

Možnosti modelování procesů v multiagentních programovacích jazycích

Jan Smolík

Abstrakt. Problémem současných přístupů k procesnímu modelování je, že ne vždy je možné nebo vhodné proces popsat jako uspořádaný sled činností. Autor se domnívá, že tento problém může překonat využití objektového přístupu. Ten se však v konceptuálním modelování příliš neprosadil. Ukazuje se, že na vině je zřejmě nevhodně definovaná sémantika. Autor ukazuje, že chápáním objektů jako aktorů/agentů lze tento problém eliminovat. V literatuře pak nachází a shrnuje existující přístupy k agentovému modelování procesů. Ty v závěru navrhuje rozšířit o simulaci průběhu procesu pomocí multiagentového jazyka, jelikož tyto přístupy zobrazují spíše strukturu procesu, než jeho průběh.

Klíčová slova: konceptuální modelování, procesy, agenti, objekty

1 Úvod

Procesní řízení je realitou současného podnikového světa. Pokud chceme pochopit procesy probíhající ve firmě, musíme je umět popsat. K tomu existuje spousta metod, z nichž mezi nejznámější patří např. ARIS prof. Scheera či jazyk BPMN.

Společným znakem těchto přístupů je, že chápou proces jako jednoznačně uspořádaný sled činností. Nezapřou přitom značnou inspiraci tzv. vývojovým diagramem (flowchartem). Ze zkušeností autora vyplývá, že studenti, kteří se učí tyto metodiky, popisují zpočátku procesy spíše vývojovým diagramem, než pochopí specifické sémantické prvky popisu procesů.

Ne vždy jsou však činnosti v procesu jednoznačně uspořádané. Všímá si toho např. [8], který identifikuje nedostatky detailního popisu procesů pomocí stávajících metod. Za prvé, při velkém množství podmínek je popis procesu příliš složitý. Za druhé, u některých procesů jasnou posloupnost nelze určit (uváděný příklad je tvorba informační strategie firmy). Za třetí, příliš detailní popis procesu omezuje kreativitu. [8] tedy navrhuje čtyři úrovně podrobnosti popisu procesů, kdy u nejnižší se stanoví pouze cíl procesu, který se v dalších úrovních doplňuje o zdroje činnosti a teprve na čtvrté úrovni se stanovuje návaznost těchto činností.

Ostatně i nejznámější Hammerova definice procesu o nutnosti uspořádaných činností nemluví: „Definujeme podnikový proces jako soubor činností, který vyža-

duje jeden nebo více druhů vstupů a tvoří výstup, který má pro zákazníka hodnotu.“ [3]

Domníváme se, že možnou odpovědí na problémy stávajících procesních modelů je užití objektového přístupu a konkrétně jeho zvláštní odnože – agentů. V části 2 ukážeme problémy objektového přístupu v konceptuálním modelování, které spočívají ve špatně definované sémantice. Část 3 ukazuje možnosti definování sémantiky objektů. Čtvrtá část naznačuje odlišnosti agentů od běžných objektů. Část 5 se zabývá existujícími přístupy pro modelování procesů pomocí agentů. Šestá část pak navrhuje rozšíření těchto modelů o simulaci v agentním programovacím jazyce a naznačuje možné mapování konceptů.

2 Problémy objektového přístupu v konceptuálním modelování

Objektový přístup se v posledních deseti až patnácti letech prosadil v programování a v analýze a designu informačních systémů. Svědčí o tom popularita objektových programovacích jazyků Java či C++ i fakt, že původně neobjektové jazyky (např. PHP) do sebe postupně objekty integrují. V oblasti designu si jazyk UML [7] vydobyl pevné místo.

V konceptuálním modelování se však objektový přístup příliš neprosazuje. Jazyk UML je zaměřený na návrh informačního systému a jeho koncepty jsou více-méně programátorské. Asi neúspěšnější aplikací jazyka UML pro modelování byznys procesů (a tudíž pro konceptuální programování) je Eriksson-Penkerův profil (rozšíření). Ten se však zaměřuje na externí pohled na procesy a jejich vazby, nikoliv na jejich vnitřní strukturu.

