

UOT 658.5.012

İNTELLEKTUAL İDARƏETMƏ SİSTEMLƏRİNDƏ BİLİKLƏR BAZASININ GENERASIYASI ÜÇÜN PROQRAM KOMPONENTLƏRİNİN TƏŞKİLİNƏ YANAŞMA KONSEPSİYASI

MANAFOVA XƏYALƏ İDRİS qızı

Sumqayıt Dövlət Universiteti, doktorant

aqil.55@mail.ru

Açar sözlər: proqram komponentlərinin avtomatlaşdırılmış işlənməsi, intellektual sistemlər, konseptual model, biliklər bazası, modellərin çevrilməsi

Hal-hazırda intellektual sistemlərin (İS) və onların proqram komponentlərinin yaradılmasında yeni metod və yanaşmaların işlənməsi elmi tədqiqatların aparılması üçün perspektivli sahələrdəndir. İS-in işlənməsi prosesinin mürəkkəbliyi əsasən İS-in layihələndirilməsində “çətin sahə” sayılan məlumatların alınması, strukturlaşdırılması və formalaşması, tətbiq sahəsinin modelləşdirilməsinə xidmət edən məlumat bazasının (BB) işlənməsi mərhələlərinin xüsusiyyətlərindən asılıdır [1]. Kənardan müraciətə, müştərək (kollektiv) istifadənin təmin olunmasına, ekspert rəylərinin koordinasiyasına, həmçinin konseptual (informasion) modellər (diqram, sxem, bilik kartları və s.) də daxil olmaqla, müxtəlif formalarda təqdim olunan toplanmış məlumatın təkrar istifadəsinə ehtiyac yarandıqda qoyulmuş vəzifələrin mürəkkəbliyi daha da artır.

Beləliklə, İS və onların proqram komponentlərinin (həmçinin BB formalaşmasının altsistemlərinin) işlənməsi effektivliyinin artırılması problemi aktualdır və müxtəlif istiqamətlərdə həll olunur: xüsusən metodologiyanın təkmilləşməsindən başlayaraq xüsusi proqram instrumentarisinə qədər. Lakin bütövlükdə konseptual modellərin transformasiyası əsasında BB-nin avtomatlaşdırılmış işini təmin edən İS proqram komponentlərinin işlənilməsi üçün hazırlanmasını dəstəkləmək üçün texnologiyaların (yanaşmanın) yaradılması probleminin həlli ayrıca tədqiqatların aparılmasını tələb edir [2].

Bununla bağlı işin əsas məqsədi konseptual modellərin transformasiyası əsasında BB-nin formalaşması üçün nəzərdə tutulan proqram konseptlərinin yaradılması prosesinin avtomatlaşdırılmasına dair yanaşmanın işlənilməsi və yanaşmanın veb-istiqamətli proqram sistemi formasında reallaşmasıdır. Nəzərdə tutulur ki verilmiş sistem konseptual modellərin elementlərinin transformasiyası yolu ilə müəyyən biliyi təsvir edən dildə (BTD) BB kodunun sintezini (generasiyasını) təmin etməklə yanaşı, bu prosesin xüsusi vizualizasiya vasitələrindən istifadə etməklə BB-nin müştərək (kollektiv) layihələndirməsini də təmin etməlidir.

Konseptual modellərin mənbəyi kimi strukturu XML (eXtensible Markup Language) vasitəsi ilə təsvir edilən modellər qrupunu istifadə etmək təklif olunur.

Bu modellər qrupuna nümunə kimi bunları göstərmək olar: XMI (XML Metadata Interchange) [3] – modellər haqqında informasiya mübadiləsi standartına uyğun təqdim olunan UML (Unified Modeling language) [4] siniflərinin diaqramları; XTM (XML Topic Maps) [5] standartlarına uyğun təqdim olunan konsept-kartlar və s. məqsədli BTD qismində hal-hazırda məhsulun və ontologiyanın təqdim olunmasının ən geniş yayılmış üsulu sayılan CLIPS (C Language Integrated Production System) [6] və OWL (Web Ontology Language) [5] seçilmişlər.

