

МОДЕЛИРАНЕ НА ПРОЦЕСИ В ИНФОРМАЦИОННИТЕ СИСТЕМИ

Валентина Кукенска
Технически университет - Габрово

PROCESS MODELING IN INFORMATION SYSTEMS

Valentina Kukenska
Technical University of Gabrovo

Abstract

The modeling of processes in information systems is related to the description of the way of their development. Most often it is abstract and is presented from a certain point of view. Can describe the individual roles of the participants in the development; the activities that each role must perform; data conversion actions; input, output parameters and dependencies.

Modeling languages, description templates, diagrams, functions, functional trees, etc. are used to model software processes in information systems. Petri nets, generalized networks, finite state machines, etc. can be used as modeling tools. The choice of modeling tool is related to the approach used in the design of information systems.

This report examines models of software processes related to different approaches to information systems design.

Keywords: modeling; processes modeling; information systems.

ВЪВЕДЕНИЕ

Моделирането на процеси в информационните системи е свързано с описание на начина на тяхното разработване. Най-често то е абстрактно и е представено от определена гледна точка. Може да описва отделните роли на участниците в разработката; дейностите, които всяка роля трябва да реализира; действия, свързани с преобразуване на данни; входни, изходни параметри и зависимости.

За моделирането на софтуерните процеси в информационните системи се използват езици за моделиране, шаблони за описание, диаграми, функции, функционални дървета и др. Като средства за моделиране могат да се използват мрежи на Петри, обобщени мрежи, крайни автомати и др. Изборът на средство за моделиране е свързан с използвания подход при проектиране на информационните системи.

В този доклад се разглеждат различни модели на софтуерни процеси, свързани с

различни подходи при проектиране на информационни системи.

ИЗЛОЖЕНИЕ*1. Моделиране на процеси.*

Моделът на процес представлява абстрактно описание на начина на разработване на информационната система, представено от определена гледна точка. В зависимост от нея моделите на процесите се разделят на:

- **Модел на потока от данни (dataflow model)** – представя дейностите, свързани с преобразуване на данни.

- **Модел на потока от дейности (workflow model)** – представя последователността от дейности, които се извършват в процеса на разработка на информационните системи.

- **Модел на роля/действие (role/action model)** – описва отделните роли на участниците в разработката на системи-

те, както и дейностите, които всяка роля трябва да извърши.

Моделите на процесите могат да бъдат описателни и предписателни. **Описателните модели** описват историята на разработването на информационните системи. Тези модели са специфични за всяки тип система.

Предписателни модели показват как трябва да се разработи новата информационна система. Те се използват като рамки за организиране и структуриране на дейностите и тяхното изпълнение във времето.

Едни от известните и използвани предписателни модели са:

Каскаден модел (модел на водопада)

Този модел предлага последователен подход при разработването на информационните системи. При него разработването на системите е разделено на пет основни стъпки/дейности:

1. събиране на изискванията към системата;
2. планиране – изготвяне на график;
3. анализ и проектиране;
4. реализация (генериране на код) и тестване;
5. внедряване и поддържане.

Моделът на водопада има следните характеристики:

- всяка дейност трябва да бъде завършена, преди да се премине към следващата;
- всяка стъпка завършва с документиране;
- ясно определени роли на разработчиците на системата;
- ясен и лесен за разбиране процес на разработване;
- трудно се адаптира към променящи се изисквания на клиента в процеса на разработката.

Модел на бързата разработка (RAD-Rapid Application Development)

Този модел има за цел да се намали времето за разработка на информационните системи. Основа се на модела на водопада. За разлика от него тук дейности 3 и 4 (виж каскаден модел) да се реализират паралелно от екипа разработчици. Например, всеки

от екипа или един екип да проектира и разработва модул и/или различна функционалност от информационната система.

Намаляване навремето за разработка може да се постигне и чрез средства за автоматично генериране на код, както и чрез имплементиране на готови софтуерни компоненти.

Еволюционни (фазови) модели

Фазовите модели могат да бъдат постъпкови (инкрементални) и итеративни.

Инкрементален (постъпков) модел

При инкременталният модел разработката на информационната системата се реализира на стъпки, като всяка стъпка се доставя само част от цялата функционалност. При него на клиента се предоставят последователност от версии, наречени инкременти. Всяка следваща версия има повече функционалности от предходната.

Итеративен модел

При този модел на клиента се доставя цялостно решение (система). На всяка следваща итерация се доставя все по-добра (позавършена и с по-добро качество) версия на информационната система.

Фазовите модели имат следните характеристики:

- на всяка стъпка се доставя част от функционалността на системата и клиентът може да я използва, преди да е готова цялата система;
- първите стъпки могат да служат като прототип, чрез които да се определят изискванията към следващите стъпки от реализацията на системата;
- по-малък риск от неуспех на целия проект;
- позволява използването на нови технологии в самия процес на разработка.