Wand[10] vidí jako hlavní příčinu neproniknutí objektového přístupu do konceptuálního světa nedostatečně objasněnou sémantiku. Jako příklad uvádí přihlášení studenta na předmět. To bude v objektovém přístupu pravděpodobně popsáno tak, že objekt student zašle zprávu objektu předmět, a vyvolá tak metodu, která ho zapíše. V konceptuálním světě však tento obrázek působí podivně, protože studenti nemluví s předměty.

K podobnému závěru dochází i [9]. Pojem objekt podle něj v sobě zahrnuje několik odlišných konceptů. Jako objekt tak bude označen např. grafický prvek (dejme tomu tlačítko) užitý v programu. Stejně tak bude objektem bankovní účet, ale i zákazník, který je ve skutečnosti aktivním aktorem (agentem).

Jako mezizávěr tedy konstatujme, že špatně definovaná sémantika objektů působí problémy v jejich užití v konceptuálním modelování.

3 Možné sémantické interpretace objektů v konceptuálním modelování

Sémanticky lze objekty v zásadě definovat dvojím způsobem. Můžeme je chápat jako pasivní entity, kterými je v procesech manipulováno. Takto je chápe například [6]. To ostatně odpovídá i samotnému významu slova „objekt“. Problém je pak se sémantikou metod. [6] metody chápe jako akce s objektem provedené (tedy z pohledu objektu pasivní) a vidí v nich přechody mezi jednotlivými stavy životního cyklu objektu.

Druhým přístupem je chápat objekty jako aktivní entity. To ostatně odpovídá i programátorskému přístupu. O tento přístup se pokoušejí například [10] [9][4][5] [11]. Ve většině případů (i když ne ve všech) nastaví na objektovém přístupu jako takovém, ale na agentovém přístupu, který k němu má blízko (rozdíly viz dále). Toto aktivní pojetí objektů jako aktorů (agentů) v zásadě není v rozporu s pasivním pojetím objektů, nicméně je třeba tyto dva koncepty odlišit. Proto i pojmové odlišení pomocí slov objekt (pasivní entita) a agent/aktor (aktivní entita). Toto rozlišení budeme používat v dalším textu.

S konceptem aktorů/agentů souvisí koncept rolí, který popisuje například [1]. Aktoři mohou vystupovat v systému v různých rolích, přičemž jeden aktor může zároveň vystupovat v různých rolích (nebo dokonce vícekrát v jedné), může role nabývat a ztrácet, aniž by přitom zanikala jeho existence. Některých rolí může aktor nabývat pouze ve vztahu k jiným aktorům. [1] provádí klasifikaci konceptů souvisejících s rolemi a rozděluje je na individuální (představující konkrétní instanci aktora) a univerzální (představující „třídu aktora“).

[1] dále hledá koncepty rolí v jednotlivých metodikách určených ke konceptuálnímu modelování firmy. Dochází k závěru, že tento koncept je sice v modelovacích metodikách hojně užíván, ale často není sémanticky ujasněný (jako v přístupu Archimate či ARIS), případně se vyskytuje pouze okrajově (BPMN).

Domníváme se, že koncept rolí je pro popis firmy přesnější než představa aktora/agenta jako fyzické entity. Jeden zaměstnanec může vystupovat v různých rolích. Například webmaster může být ve firmě podle své kvalifikace zároveň správcem podnikového serveru či pracovníkem public relations. Jako agenti by tedy v agentově orientovaném popisu byznys procesu měli vystupovat spíše role než fyzické entity.

