УДК 004.822:514

ИНТЕГРАЦИЯ ЛОГИКИ ДВИЖЕНИЯ И БИНАРНОЙ МОДЕЛИ ДАННЫХ И ЗНАНИЙ (INTEGRATION OF THE LOGIC OF MOTION AND BINARY DATA AND KNOWLEDGE MODEL)

Горкина А.А.

Московский энергетический институт (Национальный исследовательский университет), г.Москва, Россия

agorkina@mail.ru

В статье рассмотрены логика движения и бинарная модель знаний. Показано, как можно объединить их в одну систему. Благодаря этой интеграции имеется возможность адекватно описывать в одной модели структуру движущихся объектов и в то же время описывать свойства их движения, а также прогнозировать дальнейшее их поведение.

Ключевые слова: логика движения, модели данных, модели знаний, бинарная модель знаний.

Введение

В теории интеллектуальных информационных систем имеются многочисленные работы, в которых рассматривается движение объектов во времени и пространстве.

В работе [Yaman et al., 2004] была введена так называемая логика движения, которая соединяет понятия Ньютоновой механики и логики. В дальнейшем эта логика была развита в работах [Yaman et al., 2005a], [Yaman et al., 2005b], [Yaman et al.,. 2005c] и [Parker et al., 2007]..

В логике движений можно рассуждать о планах для движущихся объектах, в частности, предсказывать их дальнейшее расположение. Такие рассуждения важны для многих приложений (управление воздушным трафиком, планирование тактических военных задач и т.п.).

В логике движений объекты не имеют структуры, т.е. фактически являются материальными точками. Однако, во многих задачах требуется представление структуры объектов и необходимо манипулировать этой структурой. Обычно структура объекта определяется его связями с другими объектами, его компонентами и атрибутам. Такими возможностями обладают языки фреймового типа.

В работах [Plesniewicz, 2004] и [Плесневич, 2005] была введена Бинарная Модель Данных и Знаний (БМДЗ), которая представляет собой систему концептуальных (т.е. ориентированных на представление понятий) языков фреймового типа. В БМДЗ естественным образом можно моделировать структуру объектов, рассуждать о структурированных объектах и вычислять ответы на запросы, относящиеся к данным, представляющих эти объекты.

В этой статье мы кратко опишем способ интеграции логики движений и БМДЗ. Благодаря этой интеграции имеется возможность адекватно описывать в одной модели структуру объектов и, в то же время, описывать свойства движения объектов.

1. Краткие сведения о логике движений

Пусть R – множество всех вещественных чисел, R^+ – множество положительных чисел, O – конечное множество имен объектов, V_R – множество переменных со значениями в R и V_O – множество переменных со значениями в O. *Вещественный терм* – это элемент $\mathsf{R} \cup \mathsf{V}_\mathsf{R}$, *объектный терм* – это элемент $\mathsf{O} \cup \mathsf{V}_\mathsf{O}$.

Атомы логики движений определяются следующим образом:

- Если o_1 и o_2 – объектные термы, d, t_1, t_2 – положительные вещественные термы, то $\mathbf{near}(o_1,o_2,d,t_1,t_2)$ – атом. Интуитивный смысл: «объекты o_1 и o_2 находятся друг от друга на расстоянии d в моменты времени из интервала $[t_1,t_2]$»;

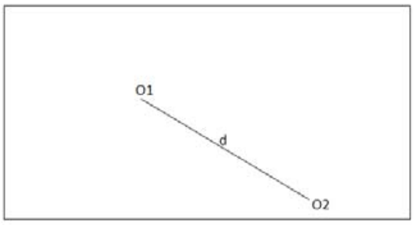

Рисунок 1. Графическая иллюстрация значения near-атома

- Если o_1 и o_2 – объектные термы, d, t_1, t_2 – положительные вещественные термы, то **far**(o_1,o_2,d,t_1,t_2) – атом. Интуитивный смысл: «расстояние между объектами o_1 и o_2 в моменты времени из интервала $[t_1,t_2]$ больше d»;

