

Radosław Klimek, Sebastian Ernst

AGH Akademia Górniczo-Hutnicza

ANALIZA MOŻLIWOŚCI FORMALNEJ WERYFIKACJI PROBLEMÓW PLANOWANIA W KONTEKŚCIE PROCESÓW BIZNESOWYCH

Streszczenie: Artykuł nakreśla relacje między modelowaniem procesów biznesowych (BPM) oraz architektury zorientowanej na usługi (SOA) a metodami planowania (*AI Planning*), z uwzględnieniem aspektów takich jak modelowanie dziedzin problemów planowania za pomocą narzędzi modelowania procesów biznesowych czy automatycznej aranżacji usług w procesy za pomocą metod planowania. Nakreślono również nowy kierunek badawczy, związany z formalną weryfikacją dziedzin problemów planowania za pomocą wnioskowania dedukcyjnego, w oparciu o wzorce projektowe van der Aalsta i in. dla modeli biznesowych. Formalna weryfikacja realizowana jest z wykorzystaniem wnioskowania dedukcyjnego metodą tablic semantycznych. Własności modelu opisane są w logice temporalnej czasu liniowego.

Słowa kluczowe: modelowanie biznesowe, planowanie, logika temporalna, formalna weryfikacja, metoda tablic semantycznych.

1. Wstęp

Nowoczesne narzędzia modelowania biznesowego takie jak BPMN¹ dzięki intuicyjnej notacji pozwalają na efektywne kształtowanie procesów biznesowych nawet przez osoby niezaznajomione z zapisem czy zagadnieniami technicznymi. Jednocześnie potencjalnie przydatne w tej dziedzinie metody sztucznej inteligencji, takie jak automatyczne planowanie, nie są wykorzystywane z uwagi na ubogie narzędzia modelowania i brak metod translacji. Tymczasem proces translacji do modeli biznesowych mógłby dodatkowo skutkować możliwością przeprowadzenia formalnej weryfikacji poprawnościowej tak otrzymanych modeli. Weryfikacja poprawności modeli może być realizowana drogą dedukcyjną z wykorzystaniem logiki temporalnej i metody tablic semantycznych jako podstawowej metody wnioskowania. Logika temporalna jest uznanym formalizmem służącym do opisu zagad-

¹ BPMN – *Business Process Modeling Notation* [OMG 2009].

nień związanych z czasem, bez konieczności stosowania rachunku kwantyfikatorów. Z kolei metoda tablic semantycznych jest pewną alternatywą, i ma pewne zalety, w stosunku do tradycyjnego wnioskowania metodami opartymi na rezolucji.

W niniejszym artykule dokonuje się krótkiej charakterystyki metod planowania i narzędzi modelowania biznesowego, opisuje aktualny stan badań związanych z translacją modeli biznesowych na dziedziny i problemy planowania, a także przedstawia propozycję nowego kierunku badań, wraz z omówieniem czynników wpływających na możliwość ich zrealizowania oraz zaleceniami do ich implementacji. Został zaprezentowany aparat formalny związany z badaniem poprawności analizowanych modeli, budową systemu wnioskowania metodą tablic semantycznych oraz możliwościami automatyzacji procesu budowy specyfikacji logicznej modelu. Automatyzacja procesu tworzenia specyfikacji logicznej jest istotna ze względu na niewydajność ręcznego budowania specyfikacji składającej się z bardzo dużej liczby formuł logiki (temporalnej), co w sposób naturalny byłoby procesem podatnym na błędy.

Struktura artykułu jest następująca. Sekcja 2 omawia formalizmy i języki wykorzystywane do zapisu dziedzin i problemów planowania. Sekcja 3 charakteryzuje narzędzia modelowania procesów biznesowych, a sekcja 4 omawia ostatnie badania w zakresie ich formalnej weryfikacji. W sekcji 5 omówiono możliwość integracji dwóch omawianych dziedzin, przedstawiono aktualny stan badań oraz możliwe kierunki jej rozwoju.

2. Metody planowania

Planowanie (*AI planning* lub *automated planning*) [Ghallab i in. 2004] jest działem sztucznej inteligencji i zajmuje się przede wszystkim problemami, których dziedziny są za duże lub zbyt skomplikowane dla klasycznych metod. W szczególności metody planowania wykorzystuje się wtedy, gdy duża złożoność zapisu stanu (np. wiele zmiennych o szerokich dziedzinach) utrudnia zbudowanie grafu przestrzeni stanów, jego klarowne przedstawienie i wykorzystanie klasycznych metod przeszukiwania.

W ogólnym rozumieniu planowaniem określa się zagadnienie wyznaczania całkowicie lub częściowo uporządkowanego ciągu czynności („akcji”), prowadzących do osiągnięcia określonego celu. Tak ogólnie postawiony problem może być rozwiązywany różnymi metodami, zarówno z dziedziny sztucznej inteligencji, jak i badań operacyjnych czy nawet teorii grafów.

Autorzy Russell i Norvig [2010, rozdz. 11] wskazują na trudności związane z rozwiązywaniem złożonych problemów za pomocą klasycznych algorytmów przeszukiwania:

Duża liczba bezużytecznych akcji. Jeżeli agent, planując², może wykonać w określonym stanie wiele podobnych akcji (np. różniących się tylko wartością parametru), klasyczne algorytmy przeszukiwania mogą być nieskuteczne; z drugiej strony, sparametryzowanie akcji tak, jak w opisie dziedziny problemu planowania, pozwala uniknąć takiego rozgałęzienia.