4 Odlišnosti mezi agenty a objekty

Agenty v tomto textu chápeme jako aktivní objekty. Nicméně existují i některé další odlišnosti. [4] uvádí následující vlastnosti agentů: autonomie, sociální schopnosti, proaktivnost a citlivost (responsivness). Autonomií se myslí, že agent je

schopen vykonávat většinu činností samostatně. Sociálními schopnostmi se rozumí možnost komunikace agenta s ostatními. Toto je na jisté nízké úrovni i u programátorských objektů, naopak u konceptuálních to jaksí ztrácí význam. Proaktivnost znamená, že agent přebírá iniciativu, když je to vhodné, nečeká na zavolání metody zvenčí, jako je tomu u objektu. Citlivost (možná spíše vnímavost) pak znamená, že agent vnímá své okolí a reaguje na podněty z něj.

[9] vidí tři následující vlastnosti agentů: schopnost vnímat okolí, schopnost svými akcemi toto okolí ovlivňovat a schopnost usuzovat (reason) k interpretování počitků z okolí, řešení problémů a stanovení akcí. Důraz je zde tedy kladen spíše na interakci s okolím, nicméně i z těchto vlastností je patrná autonomnost a proaktivnost agentů – reagují na podněty z okolí bez toho, aby byla explicitně vyvolána jejich metoda.

Agenty tedy můžeme chápat jako nezávislejší entity než objekty, které autonomně reagují přímo na podněty z okolí. Vzhledem k tomu, že s objekty sdílejí spoustu vlastností (např. v jisté míře schopnost komunikovat), je aktivní objekt jejich blízkou aproximací. Ostatně jako objekty jsou agenti v některých případech i implementováni.

5 Existující přístup k modelování procesů pomocí agentů

Podívejme se nyní na existující metody modelování byznys procesů pomocí agentů. Vzhledem k rozsahu tohoto článku jen naznačíme jejich hlavní myšlenky, pro podrobnější popis odkážeme čtenáře do příslušné literatury.

5.1 Object Oriented Enterprise Modeling

Metodu Object Oriented Enterprise Modeling prezentovaná v [10] lze přiřadit k těm, kde jsou objekty chápány jako agenti. Nicméně autoři sami tento pojem nepoužívají. Jedná se o přístup k modelování procesů, kde je sémantika všech prvků přísně založena na formálních ontologických pojmech. Jako objekty jsou chápány jen ty entity, které přímo či nepřímo reagují na požadavek zvenčí. Pasivní objekty jsou pak chápány nikoliv jako samostatné entity, ale atributy aktivních objektů, jejichž změny vyvolávají změny stavu objektu (agenta) a nějaké akce.

Objektem se v tomto přístupu může stát jak člověk, tak část informačního systému. V příkladu autoři připouštějí i existenci objektu sklad, který je zde ve významu oddělení tj. abstraktního složeného aktora. Metody nejsou chápány jako něco co je z vnějšku voláno, ale jako možné reakce objektu (které si sám zvolí) na změnu svého stavu vyvolanou akcí z venku. Přestože slovo agent není v přístupu užíváno, má tato metoda k agentovému přístupu minimálně velmi blízko.

5.2 ADEPT

Metoda ADEPT popsaná v [4] je určena pro implementaci agentově orientovaného workflow systému. Nejedná se tedy o čisté konceptuální modelování procesů, ale spíše o modelování procesů v informačním systému. Metoda byla úspěšně aplikována na proces poskytování nabídek zákazníkovi firmy British Telecom, čímž byla prokázána jeho aplikovatelnost.

Přístup ADEPT propojuje agenty a procesy se službami. Agenti mají definované služby, které mohou vykonávat. Nicméně žádná služba nemůže být vykonána bez existující SLA. Agenti, kteří podle přístupu musejí mít sociální schopnosti, jsou schopni si mezi sebou takovouto SLA vyjednat. K tomu slouží komunikační akce jako je návrh, protinávrh, přijetí či odmítnutí návrhu.

Autoři umožňují existenci jak jednoduchých agentů (lidí), tak složených agentů (oddělení). To podle autorů umožňuje modelování různě hierarchicky strukturovaných organizací.

