

занять викладачами, відкритих лекцій і контрольних відвідувань занять завідувачем кафедри, причому один раз в рік у моніторинг занять по кафедрі включається одна вступна, одна підсумкова і одна установча лекція для студентів заочної форми навчання з обов'язковим обговоренням підсумків моніторингу на засіданні кафедри.

Отже, лекція у вищій школі – це відповідальне багатоаспектне педагогічне дійство. Воно є вершиною педагогічної майстерності педагога-науковця. І тому в системі моніторингу якості навчального процесу так важливо виявити складові, за якими можна оцінити ефективність і якість проведення лекційних занять.

Література

1. Кузьминський А.І. Педагогіка вищої школи: Навч. посібник. – К.: Знання, 2005. – 485 с.
2. Гончаренко С.У. Український педагогічний словник. – К.: Либідь, 1997. – С.189.
3. Фрейман Л.С. Методика подготовки лекций по общетехническим дисциплинам. – М.: Высшая школа, 1969. – 325 с.

Надійшла 8.2.2009 р.

УДК 004.942

Р.Б. ВОВК, В.І. ШЕКЕТА

Івано-Франківський національний технічний університет нафти і газу

ФОРМАЛІЗАЦІЯ СУМАРНОЇ МОДЕЛІ СТУДЕНТА В ІНТЕЛЕКТУАЛЬНІЙ ТЬЮТОРНІЙ СИСТЕМІ НА ОСНОВІ ОБМЕЖЕНЬ

Запропоновано формальну модель студента в процесі застосування інтелектуальної навчальної системи, яка генерує коректуючі педагогічні впливи на основі активних профілів суб'єктів навчання з використанням концепції задоволення обмежень, аналізу типових помилок та зворотного зв'язку.

A formal model of the student in the application of intellectual tutoring system is proposed, which generates adjusting educational influences on the basis of learning subject actives profiles using the constraints satisfaction concepts, the analysis of typical student's mistakes and feedbacks.

Вступ. Створення навчальних систем дозволяє реалізувати функції комп'ютера як засобу оптимізації навчального процесу, організованого і керованого викладачем в умовах традиційного педагогічного спілкування „викладач – студент” [1,2]. Результати аналізу багаточисельних емпіричних даних, отриманих дослідниками даної проблеми, показують, що ефективність навчання підвищується, якщо студенти мають не лише доступ до сучасних комп'ютерних засобів організації навчального процесу, але і можливість спілкування з викладачем-людиною або комп'ютерним тьютором. Технологія тьюторних навчальних систем [3] передбачає взаємодію досвідченої людини (тьютора), яка вміє щось робити чи знає як робити в певній предметній області і початківця. Основним завданням в області навчальних тьюторингових систем та інтелектуальних систем навчання є побудова комп'ютерних аналогів індивідуального навчання (тьюторів), що по ефективності може наближатися до індивідуального тьюторингу, що проводиться викладачем навчального курсу. Така ціль є перспективною і існуючі системи в тій чи іншій степені наближаються до даного ідеального рішення.

Аналіз досліджень в області інтелектуальних систем навчання [1-3] показує, що центральним елементом кожної такої системи є вибір та реалізація засобів моделювання студента. Досягнення ефективності навчальної системи та імплементація її інтелектуальності не можлива без адаптивного контролю поточного та латентного рівня знань студентів і корекції дій системи залежно від здібностей та потреб конкретного студента. Ряд існуючих розробок в області моделювання студента як суб'єкта навчального процесу з використанням новітніх засобів обчислювальної техніки прив'язані до певної області знань або до вибраної стратегії навчання. Побудова моделі студента на основі концепції представлення та використання обмежень, виконується на основі співставлення з певною областю знань (наприклад послідовність навчальних курсів певної дисципліни). Використання такого підходу дозволяє одержати програмну реалізацію інтелектуального тьютора, що дозволяє організувати середовище проблемно-орієнтованого концептуального навчання новітнім технологіям програмування з використанням професійно-орієнтованих стратегій та мета-стратегій. Найбільш оптимальною на сьогоднішній день є реалізація тьюторної навчальної системи на основі веб-базованої платформи, що дозволяє студентам отримувати доступ в режимі онлайн і офлайн, з використанням локальної мережі університету чи кафедри або віддаленого доступу на основі незалежних провайдерів технологій WI-FI, мобільного інтернету.