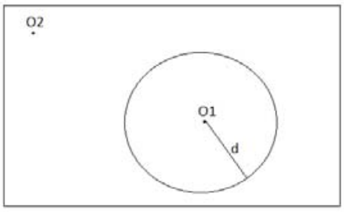

Рисунок 2. Графическая иллюстрация значения far-атома

- Если o – объектный терм, t_1 , t_2 – положительные вещественные термы, x_1, y_1, x_2, y_2 – вещественные термы, то **in**$(o,x_1,y_1,x_2,y_2,t_1,t_2)$ – атом. Интуитивный смысл: «объект o находится в некоторый момент времени из интервала $[t_1,t_2]$ в прямоугольнике с левой нижней вершиной (x_1,y_1) и правой верхней вершиной (x_2,y_2)»;

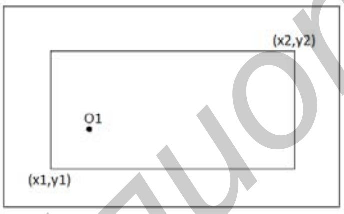

Рисунок 3. Графическая иллюстрация значения in-атома

- Если o – объектный терм, t_1^-, t_1^+, t_2^-, t_2^+, v^-, v^+, – положительные вещественные термы, x_1, y_1, x_2, $_2$ – вещественные термы, то **go**$(o,x_1,y_1,x_2,y_2, t_1^-,t_1^+,t_2^-, t_2^+,v^-,v^+)$ – атом. Интуитивный смысл: «объект o покидает точку (x_1,y_1) в некоторый момент времени из $[t_1^-,t_1^+]$, движется по прямой с некоторой скоростью из интервала $[v^-,v^+]$ и прибывает в точку (x_2,y_2) в некоторый момент времени из $[t_2^-,t_2^+]$».

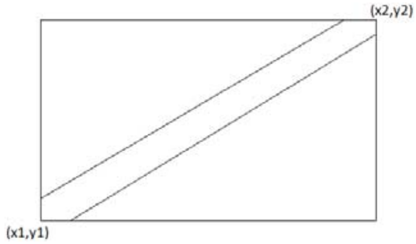

Рисунок 4. Графическая иллюстрация значения go-атома

Предложение логики движения – это атом или булева комбинация атомов, т.е. если α и β – предложения, то предложениями являются ~α, α/\β и α\/β. Выражение α→β также считается предложением и рассматривается как сокращение предложения ~α\/β.

Семантика логики движений определяется через понятие интерпретации. *Интерпретация* – это непрерывная функция

I: $O \times R^+ \rightarrow R \times R$. Интуитивный смысл: «$I(o,t)$ – положение объекта o в момент времени t».

Атом **go**$(o,x_1,y_1,x_2,y_2, t_1^-,t_1^+,t_2^-,t_2^+,v^-,v^+)$ *истинен* в интерпретации I *относительно* интервала $[t_1,t_2]$, если:

- $t_1 \in [t_1^-,t_1^+]$, $t_1 \in [t_1^-,t_1^+]$ и $I(o, t_1) = (x_1,y_1)$, $I(o, t_2) = (x_2,y_2)$;
- для всякого момента времени $t \in [t_1,t_2]$ точка $I(o,t)$ лежит на отрезке $[(x_1,y_1), (x_1,y_1)]$;
- для всех $t, t' \in [t_1,t_2]$, если $t < t'$, то имеет место неравенство $dist(I(o,t),(x_1,y_1)) < dist(I(o,t'),(x_1,y_1))$, где dist обозначает эвклидово расстояние между двумя точками;
- почти всегда в интервале $[t_1,t_2]$, т.е. кроме конечного числа моментов времени t, существует производная $v = d(I(o,t)) /dt$ и $v \in [v_1,v_2]$.