5.3 Agent-Object-Relationship Metamodel

Přístup Agent-Object-Relationship (AOR) navržený v [9] navrhuje rozdělení objektů na pasivní a aktivní. Oba tyto typy jsou modelovány v diagramu agentů. V diagramech, které se zabývají dynamikou systému, však hrají hlavní roli agenti.

Klíčovými koncepty jsou závazek a právo, které k sobě agenti mají navzájem. Dalším konceptem pak je událost. Ta může buď vzniknout v prostředí, nebo může být vyvolána akcí. Akce se pak dělí na komunikační a nekomunikační. Události pak mohou být buď vnímány, nebo v případě událostí způsobených komunikačními akcemi vysílány a přijímány.

Metoda připouští jak existenci lidských a softwarových agentů, tak i roli agentů institucionálních – složených. Kromě diagramu agentů (obdobu diagramu tříd) pak zahrnuje diagram interakčního rámce, který staticky zachycuje základní vztahy pomocí výše uvedených konceptů. Dalším diagramem pak je diagram interakční sekvence, který velmi připomíná sekvenční diagram UML. Nicméně komunikace je opět zachycena pomocí konceptů AOR. Doplnkem pak je interní model jednotlivých agentů, který však pro potřeby tohoto článku není důležitý.

5.4 An Agent-Oriented Modelling Framework (i*)

Metoda i* popsaná v [11] používá diagramu strategických závislostí k vyjádření závislostí agentů na sobě. Agent může na druhém záviset kvůli dosažení cíle či podcíle, případně na vykonání nějaké činnosti nebo poskytnutí zdroje.

Diagram strategických zdůvodnění pak ukazuje, jak jsou tyto koncepty propojené mezi sebou. Činnost, na které agent závisí například může pozitivně či nega-

tivně přispět k naplnění jeho podcíle, či cíle. Výhodou tohoto modelu (a tím i celého přístupu je, že ukazuje problémy a neefektivnosti (negativní příspěvky) a vede tak reengineeringu procesu.

5.5 Conceptual Agent Model

CAM (Conceptual Agent Model) je metoda definovaná v [5]. Navazuje na metodu OEM (viz výše). Stále je ve značné fázi rozpracovanosti a tvůrčích záměrů, než aby se dala přímo použít. Autor však navrhuje ukázkový diagram, který je vhodný pro další práci s agenty. Každý agent je prezentován jako obdélník, který má v sekcích napsány cíle, procedury, víry (beliefs), odvozování a akce. Dále jsou pak v trojúhelníkových mimo objekt naznačeny stavy prostředí, které agent pozoruje, nebo naopak ovlivňuje. Ty ukazují vazby mezi agenty. Chápeme tento diagram jako dobrý předpoklad pro simulaci procesu pomocí agentů.

6 Možnosti simulace procesů v agentových programovacích jazycích

Zjištěné metody mají společně to, že popisují spíše strukturu procesu, než jeho dynamiku. To rozhodně platí pro metodu OEM, i* a CAM. ADEPT je workflow systém, který v konečné implementaci má dynamiku, nicméně užité diagramy jsou opět statické. AOR má diagramy znázorňující dynamiku, nejsou však příliš přehledné.

Zdá se, že zobrazení dynamiky, která nemá přesně dané pořadí akcí, a skládá se z předem nepředpověditelných reakcí agentů na podněty, není jednoduché. S dostupnou počítačovou technikou však není nutné se při modelování spoléhat na dvourozměrný obrázek. Náš návrh pro zobrazení dynamiky procesu pomocí agentů spočívá v simulaci tohoto procesu v agentovém programovacím jazyce.