Однією з актуальних проблем штучного інтелекту є проблема задоволення обмежень (CSP – constraints satisfaction problem) [4], яка має ряд застосувань: прогнозування, розподілення ресурсів, планування, та ін. Наукові пошуки в області CSP базуються на класичних задачах штучного інтелекту, мовах програмування штучного інтелекту, абстрактних обчисленнях, теоріях логіки. Формалізовано

проблема задоволення обмежень може бути представлена у вигляді множини змінних $X = \{x_i\}, i = \overline{1, n}$, скінченних множин D_i їх можливих значень (доменів), $D = \{D_i\}, i = \overline{1, n}$, і множини обмежень які обмежують значення, що змінні можуть одночасно приймати $Constr = \{Constr_i\}, i = \overline{1, n}$. Розв'язком CSP є набір значень з відповідних доменів для кожної змінної, таких щоб задовольнялося кожне накладене обмеження.

Моделювання студента є центральною проблемою в області інтелектуальних систем навчання. Існує ряд підходів до побудови моделі студента. Новим підходом є моделювання на основі обмежень [3, 4]. Даний підхід є ефективним і дозволяє уникнути ряду проблем з якими стикаються інші підходи до моделювання студента. Для моделювання студента існує ряд технік та методик розвинутих для виділених предметних областей. Проте питання їх застосування в інших предметних областях є мало дослідженим та очевидно складним завданням, що потребує побудови уніфікованої формальної моделі.

Недослідженим залишається питання створення моделі студента в навчальних тьюторних системах з використанням концепції CSP.

Таким чином, **метою даної статті** є формалізація представлення сумарної моделі студента на основі підходу задоволення обмежень.

Основне завдання будь-якої інтелектуальної системи полягає у вивченні певних розділів навчального матеріалу. Згідно класичної архітектури тьюторна система повинна складатися з інтерфейсу, педагогічного модуля, який визначає часові характеристики та наповнення педагогічних дій, а також блоку моделювання студента, який аналізує запропоновані суб'єктом навчання рішення (розв'язки). Система містить теоретичні розділи, а також множину проблем та ідеальних рішень для них. Деякі тьютори включають доменні модулі, а інші не включають їх, тобто є предметно залежними. Основне завдання яке вирішується на кожному з етапів полягає в перевірці коректності рішення запропонованого студентом; для цього система використовує доменні знання представлені в формі обмежень. На початку навчальної сесії (сеансу роботи) тьютор вибирає певну проблему з якою буде працювати студент. Після того як студент відправляє рішення педагогічний модуль перенаправляє її в блок моделювання студента, який аналізує дане рішення, ідентифікує помилки і відповідно оновлює модель студента. На основі моделі студента педагогічний модуль генерує відповідну педагогічну дію (зворотній зв'язок). Після того як поточна проблема є вирішеною або студент задає запит на нову проблему, педагогічний модуль вибирає відповідну проблему на основі поточного профілю з моделі студента.

Основні вимоги до реалізації інтерфейсу тьютора це робастність, гнучкість, і простота використання.

Після завершення розробки тьютора його необхідно протестувати і оцінити динаміку зміни знань студентів, при класичному навчанні та при навчанні з використанням тьютора. Важливим фактом є те, що дослідження впроваджених тьюторів на основі обмежень показує що студенти легко засвоюють технологію навчання, що базується на основі зворотного зв'язку та контролю системи в формі обмежень. Студенти мають кілька варіантів вибору навчальних проблем в тьюторі. Зокрема вони можуть вибирати проблеми по наростаючій складності, натискаючи кнопку наступна проблема. Інша опція – це вибір проблеми системою. Такий вибір здійснюється на основі поточного профілю студента в його моделі. Третім способом є задання незалежної стратегії вибору проблем, що базується на певних математичних моделях, наприклад – мережі Байєса. Інтерфейс повинен відображати такі основні елементи – текст проблеми, структуру запиту, пояснення поточних концептів предметної області. Особливо позитивним ефектом є включення в архітектуру системи педагогічного агента, на зразок помічника віндос, який стимулює мотивацію студентів та зменшує рівень стресу.