Problem doboru heurystyk. W ostatnich latach wzrosło zainteresowanie planerami (systemami rozwiązującymi problemy planowania) opartymi na przeszukiwaniach heurystycznych [Bonet, Geffner 1999, 2001]. Problemem jest dobór heurystyk, które najczęściej muszą być definiowane przez eksperta znającego charakterystykę problemu. Z drugiej strony, odpowiednia reprezentacja problemu pozwala na automatyczne wyznaczanie heurystyk, co zostało przedstawione m.in. w [Haslum i in. 2005; Helmert, Domshlak 2009; Richter, Westphal 2010].

Zagadnienie dekompozycji problemu. Rozsądny podział problemu na podproblemy pozwala na znaczącą poprawę wydajności planowania. Niestety, wielu problemów planowania nie da się w prosty sposób zdekomponować, gdyż cele częściowe wpływają na siebie nawzajem.

Kolejne sekcje przybliżają formalizmy wykorzystywane do reprezentacji dziedzin i problemów planowania oraz języki używane do ich zapisu.

2.1. Formalizmy używane do reprezentacji problemów planowania

Formalna reprezentacja dziedzin planowania i problemów w nich umieszczonych wymaga określenia spójnej terminologii dotyczącej reprezentacji:

- zmiennych, reprezentujących elementy modelowanej rzeczywistości, wyrażanych jako fakty logiczne (STRIPS) czy skończone zbiory wartości (SAS^+),
- celów, określanych jako stan docelowy – koniunkcja podzbioru zbioru faktów, kwantyfikatory logiczne lub częściowe przypisanie wartości zmiennych stanu,
- stanu początkowego, określonego podobnie jak cel,
- akcji (operatorów), z których każda określa:
 - prewarunki, określone jako stan, który musi być spełniony, aby akcja mogła być wykonana,
 - efekty, określone jako stan po wykonaniu akcji.

2.1.1. STRIPS

STRIPS (*Stanford Research Institute Problem Solver*) jest jednym z pierwszych automatycznych planerów [Fikes, Nilsson 1971; Bylander 1994]. W modelu STRIPS problem określony jest jako krotka:

$$P = \langle F, I, G, O, c \rangle \quad (1)$$

² Termin *agenta* używany jest za [Russell, Norvig 2010] i nie stanowi nawiązania do teorii systemów wieloagentowych.

gdzie: F – zbiór faktów,
 I – stan początkowy, $I \subset F$,
 G – stan docelowy, $G \subset F$,
 O – zbiór operatorów, $O = \langle Pre(o), Add(o), Del(o) \rangle$, każdy $\subset F$,
 C – funkcja kosztu, $c \rightarrow \mathbf{R}$.

Stan w STRIPS definiowany jest jako podzbiór zbioru faktów F o wartości *true*.
 Rozmiar przestrzeni możliwych stanów wynosi więc $O(2^{|F|})$.

2.1.2. SAS⁺

Formalizm SAS [Sandewall, Rönnquist 1986] i stanowiący jego rozszerzenie SAS⁺ bazują na strukturze podobnej do STRIPS, ale bardziej elastycznej. Problem planowania w SAS⁺ definiowany jest jako krotka

$$P = \langle V, I, G, O, c \rangle \quad (2)$$

gdzie: V – zbiór zmiennych, z których dla każdej określona jest skończona dziedzi-
 dzina D_V ,

I – stan początkowy, stanowiący *pełne* przyporządkowanie do zmiennych
 $v_i \in V$ wartości z dziedzin $d_i \in D_{v_i}$,

G – stan docelowy, stanowiący *częściowe* przyporządkowanie wartości do
 zmiennych (jw., ale dla podzbioru V),

O – zbiór operatorów, $O = \langle Pre(o), Eff(o) \rangle$,

c – funkcja kosztu, $c \rightarrow \mathbf{R}$.

2.1.3. Hierarchiczne sieci zadań (HTN)

Prezentowane w sekcjach 2.1.1 oraz 2.1.2 formalizmy posiadają wspólne ograniczenie: każda akcja (operacja) traktowana jest w ich przypadku jako atomiczna encja. Może to stanowić problem w przypadku złożonych dziedzin; hierarchiczna dekompozycja jest również cechą typową dla procesów biznesowych, co jest zagadnieniem kluczowym dla integracji metod planowania z modelowaniem biznesowym.

Formalizmem często wykorzystywanym w planowaniu hierarchicznym są hierarchiczne sieci zadań (*Hierarchical Task Network* – HTN) [Erol i in. 1994; Lekavy, Návrat 2007]. Akcje znane z poprzednio omawianych formalizmów są tu nazywane *akcjami prostymi* (*primitive actions*). Nową koncepcją są *akcje wysokiego poziomu* (*high-level actions* – HLA). Dla każdej HLA określamy możliwe *uściślenia* (*refinements*), składające się z sekwencji kroków (akcji prostych, innych HLA lub tego samego HLA wywoływanego rekurencyjnie) prowadzących do osiągnięcia danego celu nadrzędnego. Uściślenie, na które składają się wyłącznie akcje proste, nazywane jest *implementacją* (*implementation*) danego HLA.