Možnosti mapování sémantiky výše zmíněných přístupů na multiagentní programovací jazyk naznačíme na příkladě jazyka Jason [2]. Jazyk JASON je založený na modelu BDI (belief-desire-intention), který popisuje vnitřní vlastnosti agentů. Víry (beliefs) jsou informace, které má agent o okolním světě. Tato informace nicméně nemusí být správná. Touhy (desires) jsou takové stavy, které by agent mohl chtít dosáhnout, nicméně není jisté, že se pro ně nakonec rozhodne. Intentiony (záměry) jsou pak takové stavy, kterých se agent rozhodl dosáhnout.

V jazyce JASON jsou pak tyto koncepty implementovány pomocí věr (beliefs), cílů a plánů. Cíle jsou stavy, kterých chce agent dosáhnout, plány pak jsou cesty, jak se k těmto cílům dopracovat (jsou aktivovány zároveň s přidáním cíle agentovi). Rozhodovací mechanismus (reasoning) pak agentovi přidělí, kterému cíli a plánu se bude v dalším kroku věnovat. Agenti si mohou zasílat zprávy, a mohou

rovněž vnímat a ovlivňovat společné prostředí (to je implementováno v jazyce Java).

Podívejme se, jak koncepty jednotlivých přístupů odpovídají konceptům jazyka JASON. Začneme přístupem AOR. Závazky a práva mohou být v Jasonu vyjádřeny jako cíle, respektive víry. Události v Jasonu existují a jsou agenty vnímány pomocí počitků (perception) okolí. Komunikativní akce budou zasílané zprávy a nekomunikativní akce ty části plánu, které ovlivňují prostředí.

V modelu CAM cíle představují spíše abstraktní smysl života, který v Jasonu není vyjádřen. Nicméně pro model ho není třeba. Procedury v CAMu pak odpovídají cílům, víry, odvozování a akce pak v Jasonu opět existují. Za akce bychom však měli opět považovat pouze ty části plánu, které ovlivňují prostředí, případně zasílají zprávy jinému agentu.

7 Závěr

Ukázali jsme, že nedostatečné prosazení objektového přístupu pro konceptuální modelování procesů může mít svou příčinu v nedostatečné sémantice. Problém způsobuje především dichotomie objektů, kdy některé jsou aktivní a jiné pasivní. V literatuře lze vysledovat řadu přístupů, které tuto dichotomii vysvětlují a většinou pomocí agentového přístupu vytvářejí zvláštní třídu aktivních objektů.

Problémem je, že tyto přístupy v diagramech zobrazují spíše strukturu procesů, než jejich návaznost. To má svou logiku, činnosti v procesu nemusí mít jednoznačné uspořádání. Náš návrh na vytvoření srozumitelného modelu spočívá v simulaci procesu v některém agentovém jazyce. V článku naznačujeme možnosti mapování konceptů jednotlivých přístupů na koncepty jazyka JASON.

Upozorňujeme, že toto mapování rozhodně není finální a bude třeba dalšího výzkumu, aby se jej podařilo jednoznačně finalizovat. Jedná se skutečně spíše o naznačení možností. Rovněž je třeba podotknout, že některé koncepty bude třeba pro implementaci v JASONu doplnit. Například činnosti vykonávané agenty v realitě bude třeba nahradit zástupnými činnostmi (např. vypiš na obrazovku název činnosti a čekej 10 cyklů představujících 10 hodin).

Modelování procesu v agentovém jazyce bohužel není bohužel jednoduchá záležitost. Vyžaduje totiž programátorské znalosti, které byznys analytici nemusí mít. V dalším výzkumu se však chceme zaměřit na namodelování vzorového procesu, tak, aby se ujasnilo mapování sémantiky a možnosti výstupů. V další fázi, pokud bude jasné mapování konceptů, by bylo možné vytvořit automatizovaný nástroj, který program bude vytvářet.

Chceme rovněž zdůraznit, že neprosazujeme nahrazení klasického procesního modelování objektovým/agentovým přístupem. Jde spíše o jeho rozšíření a řešení situací, pro které se nehodí.