Навчальні проблеми є релевантними до множини обмежень в приблизно рівних пропорціях. Хоча існує множина обмежень, яка релевантна до всіх проблем, а також деяка множина обмежень нерелевантних до жодної проблеми (у зв'язку з високою динамікою оновлення веб-базованих навчальних курсів). Перед початком рішення всіх проблем система може видавати короткий тьюторинг щодо користування системою, а також по окремих розділах теорії з яких взята навчальна проблема.

Розглядатимемо базу даних навчальних проблем тьюторної системи у вигляді деякої множини навчальних проблем $P_{set} = \{P_i\}, i = 1 \dots n_1$, де кожна проблема P_i має j - станів.

Означення. Обмеженням для стану j проблеми P_i , P_i^j будемо вважати впорядкований триплет $[Constr_r^j, Constr_s^j, Constr_v^j]$, де $Constr_r^j$ є релевантним обмеженням із множини накладених обмежень $Constr_{set}^{P_i^j}$ на поточний стан проблеми, $Constr_{set}^{P_i^j} = \left\{ Constr_k^{\{P_i^j\}} \right\}$, $k = 1 \dots n_{constr}$, де n_{constr} - кількість накладених обмежень на множину рішень навчальних проблем $Solution_{set}^{P_i^j}$, $Constr_s^j$ є обмеженням, що задовольняється і $Constr_v^j$ - є обмеженням, що було порушене в поточній сесії.

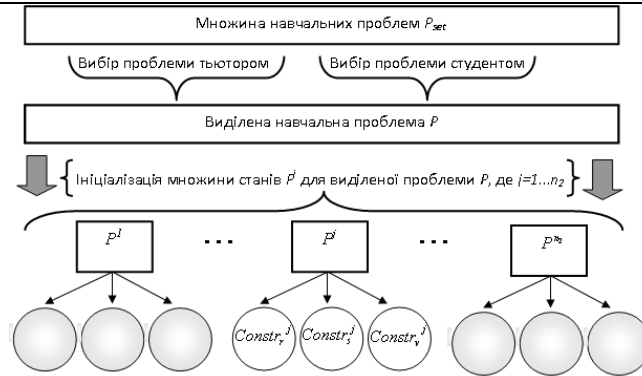


Рис. 1. Модель основних елементів архітектури навчальної системи на основі підходу задоволення обмежень

Виконаємо представлення знань предметної області *SubjectDomain* як множини обмежень стану проблеми $Constr_{set}^{P_i^j}$, тобто обмежень, що інтерпретують множину еквівалентних станів проблеми $[P_i^j]^E$, де $E \leq J$. Кожен з одержаних класів еквівалентності $Class^{P_{set}^E}$ ініціює в системі запуск однакових педагогічних дій $\{f_i\}_{i=1...9}$, де кожне з f_i прийматиме одне із значень: 1) індикація вірності чи невірності рішення, 2) повідомлення про помилку, 3) підказка, 4) детальна підказка, 5) вивід всіх помилок, 6. виведення рішення, 7) показ порушених обмежень, 8) показ задоволених обмежень, 9) показ релевантних обмежень. Тобто стани проблеми в класі еквівалентності є педагогічно еквівалентними $P_i^{j_1} \equiv P_i^{j_2}$, де $P_i^{j_1}, P_i^{j_2} \in Class^{[P_i^j]^E}$.