W planowaniu klasycznym algorytmy mogą postępować w przód (*forward planning*) lub wstecz (*backward planning*). Algorytmy planowania hierarchicznego dzielą się dodatkowo na zstępujące (*top-down*) oraz wstępujące (*bottom-up*).

2.2. Języki opisu problemów planowania

Aby zapewnić kompatybilność i ułatwić rozwiązywanie różnych zadań za pomocą różnych narzędzi, ustalono standardy wymiany informacji o dziedzinach i problemach planowania.

Język PDDL (*Planning Domain Definition Language*) powstał w 1998 roku na potrzeby konkursu *International Planner Competition* [McDermott i in. 1998]. Aktualną wersją jest wersja 3.1, opracowana w roku 2008³. Rozszerzenia i kolejne wersje języka pojawiają się zazwyczaj przy każdej edycji zawodów IPC.

Wejście do planera zgodnego z językiem PDDL składa się zazwyczaj z dwóch plików:

- plik z definicją dziedziny, zawierający predykaty oraz akcje,
- plik z definicją problemu, zawierający definicje obiektów, stanu początkowego i celów.

W celu umożliwienia reprezentacji problemów korzystających z modelu HTN stworzono rozszerzenie języka PDDL o nazwie HTN-PDDL. Aby rozwiązywać te problemy, konieczne jest wykorzystanie dedykowanego planera posiadającego obsługę sieci hierarchicznych. Istnieje wprawdzie możliwość translacji z HTN-PDDL na „zwykły” PDDL (co umożliwiłoby zastosowanie „klasycznych” planerów), ale tylko dla problemów spełniających określone ograniczenia (zob. sekcja 5.1).

3. Modelowanie procesów biznesowych

Przez *procesy biznesowe* można rozumieć uporządkowane i powiązane ze sobą aktywności i zadania, pozwalające na zrozumienie planów działań i przedsięwzięć. Celem jest tu rozwiązanie określonego problemu lub osiągnięcie pewnego zamierzonego efektu. Procesy biznesowe muszą być względnie łatwe do zdefiniowania oraz pozwalać się łatwo wbudować w strukturę działania. Modelowanie biznesowe powinno umożliwiać zrozumienie procesów organizacyjnych, ale i określać miejsca ich udoskonalania oraz potencjał poprawy. Procesy biznesowe zazwyczaj poddawane są wizualizacji jako rodzaj schematów blokowych aktywności, które muszą być wykonane w danym *modelu biznesowym*. Najbardziej popularną notacją dla modeli biznesowych jest notacja BPMN (*Business Process Modeling Notation*), zaprojektowana przez Business Process Management Initiative (BPMI), jako standard notacyjny w modelowaniu takich procesów. Ogólnie, celem notacji BPMN

³ <http://ipc.informatik.uni-freiburg.de/PddlExtension>.

jest dostarczenie języka, który jest zrozumiały dla wszystkich stron, a więc konsultantów, analityków, managerów, inżynierów informatyków, projektantów, programistów, użytkowników itd. Można już stwierdzić, że BPMN stanowi pewien konsensus dla wszystkich tych stron.

Ważnym oczekiwaniem wobec BPMN jest łatwość i prostota użycia. Modele cechują się jednoznacznością i nie sprawiają problemów interpretacyjnych. Notacja składa się z następujących kategorii elementów:

- obiektów znajdujących się na przepływach: zdarzenia, aktywności, bramki;
- obiektów łączących: przepływów sterowania, przepływów wiadomości, asocjacji;
- torów: elementów umożliwiających grupowanie obiektów;
- artefaktów: obiektów danych, grup, adnotacji.

Szczegółowe przedstawienie informacji na temat notacji BPMN przekraczają zakres tej pracy i można je znaleźć w wielu artykułach, np. [OMG 2009].

Z modelowaniem biznesowym wiąże się często koncepcję tzw. wzorców projektowych, które odgrywają istotną rolę w modelowaniu procesów biznesowych. *Wzorzec* jest „abstrakcją postaci konkretnej formy, która pojawia się w dowolnym momencie” [Riehle 1996]. Przez wzorzec projektowy rozumiemy ogólny opis pewnej struktury aktywności. Jej użycie gwarantuje rozwiązanie pewnego problemu w obszarze modelowania biznesowego. Wszystkie wzorce zostały skatalogowane i zawierają łącznie 23 obiekty [van der Aalst i in. 2003; Russell i in. 2006]. Podstawowe kategorie wzorców to:

- wzorce podstawowe,
- wzorce zaawansowane rozgałęzienia,
- wzorce strukturalne,
- wzorce wieloinstancyjne,
- wzorce stanowe, oraz
- wzorce odwołania.