Прототипен модел

Основната цел на прототипния модел е създаване на прототип, който да извлече или валидира изискванията към информационната система. Има два типа прототипи – еволюционен и хвърлен (throwaway). Целта на еволюционния прототип е да достави на потребителя работеща система, а на хвърления - да подпомогне специфицирането на изискванията към системата.

Прототипният модел може да се изпол-

зва самостоятелно и в комбинация с другите модели. Неговото използване е удачно в проекти, за които не са достатъчно ясни потребителските изисквания и дизайна на информационната система.

Спираловиден модел

Този модел се представя чрез спирала. Всяко пълно завъртане по спиралата представлява една стъпка от разработката. Специфична негова характеристика е анализ на риска. Спираловидният модел е итеративен подход и има множество точки от прогреса (anchor point milestones). При него при всяко завъртане по спиралата се преминава през следните четири сектора:

1. Определяне на целите – определят се целите на текущата фаза от разработката;
2. Оценка на риска – идентифицират се и анализират се потенциалните рискове. Предприемат се действия за намаляването или елиминирането им.;
3. Разработка и валидиране – избира се модел за разработка на текущата фаза;
4. Планиране – анализира се текущото състояние и се планира следващото завъртане по спиралата.

Спираловидният модел е подходящ при разработване на големи информационни системи. Той може да се прилага през целия жизнен цикъл на системите.

1.2. Езици за моделиране на процеси.

Езиците за моделиране на процеси се използват за представяне на характеристиките на софтуерните процеси. Те имат за цел подпомагане на разбирането, дизайна, симулирането, оптимизирането и поддържането на процеса. Чрез тях се описват дейностите, които трябва да се извършат, ролите на участниците в процеса и средствата, които се използват при проектиране и реализация на информационни системи.

Класификацията на езиците за моделиране на процеси може да се направи по различни критерии/признаци.

Спрямо тяхната организация езиците за моделиране на процеси се разделят на:

- **Entity – Relation** – организирани са като същности, с връзки между тях. Използват се основно за моделиране на процеси в информационните системи.
- **Role – Interaction** – описват ролите и техните взаимоотношения.

- **Object - Oriented** – основна единица при тях е обект. За всеки обект се описват данни и функционалност.

Спрямо тяхната формалност езиците за моделиране на процеси се разделят на:

- **Формални** – представят се чрез формален синтаксис и семантика;
- **Полуформални** – обикновено имат графично представяне с формален синтаксис и неформална семантика (могат да бъдат интерпретирани по различен начин);
- **Неформални** – нямат строго дефинирани правила, а смисълът на конструкциите им е от реалния опит и употреба.

За моделиране на процеси в информационните системи се използват следните типове езици:

- **Дескриптивни** (логически) – използват правила или тригери.
- **Мрежово базирани** (езици на базата на мрежи) – представят процесите като мрежи, мрежи на Петри, обобщени мрежи, системи за масово обслужване и др.
- **Императивни** – базирани на езиците за програмиране. Процесът е представен като програма.

1.3. Шаблони за описание на процес.

Шаблонът е описание на решение на общ проблем или въпрос, на базата на което може да се извлече детайлно решение на специфичен проблем.

Шаблон на процес е шаблон, който описва подход и/или последователност от действия за разработване на софтуер. Той представлява структурирано описание на процес. Описва какво трябва да се направи, а не как трябва да се направи.

Шаблоните могат да бъдат дефинирани на различни нива на абстракция. По нарастваща степен на абстрактност типовете шаблони са:

Шаблон на задача на процес (Task process patterns) – описва отделни стъпки за извършване на действие или задача, които са част от процеса;

Шаблон за етап на процес (Stage process patterns) – представя стъпките, които се извършват в рамките на една базова дейност на процеса;

Шаблон за фаза на процес (Phase process patterns) – дефинира последовател-

ността и взаимодействието между базови дейности в рамките на процеса.

Всеки шаблон на процес трябва да съдържа следните елементи:

✓ Начален контекст – описва условията, които трябва да са налице, за да е приложим шаблонът;

✓ Проблем – описание на проблема, които трябва да се реши;

✓ Решение – описва се какво прави шаблонът с цел да разреши проблема. За целта могат да се използват както текстове, така и диаграми;

✓ Краен контекст – какви дейности трябва да са изпълнени, какво трябва да е състоянието на процеса, каква информация е обработена/събрана;

✓ Шаблони с които е свързан – списък на всички шаблони на процеси, които са директно свързани с този шаблон.

✓ Начин на употреба – описание на случаите, в които шаблонът е приложим.

Шаблоните на процеси предоставят ефективен механизъм за описание на всеки софтуерен процес. Те позволяват да се разработи йерархично описание на процесите на разработка на информационните системи от най-високото до най-ниското ниво на абстракция.

1.4. Функции и функционални дървета.

Функциите обикновено описват определени дейности или задачи в дадена проблемна област. Те доставят определени изходни данни на основата на входни данни и предизвикват промяната на съдържанието или структурата на определени данни. Като функции могат да се представят различни задачи, които трябва да изпълнява разработваната информационна система. Функции могат да бъдат подпрограми (функции, процедури, методи) на език за програмиране, избран за реализация на системата.