Відповідно $Constr_r^j$ використовується для ідентифікації класу еквівалентності $Class^E$ або класу станів проблеми $ClassR^{\{P_i^j\}}$ в якому обмеження $Constr_r^j$ є релевантним (тобто має безпосереднє відношення до послідовності побудови коректних рішень, $\{Solution_{set}^{Correct}\}_l \subset Solution_{set}^{P_{set}^j}$, для $l=1...n_3$, і для деякого значення l існує співпадіння $\{Solution_{set}^{Correct}\}_l = [Solution_{set}^{P_i^j}]_{IDEAL}$ коректного рішення з ідеальним (оптимальним). $Constr_s^j$ ідентифікує клас релевантних станів $ClassS^{\{P_i^j\}}$, в якому $Constr_s^j$ задовольняється, а $Constr_v^j$ ідентифікує клас релевантних станів $ClassV^{\{P_i^j\}}$, в якому $Constr_v^j$ порушується. Оскільки теоретичний принцип, що базується на теорії представлення та обробки обмежень стверджує, що кожне релевантне обмеження повинне задовольнятися або порушуватися, то на множинному рівні матимемо $ClassR^{\{P_i^j\}} = ClassS^{\{P_i^j\}} \cup ClassV^{\{P_i^j\}}$. З другого боку для поточного стану проблеми P_i^j існує деяка множина не релевантних обмежень (тобто таких, що не мають безпосереднього синтаксичного та семантичного відношення до поточного рішення проблеми), які утворюватимуть клас еквівалентності $Class\bar{R}^{\{P_i^j\}}$ в якому обмеження $Constr_r^j$ є не релевантними.

Кожне обмеження специфікує властивість предметної області яка повинна задовольнятися всіма коректними рішеннями $Solution_{set}^{Correct}$. Якщо $Constr_r^j$ задовольняється в певному стані проблеми P_i^j , то для того щоб даний стан проблеми був коректно-релевантним, він повинен також задовольняти $Constr_s^j$ (тобто запропоновані студентом рішення $StudentSubmit^{P_i^j} \subset Solution_{set}$ включає розгляд стану P_i^j).

Досягнення ефективності навчальної системи та імплементація її інтелектуальності не можлива без адаптивного контролю поточного та латентного рівня знань студентів і корекції дій системи залежно від здібностей та потреб конкретного студента. Інтелектуальна тьюторингова система складається з статичних компонент: бази даних навчальних проблем, бази даних рішень і бази даних обмежень $ITS^{st.} = [DB_{set}^{Probl.}, DB_{set}^{Sol.}, DB_{set}^{Constr.}]$ та динамічних: множина моделей студента, що ініціалізує вибір педагогічного впливу $ITS^{dnm.} = [SM_{set} \xrightarrow{CBM} PA_{set} = \{f_i\}_{i=1...9}]$. Процес вибору є визначальним в роботі системи та є складовою процесу моделювання на основі обмежень (*CBM, Constraint-based modeling*).

Додатковою перевагою підходу СВМ є обчислювальна простота. Тобто можна не використовувати комплексні механізми логічного висновку оскільки моделювання студента можна звести до контролю відповідності шаблону (Pattern matching). Виділені нами параметри $Constr_r, Constr_s, Constr_v$ є комбінацією паттернів (шаблонів) і таким чином можуть бути представлені в легко компільованій формі. На першому кроці всі релевантні до проблеми P_i паттерни співставляються з станами проблеми: $Pattern_{P_i}^R \rightarrow \{P_i^j\}_{j=1...J}$, де J - кількість станів проблеми. На другому кроці компоненти контролю задоволеності для обмежень, що відповідають певному стану проблеми з першого кроку, тобто релевантні обмеження також співставляються $\{ConstrR^{[P_i^j]}\} \leftrightarrow Pattern_{P_i}^R \leftrightarrow \{P_i^j\}_{j=1...J}$. Якщо шаблон задоволення відповідає стану проблеми тоді обмеження задовольняються і тьюторна система не виконує жодних дій (коректуючих педагогічних впливів $f_i|_{i=0}$).

$$\left\{ \left\{ ConstrS^{[P_i^j]} \right\} \leftrightarrow Pattern_{[P_i^j]}^S \leftrightarrow \{P_i^j\}_{j=1...J} \Rightarrow \{f_i\}_{i=0...9} \right.$$

В протилежному випадку – обмеження порушуються. Модель студента складатиметься з усіх порушених обмежень $\{ConstrV^{[P_i^j]}\}$, де $i = 1...I, j = 1...J$.

Використання такого підходу дозволяє одержати програмну реалізацію веб-базованого інтелектуального тьютора, що робить можливим створення середовища проблемно-орієнтованого концептуального навчання новітніми технологіями програмування з використанням професійно-орієнтованих стратегій та мета-стратегій.