Dalsze rozważania, jako przykładowe, zostaną ograniczone do pięciu podstawowych wzorców przepływu sterowania:

- przepływ sekwencyjny (*sequence*) – pewna aktywność jest dopuszczona po ukończeniu innej aktywności;
- równoległe rozszczepienie (*parallel split*) – po ukończeniu pewnej aktywności dopuszcza się wykonanie jednocześnie innych aktywności;
- synchronizacja (*synchronization*) – kilka jednocześnie wykonywanych aktywności łączy się w pewnym pojedynczym punkcie i wówczas możliwe jest wykonanie innej aktywności;
- wyłączny wybór (*exclusive choice*) – po ukończeniu pewnej aktywności następuje wybór do wykonania jednej z wielu aktywności;
- proste połączenie (*simple merge*) – kilka alternatywnych aktywności może zostać zakończonych bez synchronizacji, a wówczas może zostać wykonana inna aktywność.

Wyodrębnienie wzorców jest istotne z tego względu, że wpływa korzystnie na proces automatyzacji pozyskiwania specyfikacji logicznej. Automatyzacja tego procesu z kolei jest istotna, gdyż pozwala uniknąć żmudnego i podatnego na błędy procesu budowy olbrzymiej liczby formuł logiki temporalnej składających się na całą specyfikację logiczną modelu. Możliwe jest dokonanie transformacji wzorców projektowych do formuł logiki temporalnej, które z kolei stanowią będą specyfikację logiczną modelu biznesowego, który będzie wykorzystany do wnioskowania metodą tablic semantycznych i w efekcie do weryfikacji poprawności systemu.

4. Weryfikacja modeli biznesowych

Weryfikacja modeli biznesowych odbywa się w sposób formalny, a na stosowany aparat składa się logika temporalna oraz metoda tablic semantycznych jako podstawowa metoda wnioskowania. *Logika temporalna* ma ugruntowaną pozycję w dziedzinie specyfikacji i weryfikacji systemów informatycznych, por. np. [Emerson 1990; Klimek 1999]. Logika temporalna pozwala zapisywać sekwencje i następstwa zdarzeń bez odwoływania się do rachunku kwantyfikatorów. Formuły logiki temporalnej skutecznie wyrażają własności żywotności i bezpieczeństwa systemów.

Przyjęta metoda wnioskowania jest w opozycji do klasycznej metody dedukcyjnej opartej o rezolucję. Metoda jest zwana *metodą tablic semantycznych* i jest stosowana w logice klasycznej, może być także stosowana w przypadku logik temporalnych [D'Agostino i in. 1999], cechuje się apagogicznością i analitycznością.

Ma pewne zalety w stosunku do podejścia tradycyjnego i bazuje na dekompozycji formuł. W każdym kroku dobrze zdefiniowanej procedury postępowania otrzymujemy, poprzez usuwanie spójników logicznych, formuły coraz prostsze i składające się z coraz mniejszej liczby elementów. W efekcie takiego postępowania otrzymujemy drzewo, które jest następnie przeszukiwane pod kątem sprzeczności w występujących zdaniach elementarnych. Jeżeli w gałęzi drzewa znajdują się sprzeczności, to gałąź taką uznaje się za *domkniętą*. Jeżeli wszystkie gałęzie drzewa są domknięte, to oznacza to, że nie istnieje wartościowanie spełniające formułę umieszczoną w korzeniu drzewa. Oznacza to, że formuła przed umieszczeniem w korzeniu drzewa i przed zanegowaniem była prawdziwa. Taka strategia postępowania pozwala na uniknąć budowania wielu formuł połączonych z różnymi kierunkami poszukiwania rozwiązania i szukania dowodu. W metodzie tablic semantycznych poprzez tzw. otwarte gałęzie drzewa wnioskowania otrzymujemy, w przypadku niepowodzenia w dowodzeniu własności, zbiór zdań elementarnych, które nie spełniają własności systemu. Stanowi pewną wartość informacyjną. Metoda tablic semantycznych jest poprawna, a algorytm budowy drzewa zawsze zatrzymuje się. W pracy [Klimek 2012] można znaleźć przykład drzewa wnioskowania metodą tablic semantycznych dla pewnej formuły logiki temporalnej czasu liniowego.

Na rysunku 1 została przedstawiona architektura proponowanego systemu wnioskowania i weryfikacji modeli biznesowych. System składa się z kilku elementów. Najpierw przygotowywany jest model biznesowy (BPMN). Model ten jest następnie skanowany pod kątem wyszukania wszystkich występujących wzorców projektowych. Wzorce znajdujące się w modelu zostają zapisane w postaci *wyrażenia logicznego* W_L (podobnego do znanego wyrażenia regularnego), które zawiera pełną informację o wszystkich, w tym zagnieżdżonych, wzorcach występujących w modelu. Z każdym z nich związany jest predefiniowany zbiór formuł logiki temporalnej $pat(a_1, \dots, a_n) = \{f_1, \dots, f_m\}$ opisujący własności danego wzorca, przy czym a_i stanowią aktywności biznesowe, bądź inne zagnieżdżone wzorce występujące w danym wzorcu, a f_i są formułami LTL nad aktywnościami elementarnymi. Formuły opisują zarówno aspekty żywotnościowe, jak i aspekty bezpieczeństwa każdego wzorca. Predefiniowane formuły logiki temporalnej dla każdego wzorca biznesowego zapisane są w predefiniowanym zbiorze P. Poniżej został pokazany początkowy fragment takiego zbioru, zawierający predefiniowane formuły dla pięciu podstawowych wzorców przepływu sterowania:

```

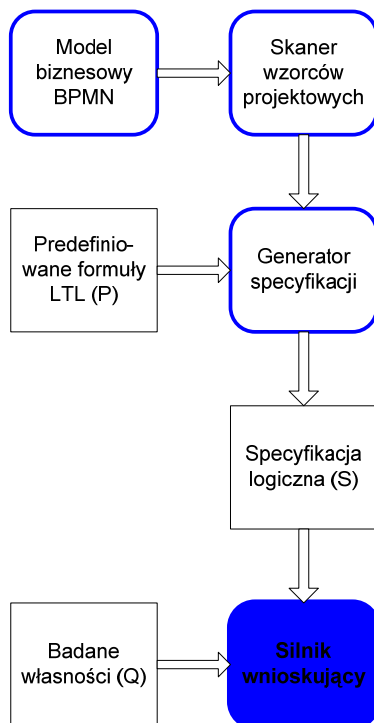
/* version 10.09.2011
/* Basic Control Patterns
Sequence(f1,f2):
f1 =><>f2
Parallel-Split(f1,f2,f3):
f1 =><>f2 &<>f3
[]~(f1&(f2|f3))
Synchronization(f1,f2,f3):
f1 & f2 =><>f3
[]~(f3&(f1|f2))
Exclusive-Choice(f1,f2,f3):
f1 => (<>f2 & ~<>f3)|(~<>f2 &<>f3)
[]~(f2 & f3)
Simple-Merge(f1,f2,f3):
f1|f2 =><>f3
[]~(f3&(f1|f2))
/* ..... [other] Patterns

