заявките към сървъра. От друга страна отделните клиенти се интерпретират чрез α -ядра, които се движат в a -позициите, а изпълняваните заявки към информационната система се интерпретират като β -ядра, които се движат в b -позициите. По принцип, означаваме α -ядрата чрез $\alpha_1, \alpha_2, \dots$, или накратко, когато няма опасност за грешка – с α ; същото важи за β -ядрата. С цел да не затрудняваме прочита няма да формализираме всички характеристики на ядрата, а ще ги описваме словесно.

В началният момент на функциониране на обобщената мрежа едно или повече α -ядра влизат в позиция a_1 с начална характеристика:

$$x_1^\alpha = \text{"IP address"}.$$

ОМ съдържа следните преходи:

$$T_1 = \langle \{a_1, a_2, b_{10}\}, \{a_2, a_3, a_4\}, r_1, \vee(a_1, a_2, b_{10}) \rangle,$$

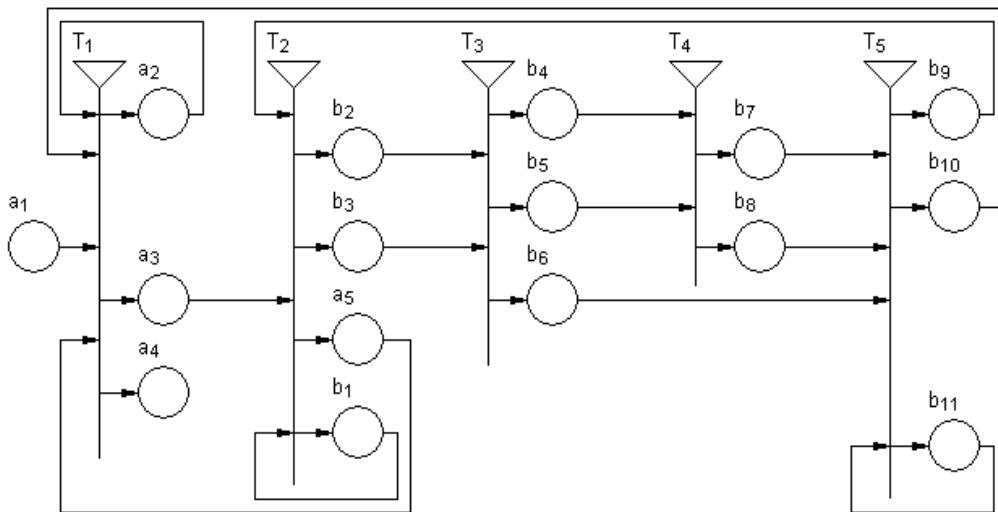
$$r_1 = \begin{array}{c|ccc} & a_2 & a_3 & a_4 \\ \hline a_1 & false & false & W_{1,4}^\alpha \\ a_2 & W_{2,2}^\alpha & W_{2,3}^\alpha & W_{2,4}^\alpha \\ a_5 & false & false & W_{1,5}^\alpha \\ b_{10} & false & false & true \end{array},$$

където

$W_{2,2}^\alpha = \text{"клиентът не извършва заявка към информационната система, а ползва друга мрежова услуга"}$,

$W_{2,3}^{\alpha}$ = “клиентът прави заявка към информационната система (заявката включва еднозначна информация за необходимите ресурси, позволяваща локализацията им и заявените към информационната система действия)”

$$W_{1,4}^{\alpha} = W_{1,5}^{\alpha} = W_{2,4}^{\alpha} = \bigcap W_{2,2}^{\alpha} \& W_{2,3}^{\alpha} .$$



Фиг. 1. OM модел на информационна система

Ядрата, постъпващи в позиция a_2 , получават характеристика:

$$x_2^{\alpha} = \text{„IP адрес, Internet-услуга”},$$

а в позиция a_3 , получават характеристика:

$$x_3^{\alpha} = \text{„IP адрес, заявка към информационната система”}.$$

Първоначално, едно β -ядро стои в позиция b_1 с начална характеристика:

$$x_1^{\beta} = \text{„контейнер към сървъра със всички достъпни сървлети”}.$$

В определен времеви момент ядрото β се разделя на две ядра. Оригиналното β -ядро остава в позиция b_1 , без да получава нова характеристика, а новото β -ядро β_{cu} (нека го означаваме по-нататък като β -ядро) постъпва в позиция b_2 или b_3 .