Атомы **near**(o_1,o_2,d,t_1,t_2), **far**(o_1,o_2,d,t_1,t_2), **in**$(o,x_1,y_1,x_2,y_2,t_1,t_2)$ *истинны* в интерпретации I, если соответственно:

- $dist(I(o_1,t),I(o_2,t)) \leq d$ для всякого $t \in [t_1,t_2]$;
- $dist(I(o_1,t),I(o_2,t)) > d$ для всякого $t \in [t_1,t_2]$;
- существуют $t \in [t_1,t_2]$, $x \in [x_1,x_2]$, $y \in [y_1,y_2]$ такие, что $I(o_1,t)= (x,y)$.

Атом **go**$(o,x_1,y_1,x_2,y_2, t_1^-,t_1^+,t_2^-,t_2^+,v^-,v^+)$ *истинен* в интерпретации I, если существует интервал времени $[t_1,t_2]$, что он истинен относительно этого интервала.

Истинность предложений определяется стандартным образом. Если α и β – произвольные предложения, то:

- I(~α) = 1 (истина) ⇔ I(α) = 0 (ложь);
- I(α/\β) = 1 ⇔ I(α) = 1 и I(β) = 1;
- I(α\/β) = 1 ⇔ I(α) = 1 или I(β) = 1;
- I(α\/β) = 1 ⇔ I(α) = 0 или I(β) = 1.

Как и во всякой логике, в логике движения имеются понятия логического следствия и логической выполнимости. Пусть E – множество предложений логики движения и α – предложение.

Тогда из E *логически следует* α (в записи: $E \models \alpha$), если не существует интерпретации, при которой все предложения из E истинны, а предложение α ложно. Множество E *выполнимо*, если существует интерпретация, при которой все предложения из E истинны.

Проблема логического следствия сводится к проблеме невыполнимости, так как ясно, что $E \models \alpha$ $\Leftrightarrow$ множество $E \cup \{\sim\alpha\}$ невыполнимо. На самом деле, проблему логического следствия (для конечных E) можно свести к проблеме невыполнимости литералов (т.е. атомов или отрицаний атомов). Для этого можно воспользоваться методом аналитических таблиц. Рассмотрим это на примере.

Пример 1.

Пусть $E = \{\sim at_1 \wedge (at_2 \rightarrow at_3),\ at_2 \rightarrow at_1\}$ и $\alpha = at_3 \vee at_1$. На рисунке 5 показано дерево вывода по методу аналитических таблиц для исходного множества $E \cup \{\sim\alpha\} = \{\sim at_1 \wedge (at_2 \rightarrow at_3),\ at_2 \rightarrow at_1,\ \sim(at_3 \vee at_1)\}$. Это дерево имеет три ветви, из которых только одна ветвь открыта. Открытая ветвь содержит множество литералов $F = \{\sim at_1, \sim at_2, \sim at_3\}$. Таким образом, имеет место логическое следствие $E \models \alpha$ тогда и только тогда, когда множество F невыполнимо.

Ф. Яман разработала алгоритмы для выяснения выполнимости конечных множеств атомов логики движения.

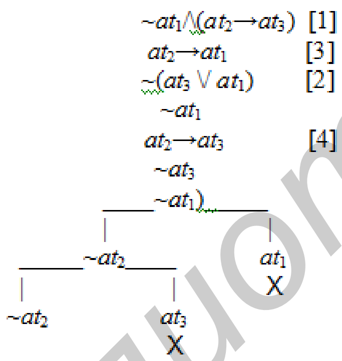

Рисунок 5. Дерево вывода

База знаний в логике движений – это конечное множество предложений этой логики. *Запросом* назовем выражение вида ?α, где α – предложение логики движений.

Пусть E – база знаний и ?α – запрос к этой базе. *Ответом* на этот запрос может быть одно из сообщений: «да». «нет» или «нет информации». Алгоритмы, выясняющие логические следствия, можно использовать для вычисления ответов на запросы к базе знаний: ответ на запрос ?α есть:

- «да», если $E \models \alpha$;
- «нет», если $E \models \sim\alpha$;
- «нет информации», если $E \not\models \alpha$ и $E \not\models \sim\alpha$.