```

Ostatnim krokiem, przed uruchomieniem silnika wnioskującego, jest zbudowanie *specyfikacji logicznej* L z wyrażenia logicznego W_L , przy czym $L(W_L) = \{f_i : i > 0\}$, gdzie f_i jest dowolną formułą logiki temporalnej. Formuły te są uzyskiwane jako wynik przekształcania wyrażenia logicznego, wraz ze zbiorem predefiniowanym P, do specyfikacji logicznej. Specyfikację logiczną systemu stanowi zbiór formuł logicznych. Po wygenerowaniu specyfikacji możliwe jest już działanie silnika wniosk-

kującego. Podstawową przetwarzaną formułą jest formuła S , gdzie S to specyfikacja systemu, a więc $f_1 \wedge \dots \wedge f_n = S$, a Q to badana własność. Formuła stanowi wejście dla silnika wnioskującego, który realizuje proces dedukcyjny metodą tablic semantycznych:

$$f_1 \wedge \dots \wedge f_n \Rightarrow Q \quad (3)$$



Rys. 1. Architektura systemu wnioskowania

Źródło: opracowanie własne.

Po zanegowaniu formuły 3 umieszczana jest ona w korzeniu drzewa wnioskowania, a następnie poddawana procesowi dekompozycji zgodnie z regułami metody tablic semantycznych. Znalezienie sprzeczności we wszystkich gałęziach drzewa oznacza brak wartościowania spełniającego umieszczoną w korzeniu formułę zanegowaną. To z kolei oznacza, że drzewo jest domknięte, co natychmiast prowadzi do stwierdzenia, że formuła początkowa 3 jest prawdziwa.

Algorytm przekształcenia W_L do L , przy założeniu wartości początkowej $L = \emptyset$ jako zbioru pustego, polega na wyszukaniu w pierwszej kolejności wzorców najbardziej zagnieżdżonych, i o ile argumentami takiego wzorca są aktywności ele-

mentarne, to następuje przepisanie formuł związanych z danym wzorcem do specyfikacji, tj. $L=LUpat()$. Jeżeli jakkolwiek wzorec wyrażenia logicznego zawiera jako argumenty inne wzorce, to wówczas w miejsce argumentu jest podstawiana alternatywa logiczna wszystkich aktywności wewnętrznych. Cały proces jest kontynuowany, począwszy od wzorców położonych w wyrażeniu najbardziej wewnątrz do wzorców znajdujących się bardziej na zewnątrz wyrażenia. Przykładowo, $ParSplit(Seq(a,b),c,d)$ prowadzi w ten sposób do specyfikacji $L=\{a\Rightarrow\Diamond b\}\cup\{(a\vee b)\Rightarrow(\Diamond c\wedge\Diamond d),\Box\neg((a\vee b)\wedge(c\vee d))\}$.

5. Integracja metod planowania i procesów biznesowych

Integracja metod planowania oraz modelowania procesów biznesowych może występować – i w dużej mierze występuje – na wielu poziomach. Najistotniejsze rozgraniczenie dotyczy podziału na:

- próby wykorzystania metod planowania do automatycznej aranżacji procesów biznesowych, oraz
- określenie ram oraz ograniczeń translacji pomiędzy formalizmami używanymi do definiowania problemów planowania oraz modelowania procesów biznesowych, w celu wykorzystania pożądanych cech obu tych podejść do usprawnienia modelowania i weryfikacji szeroko rozumianych problemów.

Kolejne sekcje omawiają szerzej powyższe zagadnienie oraz nakreślają ramy możliwych nowych obszarów badawczych.