1.Yanaşma konsepsiyası. Tətbiq sahələrinin konseptual modellərinin transformasiyası yolu ilə BB formalaşmasını təmin edən İS proqram komponentlərinin yaranması prosesinin avtomatlaşdırılması üçün işlənmiş yanaşmanın əsas elementləri:

- transformasiya modelini əhatə edən ümumiləşdirilmiş proqram komponentinin modeli;
- mövzu istiqamətli deklarativ dil – Transformation Model Representation Language (TMRL) transformasiya modelinin təqdim olunması və saxlanması üçün nəzərdə tutulub;

- ümumiləşdirilmiş proqram komponentinin əsasında proqram komponentlərinin yaradılması metodu və onun transformasiya modelinin əsasında sazlanması (xüsusişdirilməsi);
- veb-təyinatlı proqram sisteminin və onun əsas elementlərinin konseptual arxitekturası (quruluşu).

Yanaşmanın proqram təminatı üçün əlavələrin (prilojeniya) veb-istiqamətli işlənmə prinsiplərindən istifadə etmək təklif olunur. Baxılan veb-istiqamətli proqram sisteminin və tərkibinə yeni proqram komponentlərinin əlavə olunması və genişlənmək qabiliyyəti onun özəl xüsusiyyətidir. Bu proqram komponentləri konseptual modellərin fərqli formatlarının analizi və importunu həmçinin müxtəlif BTD-də BB kodunun generasiyasını təmin edir. Yaradılan proqram komponentləri ayrılıqda digər İS üçün müstəqil altsistem qismində istifadə oluna bilər.

Bu sistem həm də xüsusi qarşılıqlı əlaqələr interfeysi təqdim edir. Onun köməyi ilə xarici proqram vasitələri konseptual modellərin importu və onların əsasında BB prototipinin əldə olunması üçün müvafiq proqram komponentlərinə müraciət edə bilər. Aşağıda işlənilib hazırlanmış yanaşmanın ayrı-ayrı elementlərini daha ətraflı araşdıracağıq.

2. Ümumiləşdirilmiş (standart) proqram komponentlərinin modeli. Proqram komponentlərinin işlənməsinin effektivliyini artırmaq üçün aşağıda təqdim olunan ümumiləşdirilmiş proqram komponentinin orijinal modelindən istifadə etmək təklif olunur:

$$M_{TPC} = \langle M_T, A_{IN}, CG_{OUT}, I \rangle \quad (1)$$

burada M_T – transformasiya modeli; A_{IN} – giriş modellərinin analizatoru (pasper); CG_{OUT} – çıxış modellərinin generatoru, I – xarici İS qarşılıqlı əlaqə interfeysidir.

Eyni zamanda: $A_{IN} = \langle A_{IN}^{CM}, A_{IN}^{ONT} \rangle$

burada A_{IN}^{CM} - XML formatında təqdim olunan giriş konseptual modellərin analizatoru; A_{IN}^{ONT} - ontologiya şəklində təqdim olunmuş giriş konseptual modellərin analizatorudur.

$$CG_{OUT} = \langle CG_{OUT}^{ONT}, CG_{OUT}^{CLIPS-KB}, CG_{OUT}^{OWL-KB} \rangle$$

burada CG_{OUT}^{ONT} – ontologiyanın çıxış modelinin generatorudur; $CG_{OUT}^{CLIPS-KB}$ – BTD CLIPS-də BB kodunun generatorudur, CG_{OUT}^{OWL-KB} – BTD OWL-də BB kodunun generatorudur.