Literatura

- [1] Almeida, J., Guizzardi, G.: *A Semantic Foundation for Role-Related Concepts in Enterprise Modelling* [online]. 2008 [cit. 2009-01-25]. Dostupný z WWW: <<http://www.inf.ufes.br/~jpalmeida/cvitae/almeida-guizzardi-edoc2008.pdf>>.
- [2] Bordini, R., Hubner, J., Wooldridge, M.: *Programming Multi-Agent Systems in AgentSpeak using Jason*. 2007. 272 s. ISBN 978-0-470-02900-8.
- [3] Hammer, M., Champy, J.: *Reengineering - radikální proměna firmy : Manifest revoluce v podnikání*. Vodáček Leo. 2. vyd. Praha: Management Press, 1996. 210 s. ISBN 80-85943-30-1.
- [4] Jennings, N., et al.: *Agent-based business process management* [online]. 1996 [cit. 2009-01-25]. Dostupný z WWW: <<http://www.ecs.soton.ac.uk/~nrj/download-files/IJCIS96.pdf>>.
- [5] Kafui, M.: *A Conceptual Modeling Method to Use Agents in Systems Analysis* [online]. 2008 [cit. 2009-01-25]. Dostupný z WWW: <<http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-343/paper2.pdf>>.
- [6] Řepa, V.: *Podnikové procesy: Procesní řízení a modelování*. 2. rozš. vyd. Praha : Grada, 2007. 288 s. Management v informační společnosti. ISBN 978-80-247-2252-8.
- [7] OMG: *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2* [online]. 2007 [cit. 2009-01-25]. Dostupný z WWW: <<http://www.omg.org/docs/formal/07-11-02.pdf>>.
- [8] Voříšek, J.: *Nová dimenze systémové integrace - integrace podnikových procesů a znalostí, Systémová integrace '2000, sborník mezinárodní konference*, VŠE, Praha, 2000, 195-206, ISBN 80-245-0041-8
- [9] Wagner, G. *The Agent-Object-Relationship Metamodel : Towards a Unified View of State and Behavior* [online]. 2003 [cit. 2009-01-25]. Dostupný z WWW: <<http://is.tm.tue.nl/research/icta/documents/Wagner-AOR03.pdf>>.
- [10] Wand, Y., Woo, C.: *Ontology-Based Rules for Object-Oriented Enterprise Modeling* [online]. 1999 [cit. 2009-01-25]. Dostupný z WWW: <<http://mis.sauder.ubc.ca/members/woo/papers/ooem.rules.pdf>>.

[11] Yu, E.: *Agent Orientation as a Modelling Paradigm* [online]. 2001 [cit. 2009-01-25]. Dostupný z WWW: <<ftp://ftp.cs.utoronto.ca/pub/eric/WIj.pdf>>.

Summary

On the Possibilities of the Business Process Modelling in Multi-Agent Programming Languages

Process modeling is an inseparable part of conceptual analysis used for the analysis and design of an information systems. Current approaches (ARIS, BPMN) model processes as ordered activities. However, this is not useful in many cases. Sometimes the order of activities is unclear or unimportant, or such a detailed description would be unnecessary.

Object oriented approach might be answer to this drawback. However, it is not used much in conceptual analysis. From the literature review it seems that unclear semantics might be the cause. There are two possible semantics for conceptual objects : passive and active (agents/actors).

In this article we analyze approaches that use agent oriented analysis to describe business processes – namely Object Oriented Enterprise Modeling, ADEPT, Agent Object Relationship Metamodel, i* and Conceptual Agent Model.

We conclude, these approaches can provide description to the business processes. However they describe structure of the business process not its flow. Indeed it is difficult to describe flow, where there are many possibilities.

We suggest expanding these approaches by simulating a business process in agent-oriented language environment. We demonstrate it is possible to map concepts of described approaches to the concepts of the agent-oriented language Jason. As of now, this mapping is a mere suggestion that needs further research. The disadvantage is that simulating process requires programming skills.