5.1. Możliwości translacji pomiędzy formalizmami i metodami reprezentacji

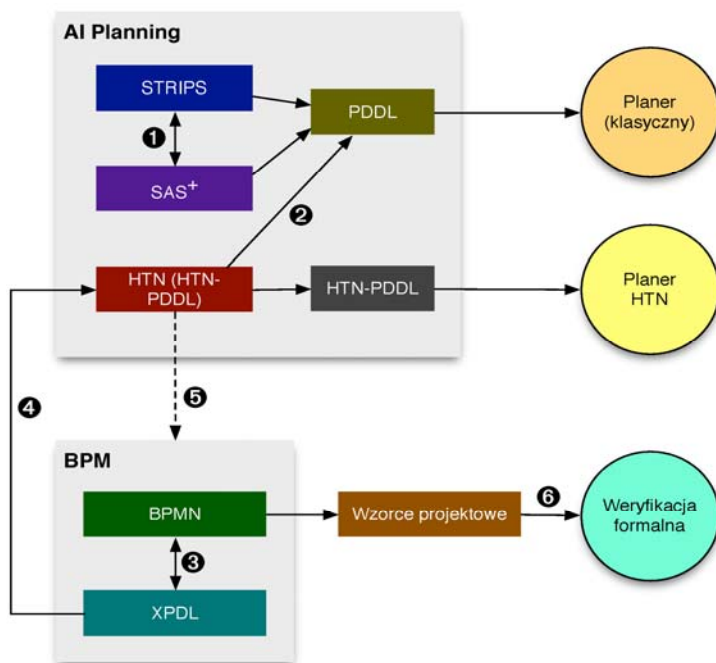
Na rysunku 2 przedstawiono najważniejsze elementy składowe dwóch głównych dziedzin stanowiących przedmiot analizy oraz oznaczono związki i możliwości translacji pomiędzy nimi.

Translacja pomiędzy formalizmami STRIPS i SAS⁺, oznaczona na rys. 2 numerem 1, jest możliwa w obie strony, gdyż ich siła wyrazu jest porównywalna [Bäckström 1995]. W przypadku translacji ze STRIPS na SAS⁺ konieczna jest identyfikacja stanów odpowiadających różnym wartościom danej zmiennej⁴. Bezpośrednia, automatyczna translacja pomiędzy wymienionymi formalizmami a HTN nie miałaby sensu, gdyż wymaga ona dodania zadań złożonych, co musi być przeprowadzone przez eksperta.

Istnieje możliwość wyrażenia problemów planowania HTN w języku PDDL, pod warunkiem spełnienia pewnych ograniczeń. Zaletą takiej translacji, oznaczonej na rys. 2 numerem 2, jest możliwość wykorzystania „zwykłych” planerów do roz-

⁴ Między innymi Keyder i Bonet [2009] proponują podejście oparte na zbudowaniu grafu o węzłach odpowiadającym faktom STRIPS i krawędziach pomiędzy każdą parą faktów, które nie mogą wystąpić w jednym stanie. Kliki w grafie odpowiadają wtedy zmiennym wielowartościowym.

wiązywania problemów HTN. Wymagania, które muszą zostać spełnione, opisano szczegółowo w [Alford i in. 2009, sekcja 3, s. 1630].



Rys. 2. Związki pomiędzy najważniejszymi zagadnieniami składowymi omawianych dziedzin

Źródło: opracowanie własne.

Standard BPMN [OMG 2009], stanowiący klarowne i przenośne narzędzie modelowania procesów biznesowych, oraz język XPDL [WfMC 2008], będący standardem definiowania i wykonywania procesów biznesowych, powstawały niezależnie od siebie. Organizacje stojące za nimi dokładają jednak od lat starań, aby umożliwić pełną translację pomiędzy nimi [White 2003]. Większość narzędzi do wizualnego modelowania procesów biznesowych pozwala na odczyt i zapis modeli zapisanych w języku XPDL. Możliwość tę oznaczono na rys. 2 numerem 3.

W sekcji 4 opisano możliwości formalnej weryfikacji modeli biznesowych metodą wnioskowania dedukcyjnego, poprzez przekształcenie wzorców projektowych na formuły logiki temporalnej. Proces weryfikacji oznaczono na rys. 2 numerem 6.

5.2. Translacja modeli biznesowych na dziedziny i problemy planowania

Relacje pomiędzy szeroko rozumianymi metodami sztucznej inteligencji a modelowaniem procesów zauważono już w [Myers i Berry 1998]. Dopiero jednak

[González-Ferrer i in. 2008] opisują propozycję translacji z języka XPDL do HTN-PDDL, oznaczoną na rys. 2 numerem 4. Jako motywację autorzy podają z jednej strony brak wygodnych narzędzi modelowania dziedzin planowania, z drugiej zaś – ograniczone możliwości wykorzystania metod sztucznej inteligencji do automatyzacji planowania biznesowego. Translacja dokonuje się w oparciu o następujące założenia:

- czynności modelowane jako akcje z czasami trwania,
- wzorce projektowe i podprocesy modelowane jako zadania złożone (zob. sekcja 2.1.3).

Artykuł przedstawia mapowanie dla kilku podstawowych wzorców projektowych (zob. sekcja 3). W [González-Ferrer i in. 2011] przedstawiono rozszerzenie koncepcji oraz stworzone narzędzie wykorzystane do przeprowadzenia eksperymentów. Ramowy algorytm konwersji modeli procesów na dziedzinę i problem planowania HTN proponowany przez autorów składa się z następujących kroków:

1. Wyodrębnienie następujących elementów z modelu: uczestnicy (*participants*), parametry (*parameters*), przejścia (*transitions*), czynności (*activities*), zbioru czynności (*activity sets*) oraz tory (*lanes*).