Konseptual modellərdən alınmış biliklərin unifikasiya edilmiş şəkildə təqdim olunması və saxlanması üçün xüsusi işlənilib hazırlanmış ontologiya modelindən (ONT) istifadə etmək təklif olunur. Bu model BB-nin reallaşması zamanı (məsələn CLIPS, JESS, Drools, RuleML, SWRL, OWL, RDF və s.) istifadə edilən müxtəlif BTDbiliklərin təsviri xüsusiyyətlərindən abstraktlaşmağa və bilikləri özünün müstəqil formatında saxlamağa imkan yaradır. Ontologiya modeli həm giriş, həm də çıxış modeli kimi çıxış edə bilər. Modelin ətraflı təsviri [6] göstərilir.

$$I = \{i_1, \dots, i_n, i_j = \text{name}_j, \text{command}_j, j \in 1, n\}$$

burada name_j qarşılıqlı əlaqələrin j üsulunun adı; command_j - qarşılıqlı əlaqə metodunu idarə edən komandadır.

Qarşılıqlı əlaqə interfeysi BB müəyyən BTD (CLIPS və ya OWL) avtomatik formalaşması üçün xarici proqram sistemlərinə konseptual modelin ötürülməsi (importu) yolu ilə proqram komponentinə (sorgu vasitəsi ilə) müraciət etmək imkanı verir.

Beləliklə, proqram komponentinin yaranması prosesində (M_{TPC} ümumiləşdirilmiş proqram komponenti modelinin xüsusişdirilməsi) istifadəçi ilkin konseptual modellərin məqsədli BB-də çevrilməsi (transformasiya ssenarisi) qaydalarını müəyyənləşdirən M_T transformasiya modelini formalaşdırmalıdır. Həmçinin istifadəçi proqram komponentinin “yığıldığı” müvafiq analiz və generasiya bloklarını da müəyyən etməlidir (seçməlidir).

2.1. Transformasiya modeli. (1)-dən istifadə edərək M_T təyin edək:

$$M_T = \langle MM_{IN}, MM_{OUT}, T \rangle, \quad (2)$$

Burada MM_{IN} – ilkin (giriş) konseptual modelin metamodelidir; MM_{OUT} - çıxış modelinin biliyi təqdim etmə (BB) modelinin metamodelidir; T - modellərin çevrilmə operatorudur. M_T –nin əsas elementi çevirmə operatorudur (T):

$$T: CM \rightarrow KB, \quad (3)$$

burada CM – ilkin konseptual modeldir; KB – məqsədli BB. (3) dəqiqləşdirmək üçün [5]-ə əsasən modelləşdirmənin konseptual dillərinin sintaks növlərini və BTD ayrağı. Beləliklə abstrakt (abstract syntax) və konkret (concrete syntax) səviyyələrdə alırıq:

$$T_{AS}:MM_{CM} \rightarrow MM_{KB}, T_{TC}:M_{XML} \rightarrow Code_{KRL}, \quad (4)$$

burada MM_{CM} – ilkin konseptual modelin metamodelidir; MM_{KB} – məqsədli BB metamodelidir; M_{XML} – XML formatında təqdim olunmuş ilkin konseptual modeldir; $Code_{KRL}$ – məqsədli BTD təqdim olunmuş BB kodudur (Knowledge Representation Language, KRL). Eyni zamanda

$$Code_{KRL} \in \langle Code_{KRL}^{CLIPS}, Code_{KRL}^{OWL} \rangle,$$

burada – $Code_{KRL}^{CLIPS}$ – BTD CLIPS üzrə məqsədli BB kodudur; $Code_{KRL}^{OWL}$ – BTD OWL üzrə məqsədli BB kodudur.

(4)-ə əsasən $Code_{KRL}$ -də M_{XML} transformasiyası prosesi ilkin MM_{CM} metamodelin abstrakt elementləri ilə MM_{KB} – məqsədli metamodel arasında uyğunluqların tapılması yolu ilə həyata keçirilir.

Metamodellərin elementlərinin uyğunluq qaydalarını qeyd etmək üçün (transformasiya qaydalarını) R_T operatorunu daxil edək:

$$R_T = r_1, \dots, r_n: MM_{CM} \rightarrow MM_{KB},$$

burada r_i – transformasiya qaydasıdır. Eyni zamanda:

$$r_i = e_i^{in}, e_i^{out}, p_i, i \in 1, n,$$

burada e_i^{in} – MM_{CM} metamodelinin ilkin elementidir (qaydanın sol hissəsi); e_i^{out} – MM_{KB} metamodelinin məqsədli elementidir (qaydanın sağ hissəsi); p_i – qaydaların yerinə yetirilməsinin ardıcılığını təmin edən qaydanın yerinə yetirilməsi prioritetidir, $p_i \in 1, k$.