Преходът включва пренос на заявката по мрежата и декодиране на заявката на приемащата страна. Той има вида:

$$T_2 = \langle \{a_3, b_1, b_9\}, \{b_1, b_2, b_3, a_5\}, r_2, \vee(\wedge(a_3, b_1), b_9) \rangle,$$

$r_2 =$	b_1	b_2	b_3	a_5
a_3	$W_{3,1}^\alpha$	<i>false</i>	<i>false</i>	$W_{3,5}^\alpha$
b_1	<i>true</i>	$W_{1,2}^\beta$	$W_{1,3}^\beta$	$W_{1,5}^\beta$
b_9	$W_{9,1}^\beta$	<i>false</i>	<i>false</i>	<i>false</i>

където:

$$W_{3,1}^\alpha = \neg W_{3,5}^\alpha,$$

$W_{3,5}^\alpha$ = ”няма връзка със сървъра на информационната система”,

$W_{1,2}^\beta$ = ”клиентът вече се е обръщал към сървъра и в заявката има ”бисквитки”,

$W_{1,3}^\beta$ = ”в заявката на клиента няма ”бисквитки”,

$W_{1,5}^\beta$ = ”заявката не може да бъде изпълнена от информационната система (некоректна)”,

$W_{9,1}^\beta$ = ”необходимо е обръщение към друг сървър, (където се намира базата от данни)”.

β -ядрото, постъпващо в позиции b_2 и b_3 получава характеристика съответно:

$$x_2^\beta = \text{”IP адрес, URL на сървъра, “[<name, value>, <name, value>, ...]”},$$

където *name* е идентификатор на ”cookie” и *value* е стойността ѝ,
и $x_3^\beta = \text{”IP адрес, URL на сървъра”}.$

Преходът T_3 представя процеса на специфична обработка в зависимост от типа на заявката. Всеки път, когато сървърът получи заявка той предизвиква създаване на нова *нишка*, проверява типа на Http заявката – Get, Post, Put, Delete, Trace и т.н.

Този преход има вида:

$$T_3 = \langle \{b_2, b_3\}, \{b_4, b_5, b_6\}, r_3, \vee(b_2, b_3) \rangle,$$

$r_3 =$	b_4	b_5	b_6
b_2	$W_{2,4}^\beta$	$W_{2,5}^\beta$	$W_{2,6}^\beta$
b_3	$W_{3,4}^\beta$	$W_{3,5}^\beta$	$W_{3,6}^\beta$

където:

$W_{2,4}^\beta = W_{3,4}^\beta = \text{”Има заявка за извличане на данни от базата данни на информационната система”},$

$W_{2,5}^\beta = W_{3,5}^\beta = \text{”Има заявка за актуализиране на данни (въвеждане, промяна, изтриване)”},$

$W_{2,6}^\beta = W_{3,6}^\beta =$ “Има заявка за промяна на структурата на базата от данни (чрез език за описание на данните)”.

Ядрата, постъпващи в позиции b_4 , b_5 , b_6 , получават характеристики съответно:

$x_4^\beta =$ ”на нова нишка за клиента: IP адрес, URL на сървъра, cookies, тип на заявката - извличане на данни от БД”.

$x_5^\beta =$ ”на нова нишка за клиента: IP адрес, URL на сървъра, cookies, тип на заявката - актуализиране на данни от БД”.

$x_6^\beta =$ ”на нова нишка за клиента: IP адрес, URL на сървъра, cookies, тип на заявката - промяна на структурата на БД”.

Могат да се определят различни приоритети на изпълняваните нишки [3, 4], така че да не се понижава работоспособността на сървъра. Например, ако имаме заявка, извършваща изчисление отнемащо определено време, то сървлетът може да върне незабавно първоначални резултати, но след това да продължава изчислението, използвайки нишка с по-нисък приоритет. Ако изчисленията изискват повече време, то сървърът може да инструктира браузера да се обърне отново за резултат след няколко секунди, като му изпрати заглавната част Refresh.

$$T_4 = \langle \{b_4, b_5\}, \{b_7, b_8\}, r_4, \vee(b_4, b_5) \rangle,$$

$$r_4 = \begin{array}{c|cc} & b_7 & b_8 \\ b_4 & W_{4,7}^\beta & W_{4,8}^\beta \\ b_5 & W_{5,7}^\beta & W_{5,8}^\beta \end{array},$$

където,

$W_{4,7}^\beta = W_{5,7}^\beta =$ ”няма входни параметри на заявката“,

$W_{4,8}^\beta = W_{5,8}^\beta =$ ”има входни параметри от заявката на обработка“.