Декомпозирането на задачите на по-прости функционални идентичности може да се представи като функционално дърво. То представя определена йерархия от функции (Функции → Подфункции → ...). Възможни са различни интерпретации на една такава йерархия.

Основните характеристики на функционалните дървета са следните:

- ✓ Представят концепция за систематична разработка и декомпозиция на сложни проблеми (системи);
- ✓ Дават първоначална насока за представяне на диалога между участниците в процеса;
- ✓ Представят само функционалния аспект на системите.

1.4. Диаграми на потока от данни.

Диаграмите на потока от данни (Data Flow Diagrams – DFDs) представят разработваната система като един информационен поток от данни между функции, памет и интерфейси. При движението данните могат да бъдат трансформирани от един в друг вид.

Съществуват различни възможности за представяне на потоците от данни. Една от тях е за базовите елементи на диаграмите да се използват следните графични символи:

✓ Поток от данни – представя се като именувана стрелка;

✓ Функция (съответства на процес) – представя се като именувана обръжност;

✓ Памет (място за съхранение на данни) – представя се като две паралелни линии, между които се задава името;

✓ Интерфейс (към външни обекти) – представя се като правоъгълник, именуван с името на интерфейса.

Базовите елементи могат да се комбинират за представяне на по-сложни структури на информационни потоци.

Приложението на диаграмите на потока от данни е свързано със следните правила:

- Всяка диаграма трябва да съдържа поне един външен обект;
- Всеки външен обект се представя само веднъж;
- Всеки поток от данни има име;
- Между интерфейсите към външни обекти не се представят потоци от данни;
- Между външните обекти и паметите трябва винаги да се дават функции.

Диаграмите описват потоци от данни. Те не съдържат възможности за разклонения и цикли. Интерфейсите в диаграмите трябва да се избират така, че да дават ясна представа за оригиналния източник или предназначение на информацията. Изборът на

интерфейси не зависи от конкретния начин на въвеждане на информацията.

Диаграмите на потоци от данни лесно се създават и са лесни за разбиране. Те съдържат повече информация от функционалните дървета.

1.5. Диаграми на сценарии.

Сценариите се представят като диаграми, които описват:

- ✓ Бизнес процесите в една информационна система;
- ✓ Връзките между бизнес процесите;
- ✓ Връзките между бизнес процесите и актьорите на системата.

Сценариите описват функционални изисквания към разработвано приложение.

Бизнес процесите обикновено се описват като сценарии за използване (use cases).

При създаване на диаграми за използване анализаторите първо идентифицират актьорите на системата, а след това и самите сценарии. Актьорите на системата са субектите на разработваната система. Те могат да бъдат един или група потребители, както и различни видове външни устройства или софтуер.

Актьорите представят определени роли, които потребителите изпълняват при работа със системата. Формално те се дефинират като субекти, които комуникират с информационната системата и са външни за нея.

Принципната разлика между потребител и актьор се състои в това, че потребителят може да изпълнява различни роли, докато на актьорът се присвоява само една.

За представяне на сценариите могат да се използват различни инструменти:

- ✓ Текстово представяне (текстова схема) – шаблон на естествен език, по който трябва да се опише всеки сценарий;
- ✓ Диаграми на сътрудничество (collaboration diagrams) – представят сценариите като обекти на информационната система и връзки между тях. Представят се и съобщенията, обменяни между обектите;
- ✓ Диаграми на последователности (sequence diagrams) – при тях обменяните

съобщения са подредени последователно във времето;

- ✓ Диаграми на действия (activity diagrams) – представят сценария като алгоритъм;
- ✓ Крайни автомати (state-transition diagrams) – представят поведението на даден обект в системата като възможни състояния, които той може да приема по време на изпълнение на сценария;
- ✓ Мрежи на Петри;
- ✓ Обобщени мрежи.

ЗАКЛЮЧЕНИЕ

За моделирането на софтуерните процеси в информационните системи се използват езици за моделиране, шаблони за описание, диаграми, функции, функционални дървета и др. Като средства за моделиране могат да се използват мрежи на Петри, обобщени мрежи, крайни автомати и др. Изборът на средство за моделиране е свързан с използвания подход при проектиране на информационните системи.

БЛАГОДАРНОСТИ

Работата по този доклад е извършена благодарение на финансовата подкрепа по проект 2006Е: IoT-базирана услуга за персонализирано доставяне на съдържание и автоматично профилиране на клиентите.

REFERENCE

- [1] Ilieva S., V. Lilov, I. Manova, Approaches and methods for realization of software systems, Sofia, 2010.
- [2] SWEBok, Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2014.
- [3] ACM Transactions of Software Engineering and Methodology, <http://www.acm.org>.
- [4] IEEE Transactions on Software Engineering, <http://www.computer.org/tse/>.
- [5] Stoyanov S., A. Doicheva, M. Trendafilova, E. Doichev, Software technologies, Paisii Hilendarski University Publishing House, Plovdiv, 2006.