3. Predmet təyinatlı TMRL dili. M_T transformasiya modelinin təqdim olunması və saxlanması üçün mövzu istiqamətli dil Domain Specific Language (DSL)- işlənib hazırlanmışdır – Transformation Model Representation Language TMRL). Hazırlanmış TMRL qrammatikası sərbəst kontekstli qrammatikalar sinifinə aiddir (SK-qrammatikalar – LL(1)) [1]. TMRL konstruksiyaları M_T transformasiya modelinin elementlərini, xüsusi ilə də R_T metamodellərinin uyğunluq qaydalarını deklarativ şəkildə təsvir etmək üçün imkan yaradır.

4. Proqram komponentlərinin işlənməsi metodu. Proqram komponentlərinin işlənməsi metodu ümumiləşdirilmiş proqram komponentinin qurulması və kopyalanması əsasında İS-in proqram komponentinin avtomatlaşdırılmış işlənmə məsələsinin həllinə istiqamətlənən fəaliyyətin sistemləşdirilmiş toplusudur (M_{TPC} ümumiləşdirilmiş proqram komponenti modelinin M_T transformasiya modelinin formalaşması yolu ilə ixtisaslaşması və uyğun analiz və generasiya bloklarının seçilməsi). Proqram komponentlərinin yaradılması metodunun əsas prinsipləri aşağıdakılardır:

- Ümumi istifadə üçün nəzərdə tutulan modellərin (məsələn QVT, ATL və s.) ixtisaslaşdırılmış transformasiya dillərindən istifadə etmədən ilkin konseptual modelin metamodelinin elementlərinin məqsədli BB metamodelinin elementlərinə transformasiyası (uyğunluq) qaydalarının vizual quruluşu;

- Transformasiya qaydalarının XSD – strukturların analizi əsasında vizual qurulması prosesi üçün giriş metamodellər elementinin avtomatik ayrılması

- Metamodellərin elementlərinin predmet təminatlı deklarativ dildən (TMRL) istifadə etməklə, uyğunluq qaydalarının təqdim olunması və saxlanması.

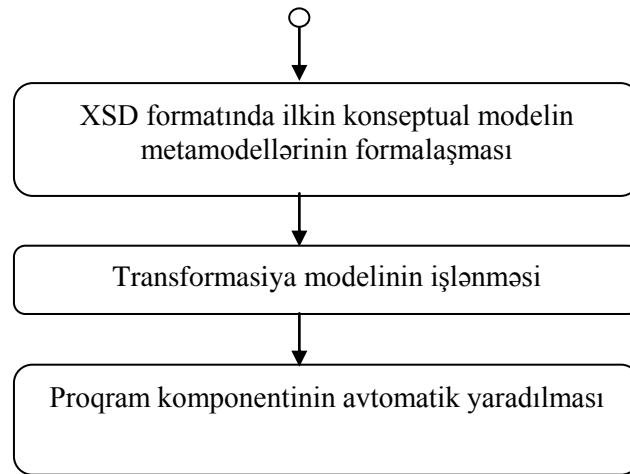
Giriş və çıxış metamodellərinə qoyulan əsas məhdudiyyətlər:

- MM_{CM} ilkin konseptual modellərin metamodeli XML – sxem formatında (XML Schema) təqdim olunmalıdır – XML Schema Definition (XSD) [4];