2. Połączenie wyodrębnionych elementów w graf.

3. Detekcja bloków, obejmująca: wykrywanie podziałów i złączeń, wykrywanie wzorców projektowych, utworzenie modelu drzewiastego [Bae i in. 2004].

4. Translacja modelu drzewiastego na kod HTN-PDDL.

Poszczególne elementy zidentyfikowane w 1. kroku powyższego algorytmu mapowane są na odpowiednie elementy dziedziny i problemu planowania, z wykorzystaniem określonego przez autorów nazewnictwa.

Autorzy stworzyli zestaw narzędzi JABBAH [González-Ferrer i in. 2009], implementujący opisaną powyżej metodę. Wskazuje to na dojrzałość koncepcji i pozwala na praktyczną weryfikację jej możliwości.

Jednocześnie powstała dziedzina i problem planowania osadzone są w określonych „ramach” – wykorzystują z góry określone konstrukcje strukturalne oraz nomenklaturę – co nie stwarza kłopotu z punktu widzenia automatycznego rozwiązywania takich problemów, ale stanowi trudność w przypadku próby dokonania translacji odwrotnej, której koncepcję opisano w sekcji 5.3.

5.3. Możliwości translacji dziedzin planowania na modele biznesowe

Translacja dziedzin planowania na modele biznesowe nie była dotychczas szeroko podejmowana w literaturze. Może leżeć za tym kilka przyczyn. Po pierwsze, modelowanie biznesowe poprzez tworzenie dziedzin i problemów planowania nie ma sensu, gdyż narzędzia wizualne i języki (np. BPMN) wykorzystywane do tego pierwszego są zdecydowanie bardziej rozbudowane niż narzędzia modelowania dziedzin planowania. Po drugie, dziedziny planowania mogą operować na dowolnym zbiorze zmiennych, predykatów i akcji, co może utrudnić ich mapowanie na

dokładnie określone klasy elementów występujące w modelach BPMN. W końcu, problemy planowania wyrażone w języku PDDL (lub językach pokrewnych) mogą być automatycznie rozwiązywane przez planery, podczas gdy problem wykonywania procesów biznesowych jest bardziej złożony.

Z drugiej strony, istnieją przesłanki stojące za przeprowadzeniem próby takiej translacji:

1. Planowanie automatyczne rzadko jest wykorzystywane w kontekście biznesowym. Jednym z powodów takiej sytuacji może być fakt zaznaczonego już braku wygodnych narzędzi modelowania dziedzin planowania, a co za tym idzie – brak możliwości klarownej wizualizacji modelowanego problemu. Możliwość nawet półautomatycznej translacji mogłaby pozwolić co najmniej na wizualizację problemów planowania za pomocą narzędzi takich jak BPMN.

2. Identyfikacja wzorców projektowych w dziedzinach planowania i późniejsze uwzględnienie ich w modelu biznesowym może pozwolić na ich automatyczną weryfikację (zob. sekcja 4).

Translacja taka powinna opierać się na następujących zasadach:

- ze względu na dowolność doboru zmiennych i predykatów może być konieczne wykonanie wstępnego mapowania ich na elementy modelu biznesowego przez eksperta,
- wzorce projektowe powinny być możliwie wcześnie wyodrębnione i, w miarę możliwości, oznaczone, aby uniknąć potrzeby ich ponownej detekcji w celu weryfikacji.

Prowadzone obecnie badania mają na celu weryfikację możliwości dokonania opisywanej translacji i doprecyzowanie przedstawionych ograniczeń oraz zaleceń.

6. Podsumowanie

W artykule przedstawiono nowoczesne problemy związane z modelowaniem biznesowym, automatycznym planowaniem, możliwościami translacji pomiędzy poszczególnymi notacjami oraz formalnymi metodami wnioskowania o poprawności procesów biznesowych metodą dedukcyjną. Translacja procesów planowania do notacji BPMN umożliwia formalną weryfikację planów. Dalsze prace powinny obejmować problem wyodrębniania w otrzymanym modelu biznesowym wzorców projektowych, co z kolei będzie korzystnie wpływać na proces automatycznej generacji specyfikacji logicznej, niezbędnej w procesie dedukcyjnym.

Literatura

- Alford R., Kuter U., Nau D. [2009], *Translating HTNs to PDDL: A small amount of domain knowledge can go a long way*, [in:] IJCAI'09, s. 1629–1634.
- Bäckström C. [1995], *Expressive equivalence of planning formalisms*, "Artificial Intelligence", 76(1–2), s. 17–34.