2. Краткие сведения о бинарной модели данных и знаний

Семантическими примитивами для концептуальных языков служат формальные понятия, смысл которых близок к смыслу понятий класса или типа объектов.

В БМДЗ входит язык структурной спецификации (ЯСС), с помощью которого определяется структура объектов – примеров понятия. В ЯСС можно определять схемы, экземплярами которых служат состояния баз данных.

Мы рассматриваем два вида понятий – классы и бинарные связи (т.е. бинарные отношения между классами). Имена классов бывают простыми или составными. Составные имена классов или бинарных связей обозначают классы или бинарные связи, полученные при помощи соответствующих операций. Используются следующие операции над понятиями:

- C(*) обозначает класс, экземплярами которых служат конечные множества экземпляров понятия C
- C(p,q), где p, q – натуральные числа и p ≤ q , обозначает понятие, экземплярами которого служат конечные множества экземпляров понятия (класса или бинарной связи) C, причем размеры этих множеств заключены в пределах чисел p и q. Число q можно заменить на символ *, который означает неограниченность размера множеств – экземпляров класса. Вместо C(1,1) пишем просто C. Кроме того, вводятся следующие сокращения: (*), (+), (p), (≤p), (p) соответственно для (0,*), (1,*), (p,p), (0,p), (p,*);
- L((p,q),(r,s)), где p ≤ q и r ≤ s, обозначает бинарную связь, экземплярами которой служат конечные множества экземпляров L. При этом, если множество F L((p,q),(r,s)) рассматривать как двудольный граф, то степень любой левой (правой) вершины должна быть заключена между числами p и q (соответственно, между числами r и s). Для пар (p,q) и (r,s) применяются также вышеупомянутые сокращения;
- C1 | C2 |…| Cn обозначает класс (бинарную связь), экземплярами которого служат все экземпляры классов (соответственно, бинарных связей) Cj.

Произвольные предложения ЯСС составляется из примитивных предложений, которые имеют следующие формы:

C[D], C[A:D], C[A:T], (C L D), (C L D)[E], (C L D)[A:D], (C L D)[A:T],

где C, D, E – имена классов, L – имя бинарной связи, A – имя атрибута и T – имя типа данных. Типы данных могут быть примитивными или составными. (Фактически, составной тип данных можно рассматривать как абстрактный тип данных, если для него задать операции.)

Примитивные предложения имеют в БМДЗ естественную денотативную семантику. Например, если декларируется предложение (C L D)[A:E], то в универсум U^L для бинарной связи L включаются имена вида [C:x,D:y,A:z], где x, y, z – произвольные идентификаторы объектов (суррогаты). (Универсум понятия – это множество всех имен для возможных экземпляров понятия.)

Произвольные предложения ЯСС получаются из примитивных путем их соединения способом, который можно понять на примере.

Пример 2.

Самолет[№Борта:Integer, Тип:{пассаж, груз, воен}, Марка:String, СрокЭкспл:Integer],

Полет[Самолет, АэропортВылета: Аэропорт, Расписание: (Отпр:Time, Прилет:(Time | {?})), ОбъемГорюч(T):(Integer | String), Локация(T):(Real,Real)], T IN (Time | {now, before}), Аэропорт[Назв:String, Индекс:Char(3), Зона: Rectangle], Rectangle = (Ll: (Real,Real), Rt: (Real,Real)).

БМДЗ содержит язык запросов (ЯЗ). Дадим пример простого запроса к схеме примера 2.

Пример 3.

Запрос1: «Найти номера бортов марки Boeing 747, выполняющих полеты из аэропорта Домодедово».

?X.№Борта–Полет(Самолет=X; АэропортВылета = Домодедово; X.Марка = ‘Boeing 747’).