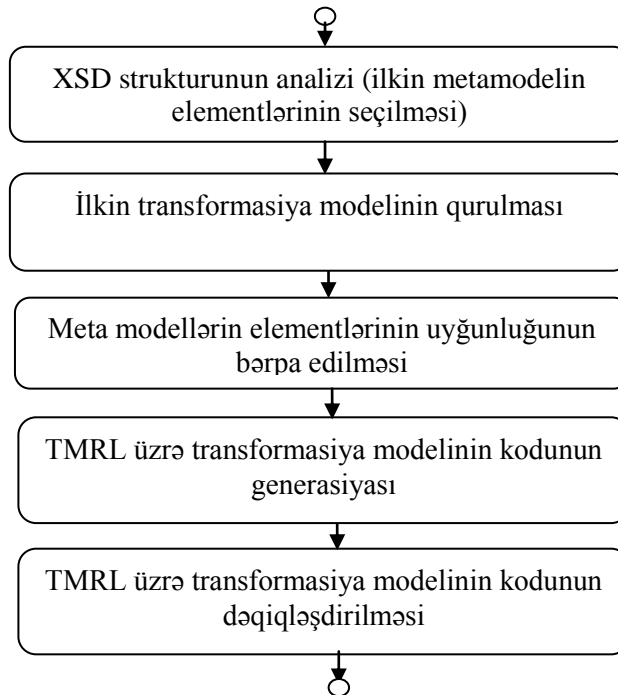
- XML - sxemin avtomatik qurulması üçün müxtəlif layihə şablonları işlənib hazırlanmışdır. Onlardan ən çox yayılanlar: Russian Doll, Salami Slice, Venetian Blind, Garden of Eden. Bu şablonlar istifadə olunan qlobal elementlərin və tiplərin (növlərin) miqdarı ilə fərqlənir. Belə hesab olunur ki, Venetian Blind və Garden of Eden [3] şablonların istifadəsi daha perspektivlidir;

- BTM CLIPS və OWL-də təqdim olunan məqsədli BB MM_{KB} metamodelləri transformasiya modelinin qeyri-şərtsiz qurulması üçün mövcuddur.

Ümumilləşdirilmiş proqram komponentinin modeli əsasında transformasiya modelinin formalaşması da daxil olmaqla, proqram komponentinin yaradılması metodunun əsas mərhələləri şəkil 1 və 2-də göstərilib.



Şəkil 1. Proqram komponentlərinin yaranma mərhələləri



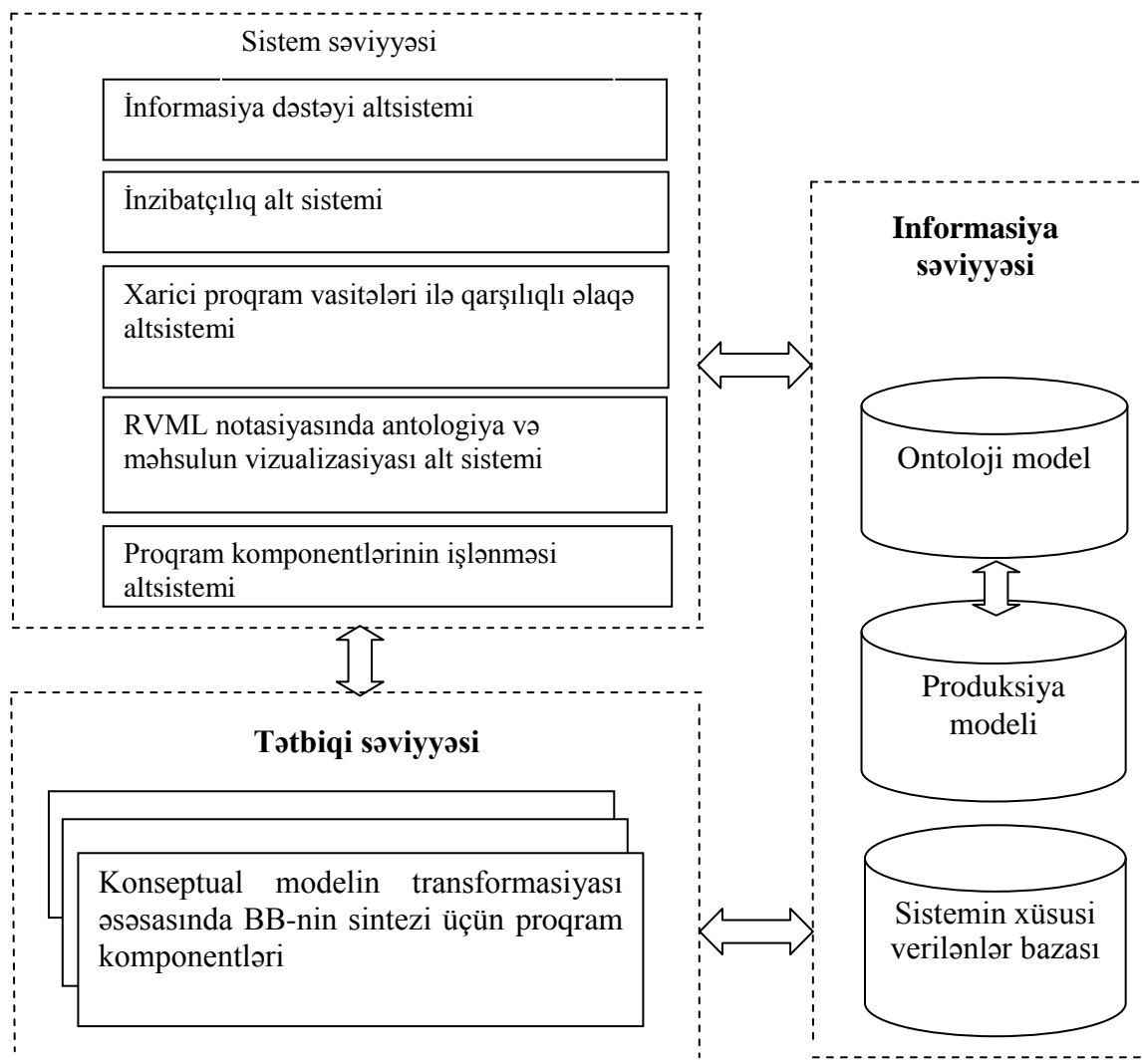
Şəkil 2. Transformasiya modelinin yaradılması mərhələləri