β -ядрата в позиция b_7 не получават нова характеристика, а в b_8 получават характеристика:

$x_8^\beta =$ ”IP адрес, URL на сървъра, cookies, тип на заявката, данни получени от входните параметри на заявката”.

В представения тук модел чрез ОМ комуникацията между отделните възли в мрежата се осъществява посредством базов модел request – response. Ако се използва трислойният модел (3 – tier architecture) за изграждане на Интернет базирани многопотребителски системи, то той се реализира като се разграничават различни слоеве, които в същност представляват приложение на един от основните принципи за изграждане на системи базирани на компоненти: отделяне на специфичните услуги в самостоятелни програмни единици.

На последният преход от ОМ се формира отговор, извършва се пренос по мрежата и последваща консумация на ресурс. Тук се генерира резултата от заявката и се връща на клиента или се извършва обръщение към друг сървър, продължаващ обработката.

Това е в случаи, че се използва трислойния модел на разпределена Web базирана информационна система и базата от данни е разположена на друг сървър (различен от сървъра на информационната система). Самият процес на генериране на резултата може да бъде сложен процес, който е възможно да бъде описан детайлно с отделна подмрежа, която в последствие да замести позиция b_{10} чрез йерархичен оператор.

$$T_5 = \langle \{b_6, b_7, b_8, b_{11}\}, \{b_9, b_{10}, b_{11}\}, r_5, \vee(b_6, b_7, b_8, b_{11}) \rangle,$$

$r_5 =$	b_9	b_{10}	b_{11}
b_6	<i>false</i>	<i>false</i>	<i>true</i>
b_7	<i>false</i>	<i>false</i>	<i>true</i>
b_8	<i>false</i>	<i>false</i>	<i>true</i>
b_{11}	$W_{11,9}^\beta$	$W_{11,10}^\beta$	<i>true</i>

където:

$W_{11,9}^\beta$ = ”необходимо е обръщение към друг сървър, (където се намира базата от данни)”

$W_{11,10}^\beta$ = ”формиран е резултат към клиента ”.

β -ядрото в позиция b_9 получава характеристиката

x_9^β = “ IP адрес, URL на сървъра, cookies, тип на заявката, URL на сървъра за препращане на заявката”

а в позиция b_{10} получава като характеристика

x_{10}^β = ” IP адрес, URL на сървъра, cookies, резултат от изпълнението на заявката”.

Ако сървърът не е приключил с пълното изчисление на отговора, то той изпраща частичен резултат на клиента и заглавната Refresh, за да му укаже да се върне след определен интервал за нов резултат. Времето за повторно обръщение може да бъде изчислено на базата на получените оценки, относно времето на изпълнение на отделните заявки на сървъра.

3. Заключение

Разглежданият модел може да се счита като един от възможните модели на процеса на функциониране на Web базирана информационна система. Могат да бъдат направени много конкретизации чрез използване на йерархични оператори, които заместват даден преход или позиция с подмрежа, имаща същото, но по-детайлно поведение. На базата на такъв модел и натрупана статистика от реални данни могат да се правят оценки, относно времето за отговор на клиентските заявки. Могат да се търсят оптимални решения относно ограничаване на броя на заявките, обслужвани от даден сървър. Могат да се въведат допълнителни параметри на модела или да бъдат разглеждани допълнителни характеристики на ядрата, като се отчитат различни фактори, влияещи на процеса и да се постигне оптимизация по отношение на дадената цел.

Литература

- [1] Atanassov, K., Generalized Nets, World Scientific, Singapore, 1991.
- [2] Atanassov, K., H. Aladjov. Generalized Nets in Artificial Intelligence. Vol. 2: Generalized Nets and Machine Learning. Academic Publishing House, Sofia, 2001.
- [3] Hall M., Core Servlets and JavaServer Pages, Second edition, 2001, Pearson Education Company.
- [4] <http://java.sun.com/docs/index.html>