- Bae J., Bae H., Kang S.H., Kim Y. [2004], *Automatic control of workflow processes using ECA rules*, "IEEE Transactions on Knowledge and Data Engineering", 16(8), s. 1010–1023.
- Bonet B., Geffner H. [1999], *Planning as Heuristic Search: New Results*, [in:] ECP '99 Proceedings of the 5th European Conference on Planning: Recent Advances in AI Planning, vol. 46, s. 368–369.
- Bonet B., Geffner H. [2001], *Planning as heuristic search*, "Artificial Intelligence", 129(1-2), s. 5–33.
- Bylander T. [1994], *The computational complexity of propositional STRIPS planning*, "Artificial Intelligence", 69(1-2), s. 165–204.
- D'Agostino M., Gabbay D.M., H.R.P.J.e. [1999], *Handbook of Tableau Methods*, Kluwer Academic Publishers.
- Emerson E. [1990], *Handbook of Theoretical Computer Science*, vol. B: *Formal Models and Semantics*, chapter Temporal and Modal Logic, Elsevier, MIT Press, s. 995–1072.
- Erol K., Hendler J., Nau D.S. [1994], *HTN Planning: Complexity and Expressivity*, [in:] *AAAI-94*, s. 1123–1128.
- Fikes R.E., Nilsson N.J. [1971], *STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving*, "Artificial Intelligence", 2(3-4), s. 189–208.
- Ghallab M., Nau D., Traverso P. [2004], *Automated Planning: Theory & Practice*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- González-Ferrer A., Fernández-Olivares J., Castillo L. [2009], *JABBAH: A Java Application Framework for the Translation Between Business Process Models and HTN*, [in:] International Competition on Knowledge Engineering for Planning and Scheduling.
- González-Ferrer A., Fernández-Olivares J., Castillo L., Morales L. [2008], *Towards the Use of XPDL as Planning and Scheduling Modeling Tool: the Workflow Patterns Approach*, *Advances in Artificial Intelligence—IBERAMIA 2008*, s. 52–61.
- González-Ferrer A., Fernández-Olivares J., Castillo L. [2011], *From Business Process Models to Hierarchical Task Network Planning Domains*. Knowledge Engineering Review Accepted for publication in October 2010.
- Haslum P., Bonet B., Geffner H. [2005], *New admissible heuristics for domain-independent planning*, [in:] *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, page 1163. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Helmert M., Domshlak C. [2009], *Landmarks, critical paths and abstractions: What's the difference anyway*, [in:] *Proc. ICAPS*, vol. 9.
- Keyder E., Bonet B. [2009], *Tutorial: Heuristics for Planning*, [in:] *ICAPS 2009*, vol. 129.
- Klimek R. [1999], *Wprowadzenie do logiki temporalnej*, Wydawnictwa Naukowo-Techniczne AGH, Kraków.
- Klimek R. [2012], *Towards formal and deduction-based analysis of business models for soa processes*, 4th International Conference on Agents and Artificial Intelligence, Vilamoura, Algrave, 6–8 February, 2012 [praca przyjęta].
- Lekavy M., Návrat P. [2007], *Expressivity of STRIPS-like and HTN-like planning*, "Agent and Multi-Agent Systems: Technologies and Applications", s. 121–130.
- McDermott D., Ghallab M., Howe A., Knoblock C., Ram A., Valoso M., Weld D., Wilkins D. [1998], *PDDL: The Planning Domain Definition Language, Version 1.2*. Technical report, AIPS-98 Planning Competition Committee.
- Myers K., Berry P. [1998], *Workflow Management Systems: An AI Perspective. Technical report*. Artificial Intelligence Center, SRI International, Menlo Park.
- OMG [2009], *Business Process Modeling Notation Specification. Version 1.2*.
- Richter S., Westphal M. [2010], *The LAMA planner: Guiding cost-based anytime planning with landmarks*, "Journal of Artificial Intelligence Research", 39, s. 127–177.
- Riehle D., Z.H. [1996], *Understanding and using patterns in software development*, "Theory and Practice of Object Systems", 2(1), s. 3–13.

- Russell S.J., Norvig P. [2010], *Artificial Intelligence: A Modern Approach*, Third Edition, Pearson Education, Upper Saddle River, New Jersey.
- Russell N., ter Hofstede A.H.M., v. d. A. W. M. N. [2006], *Workflow control-flow patterns: A revised view*. Technical report, BPM Center Report BPM-06-22, BPMcenter.org.
- Sandewall E., Rönnquist R. [1986], *A Representation of Action Structures*, [in:] *AAAI-86*. Department of Computer and Information Science, Linköping University.
- van der Aalst W.M.P., ter Hofstede A.H.M., K. B. B. A. [2003], *Workflow patterns*, “Distributed and Parallel Databases”, 4(1), s. 5–51.
- WfMC (2008). XPD L 2.1 Complete Specification.
- White S.A. [2003], *XPD L and BPMN*, [in:] *Workflow Handbook 2003*, L. Fischer (ed). http://www.omg.org/bpmn/Documents/XPD L_BPMN.pdf.

FORMAL VERIFICATION OF PLANNING PROBLEMS IN THE BUSINESS PROCESS CONTEXT. ANALYSIS OF FEASIBILITY

Summary: The paper presents the relations between business process modeling (BPM), service-oriented architecture (SOA) and AI planning methods, including such issues as planning problem domain modelling using business modelling tools or automatic service composition using planning methods. A new research direction, related to formal verification of planning domains using deductive reasoning, based on workflow patterns (van der Aalst et al.), is presented. Formal verification is implemented using deductive reasoning with semantic tables. The properties of the model are described using linear time temporal logic.

Keywords: business modeling, planning, temporal logic, formal verification, semantic tableaux method.