5.Proqram sisteminin konseptual arxitekturası (quruluşu). Təklif olunan texnologiyanın proqram təminatını reallaşdırmaq üçün əsas funksional elementləri və onların qarşılıqlı əlaqəsini təsvir edən veb-istiqamətli proqram sisteminin konseptual arxitekturası (quruluşu) işlənilib hazırlanmışdır. Veb-istiqamətli (şək.3) proqram sisteminin konseptual arxitekturası (quruluşu) elementlərinin tərkibi və növləri, həmçinin fəaliyyətinin prinsipial özəllikləri də daxil olmaqla onun strukturunu təsvir etməyə imkan yaradır. Sistemin əsas səviyyələri (hissələri):

- İnformasiya səviyyəsi – proqram sisteminin xidməti məlumatını saxlamaq üçün nəzərdə tutulan və altsistemlər tərəfindən (sistem və tətbiqi səviyyədə) həm öz fəaliyyətinin təmini, həm də BB-nin avtomatlaşdırılmış işlənmə məsələlərinin həlli üçün istifadə olunan bütün informasiya resurslarını əks etdirir;

• Sistem səviyyəsi – həm istifadəçinin veb-istiqamətli proqram sistemi ilə baza əlaqələrini, həm xarici proqram sistemləri ilə (İS) interfeysi təmin edən, həmçinin nümunə proqram komponentinin əsasında tətbiqi proqram komponenti yaratmaq üçün təqdim edən istifadəçinin təklif etdiyi bütün altsistemlərin toplusudur;

• Tətbiqi səviyyə - konseptual modellərin transformasiyası yolu ilə BB avtomatik sintezini təmin edən istifadəçi tərəfindən işlənmiş proqram komponentlərinin toplusudur. Proqram komponentləri nümunə proqram komponentinin əsasında onun “köçürülməsi” (kopyalanması) və xüsusişdirilməsi yolu ilə yaradılır.