3. Интеграция логики движения и бинарной модели данных и знаний

При интеграции в предложения ЯЗ включаются атомы логики движений как методы (параметрические атрибуты). Рассмотрим примеры запросов, содержащие обращения к базе данных схемы из примера 2.

Пример 4.

Запрос 2:

- «Найти все пассажирские самолеты марки Boeing 747, находящиеся близко к зоне аэропорта Домодедова и имеющие в настоящий момент мало горючего».

X.№Борта – Полет(Самолет = X;
ОбъемГорюч(now)= мало;
Локация(now); Near Z);
Аэропорт(Назв=Домодедово; Зона= Z).

Запрос 3:

- «Существует ли небезопасный полет в данном прямоугольнике?». Полет считается небезопасным, если самолет эксплуатируются без ремонта более трех лет или имеется другой самолет в том же прямоугольнике.

?X – X IN НебезопПолет;
НебезопПолет= Полет(Самолет.ВремяЭксп > 3
OR X IN Полет(Самолет = Y);
Самолет In (Ll: a, Rt: b);Y In (Ll: c, Rt: d))).

При построении интегрированной системы используются два способа интеграции: интеграция с помощью интерфейсов и логическая интеграция.

3.1. Интеграция с помощью интерфейсов

В нашем случае использование интерфейсов помогает нам соединить модели логики движения и бинарной модели знаний в оптимальном для них виде.

Соответственно, блоки реализации логики движения, а именно задание и проверка на выполнимость атомов go, in, near все так же необходимы, но их внешний интерфейс, как то, что будет показано пользователю, несколько видоизменяется.

Во-первых, каждый из атомов будет внутренне, на уровне бинарной модели знаний, определен с помощью кортежа и линейного конструктора LLIST. Расширенные функции манипуляции с бинарной моделью знаний позволяют так же разбирать полученные кортежи на составляющие, а, значит, и работать с более глубокими знаниями, чем те, которыми изначально позволяет манипулировать логика движения. Для этого будет необходимо использовать деконструкторы кортежей.

В чем преимущества такого подхода? Во-первых, упрощается анализ для различных объектов движения. Во-вторых, на много легче становится добавление размеров материальной точке, которую мы сейчас рассматриваем в наших расчетах. Так же приведение оси к объемному пространству намного упрощается.

3.2. Логическая интеграция.

Логическая интеграция схожа с интерфейсной интеграцией, но требует дополнительного задания связывания доступа к go-атомам через бинарную модель знаний. В частности, необходимо создать конструкторы для сборки каждого атома через бинарную модель. Легче всего это сделать с помощью конструкторов и деконструкторов, указанных в предыдущем способе.

Выводы

В данной работе были проанализированы логика движения, дающая возможность расчета полосы движения в реальном времени, бинарная модель знаний, а так же сделана их интеграция.

Этап интеграции очень важен, так как базовая логика движения без вспомогательных функций несколько громоздка и трудноупотребима. Использование бинарной модели знаний позволяет облегчить как внешнее общение с реализацией логики движения, а так же дальнейший анализ и расширение логики движения.

При интеграции бинарной модели знаний с логикой движения, упрощается не только графический интерфейс работы с программой, но также возникают преимущества самой получившейся системы перед начальной. Первым преимуществом является то, что упрощается поиск по объектам. Вторым преимуществом является возможность более легкого переведения теории с плоскости на пространство.

Таким образом, интеграция бинарной модели знаний и логики движений – оправдана и полезна

БЛАГОДАРНОСТЬ

Автор благодарит Российский Фонд Фундаментальных Исследований за финансовую поддержку работы, выполненной по проекту №11-01-00538.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Parker et al., 2007] A. Parker, F. Yaman, D. Nau and V.S. Subrahmanian.] Probabilistic Go Theories // Proc. 2007 International Joint Conference on Artificial Intelligence (IJCAI-2007), Hyderabad, India , Jan 6-12, 2007.

[Yaman et al., 2004] F. Yaman, D. Nau,V.S. Subrahmanian. The Logic of Motion // Proc. 2004 International Conference on the Principles of Knowledge Representation and Reasoning (KR-2004), Whistler, Canada, June 2-5, 2004.

[Yaman et al., 2005a] F. Yaman, D. Nau and V.S. Subrahmanian. A Motion Closed World Assumption // Proc. 2005 International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, Aug 2-5, 2005.

[Yaman et al., 2005b] F. Yaman, D. Nau, V.S. Subrahmanian. Going Far, Logically // Proc. 2005 International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, Aug 2-5, 2005.

[Yaman et al., 2005c] F. Yaman, D. Nau and V.S. Subrahmanian. A Motion Closed World Assumption // Proc. 2005 International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, Aug 2-5, 2005.

[Plesniewicz, 2004] G.S. Plesniewicz. Binary Data and Knowledge Model // In: Stefanuk, Kaijiri (eds.) Knowledge-based software engineering. – IOS, 2004.

[Плесневич, 2005] Г.С. Плесневич. Бинарные модели знаний // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов III-го Международного научно-практического семинара (Коломна, 15-17 мая 2005 г.). – М.: Физматлит, 2005.

INTEGRATION OF THE LOGIC OF MOTION AND BINARY DATA AND KNOWLEDGE MODEL

Gorkina A.A. *

* *National Research University "Moscow Power Engineering Institute", Moscow, Russia*

agorkina@mail.ru

The article considers the logic of the movement and the binary model of knowledge. It is shown how to combine them into one system. Through this integration there is a possibility to adequately describe in a model of the structure of moving objects and at the same time describe the properties of their movement, as well as predict future behavior.

INTRODUCTION

In the theory of intelligent information systems, there are a lot of works in that it is discussed the motion of objects in space and time.

In the logic of movements it can be discussed about the plans for moving objects, in particular, to predict their future location. These considerations are important for many applications (air traffic management, planning of tactical military objectives, etc.).

In the logic of motion objects have no structure, that means that in fact are material points. But many tasks require the representation of the object structure and to manipulate with this structure. Typically, the object structure is defined by its relationships with other objects, their components and attributes. These capabilities have languages of framing type.

In works [Plesniewicz, 2004] and [Plesnevich, 2005] was idefined Binary Data and Knowledge Model (BDKM), which is a conceptual system (that means, presentation-oriented concepts) languages of framing type. In BDKM naturally can represent the structure of objects and reason about structured objects and calculate answers to queries relating to data representing the objects.

In this article, we briefly describe how to integrate logic of motion and BDKM Through this integration it is possible to describe adequately in a model of the structure of objects, and at the same time, to describe the properties of moving objects.

MAIN PART

This article consists of those parts. The first part gives a general information about the logic of the motion and also describes the basic atoms and the rules of representation of logic of motion. The second part gives general information about the binary model of data and knowledge. It describes the general rules of BDKM and gives examples of using of this theory. The third part describes how to integrate them and examples are considered to illustrate the call to atoms after the integration.

The logic of motion consists of three main and one additional atoms. They are defined as follows:

- If O1 and O2 are object terms, d, t1, t2 are positive real terms, then near (O1, O2, d, t1, t2) is atom. The intuitive meaning: "the objects O1 and O2 are separated by a distance d at time in interval of [t1, t2]»;
- If O1 and O2 are object terms, d, t1, t2 are positive real terms, then far (a1, a2, d, t1, t2) is atom. The intuitive meaning: "the distance between objects O1 and O2 at time in the interval [t1, t2] more than d»;
- If a is object term, t1, t2 are positive real terms, x1, y1, x2, y2 are real terms, then in (a, x1, y1, x2, y2, t1, t2) is atom. The intuitive meaning: "the object is at some time in the interval [t1, t2] in the rectangle with lower left vertex (x1, y1) and upper right vertex (x2, y2);
- If o is object term, t1-, t1 +, t2-, t2+, v-, v+ are positive real terms, x1, y1, x2, y2 are real terms, then

Semantic primitives for conceptual languages are formal concepts whose meaning is close to the meaning of the concepts of a class or object type.