Şəkil 3. Veb-təyinatlı proqram sisteminin konseptual arxitekturası

Nəticə. Tədqiqat işində konseptual (informasiya) modellərin transformasiyası yolu ilə CLIPS və ya OWL-də təqdim olunmuş və BB kodunun avtomatik formalaşması üçün nəzərdə tutulan İS proqram komponentlərinin avtomatlaşdırılmış işlənməsinə yanaşma (texnologiya) təklif olunur. Təklif olunan yanaşmanın əsasında: tipləşdirilmiş proqram komponentinin modelindən (transformasiya modeli daxil olmaqla); transformasiya modelinin təqdim olunması və saxlanılması üçün nəzərdə tutulan Transformation Model Representation Language (TMRL) deklarativ dildən; ümumiləşdirilmiş komponentin köçürülməsi və onun sazlanması (xüsusişdirilməsi) əsasında proqram komponentlərinin işlənməsi metodundan; veb-istiqamətli proqram sisteminin konseptual arxitekturasından (quruluşundan) istifadə olunması durur.

ЛИТЕРАТУРА

1. Ахо А.В., Лам М.С., Сети Р. Ульман Дж.Д. Компиляторы, принципы, технологии и инструментарий. 2-е изд.: Пер. с англ. М.: ООО И.Д. Вильямс, 2008. 1184 с. [Aho A.V, Lam M.S., Sethi R., Ullman J.D. Compilers: principles, Techniques, and Tools. 2nd ed. Addison-Wesley, 2006. 1000 p.]
2. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000, 384 с.
3. Документация XML Metadata Interchange (XMI). Режим доступа: <http://www.omg.org/spec/XMI> (дата обращения 10.03.2016).
4. Дородных Н.О., Юрин А.Ю. Использование диаграмм классов UML для формирования продукционных баз знаний // Программная инженерия. 2015. №4. С. 3-9.
5. Частиков А.П., Гаврилова Т.А., Мелов Д.Л. Разработка экспертных систем. Среда CLIPS. СПб.: БХВ-Петербург, 2003, 608 с.
6. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide, 2nd Edition. Addison-Wesley, New York, 2005, 496 p.

РЕЗЮМЕ

КОНЦЕПЦИЯ ПОДХОДА К СОЗДАНИЮ ПРОГРАММНЫХ КОМПОНЕНТОВ ГЕНЕРАЦИИ БАЗ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ УПРАВЛЯЮЩИХ СИСТЕМАХ

Manafova X.I.

Ключевые слова: автоматизация разработки программных компонентов, интеллектуальная система, концептуальная модель, база знаний, трансформация моделей

Работа посвящена проблеме повышения эффективности создания интеллектуальных систем и их подсистем (компонентов). Основную сложность при разработке интеллектуальных систем представляет этап формирования баз знаний. Эффективность данного этапа может быть повышена путем трансформации концептуальных моделей, построенных при помощи различных программных средств онтологического моделирования или CASE- средств, в код баз знаний, представленных на определенном языке представления знаний. При этом актуальным является создание унифицированного подхода (технологии) для поддержки разработки программных компонентов интеллектуальных систем, обеспечивающих формирование (генерацию) баз знаний путем трансформации концептуальных моделей.

SUMMARY

CONCEPT OF APPROACH TO CREATING SOFTWARE COMPONENTS FOR GENERATION OF KNOWLEDGE BASES IN INTELLECTUAL CONTROL SYSTEMS

Manafova Kh.I.

Key words: automation of software components development, intellectual systems, conceptual model, knowledge base, model transformation

The article discusses the problem of improving the process of designing intellectual systems and their components. The main problem in designing of intellectual systems is the stage of creating knowledge bases. The efficiency of this stage can be improved by the transformation of conceptual (information) models into program codes of knowledge bases. In turn the conceptual models can be created with the aid of different CASE tools or software for cognitive and ontological modeling; knowledge bases can be represented by the specific knowledge representation languages. Therefore, the creation of the unified approach (technology) for developing software components of intelligent systems providing generation of knowledge bases by transformation of conceptual models is actual.

| | | |
|-------------------|---------------|------------|
| Daxilolma tarixi: | İlkin variant | 05.04.2019 |
| | Son variant | 25.09.2019 |