In BDKM includes structural specification language (LSN), used to define the structure of objects such as examples of the concept. In LSN can be defined the schema instances which serve as the state databases.

We consider two kinds of concepts: classes and binary relation (that means, binary relations between classes). Class names can be simple or compound. Compound names of the classes or binary relations represent classes or binary communications received by means of the transactions. The following operations are on concepts:

- C(*)means class whose instances are finite sets of instances of concept C
- C(p,q), where p, q are natural numbers and $p \leq q$, means concept whose instances are finite sets of instances of the concept (a class or binary relation) C, and the sizes of these sets are in the range of numbers p and q. The number q can be replaced by the symbol *, which means unlimited size of sets of instances of the class. Together C(1.1) we write simply C. In addition, we introduce the following abbreviations: (*) (+) (p), (≤ p), (p), respectively, for (0,*), (1,*), (p,p), (0,p), (p,*);
- L((p,q),(r,s)), where $p \leq q$ and $r \leq s$means a binary relation, instances of which are finite sets of instances L. In this case, if the set FL ((p, q), (r, s)) viewed as a bipartite graph, then the degree of each left (right) vertex must be enclosed between the numbers p and q (respectively, between numbers r and s). For pairs (p, q) and (r, s) shall be also applyed aforementioned reductions;
- C1 | C2 |…| Cn means class (binary relation), instances of which are all instances of classes (respectively, binary relations) Cj.

Arbitrary proposals LSN compiled from primitive proposals, which have the following forms:

C[D], C[A:D], C[A:T], (C L D), (C L D)[E], (C L D)[A:D], (C L D)[A:T],

where C, D, E are class names, L is the name of a binary relation, A is attribute name and the T is the data type name. Data types can be primitive or composite. (In fact, a composite data type can be considered as an abstract data type, if we can set him the set of operations.)

Primitive sentences in BMDZ have natural denotative semantics. For example, if the offer is declared (with LD) [A: E], then the universe of UL for a binary relation L includes the names of the form [C: x, D: y, A: z], where x, y, z are arbitrary object identifiers (substitutes). (The universe of concepts is the set of names for possible instances of the concept.)

Arbitrary proposals can be obtained from LSN primitive by their connection.

During the integration proposals of LSN include atoms of logic of motion as methods (parametric attributes).

There are two approaches described above, the integration of models at the logical level and using the interface approach. In this work presents both methods.

Integration with Interfaces as basice

In our case, the use of interfaces helps us to connect the logic of model of and the binary model of the movement of data and knowledge in an optimal form for them. Accordingly, a block of logic of motion, namely the task and check on the feasibility of the atoms go, in, near and far still necessary but their external interface is multiplicity modified from the case without integration.

At first, each atom will be internally at the level of the binary model of data and knowledge, is defined by the tuple and constructor of linear lists LLIST. Advanced functions manipulate binary data model and knowledge allows also to disassemble obtained tuples into components and, therefore, to work with deeper data and knowledge than that which was originally to manipulate the logic of the motion. This would involve to use for deconstructors of tuples.

The advantages of this approach are:

First of all, it simplifies the analysis of motion of various objects.

Second, it becomes much easier to add structure to the material points.

Logical integration

Logical integration is similar to the interface integration, but requires an additional set of binding to access go-atoms via binary model of data and knowledge. In particular, it is necessary to create constructor for the assembly of each atom in the binary model. The easiest way to do it with constructors and deconstructors mentioned in the previous method.

CONCLUSION

In this work were analyzed logic of the motion giving an opportunity to calculate the traffic lanes in real-time, binary model of knowledge, as well as their integration was done.

Integration phase is very important, because the basic logic of the motion without auxiliary functions is several cumbersome and difficult to implement. The use of binary model of data and knowledge can facilitate the external communication with the implementation of logic of motion, as well as further analysis and extension of logic of the motion.