Якщо виходити з основного завдання – моделювання знань студента *StudentKnowledges* в найбільш повному та точному обсязі то така задача виходить за рамки теорії розробки комп'ютерних та інформаційних систем і повинна базуватися на фундаментальних теоріях когнітології, психології та педагогіки навчання. В нашому випадку інтерес представляють моделі які можна формалізувати математичною мовою, оскільки їх легко запрограмувати. В той же час побудова формальних моделей в когнітології, психології та педагогіці є перспективними темами незалежних досліджень в області прикладної математики. Дослідники виділяють значну ефективність неповних *InCompleteModels* та неточних *InConsistentModels* моделей студента з точки зору практичного застосування для інтелектуалізації навчальних систем. Така ідея спирається на величезний емпіричний досвід викладання фахових дисциплін в умовах ВУЗу, який говорить про те, що викладач-людина в своїй педагогічній діяльності часто спирається на неповні моделі і в той же час є ефективним викладачем, що виконує поставлені цілі навчального процесу. Цікавою є також думка про те, що найбільш ефективним способом зворотного зв'язку (*FeedBacks*) із студентом є вказання інформативних повідомлень про помилки допущені в поточній сесії роботи. В той же час виведення системою більш глибоких причин виникнення певних помилок які допустив студент, базуючись на вказанні хибних концепцій в його розумінні навчального матеріалу призводить до зниження самооцінки та мотивації суб'єкта навчання. Таким чином – основним завданням дослідження в області побудови адекватних моделей студента є зниження обчислювальної складності задачі. Одним з ефективних підходів даного напрямку є підхід на основі обмежень [1-3]. Підхід СВМ (*Constraints Based Modelling*) базується на теоретичній концепції представлення та задоволення обмежень та теорії Олсона [3], як навчання на основі помилок виконання. СВМ в якості центрального елемента виділяє хибні знання студента. Проте іншим аксіоматичним припущенням підходу є те, що виділених хибних знань є недостатньо для опису множини коректних знань студента, тобто ідеальна сумарна модель знань студента не дорівнює формулі хибні знання + коректні знання:

$$StudentModel^{Sum} \neq InCorrectKnowledges + CorrectKnowledges$$

Базовим припущенням є також те, що діагностична інформація не закладена в послідовності дій студента *StudentActions*, а закладена в певній ситуації предметної області *SubjectDomain* (наприклад, навчальна проблема P , та її стан P^j) до якої дійшов студент в поточній сесії *CurrentSession*. Основне підтвердження даного факту полягає в тому, що не існує коректного рішення проблеми *CorrectSolution*,

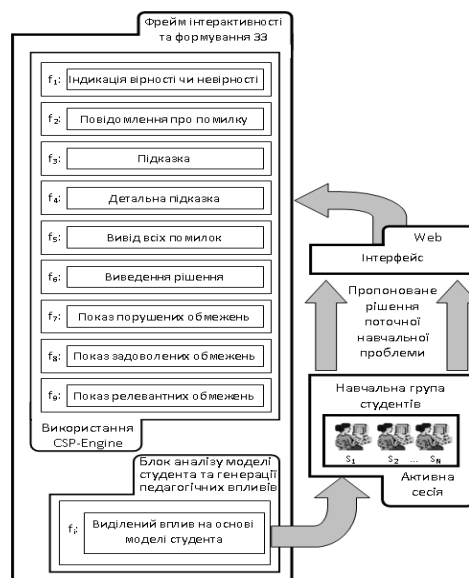


Рис. 2. Структура зворотного зв'язку тьюторної системи на основі обмежень

що виходить за простір описаних станів проблеми $Scope^{P^j}$, що порушує фундаментальні ідеї та концепції предметної області $SubjectDomain = \{Conceptions_i\}$. Таким чином модель студента не представляє безпосередньо дії студента, а в більшій мірі описує ефекти виконаних дій

$$StudentModel^{Sum} \neq \{StudentActions_i\} \approx \{Actions Implications_i\}.$$

Даний підхід представляє знання предметної області як множину обмежень стану проблеми, тобто обмеження означають множину еквівалентних станів проблеми. Клас еквівалентності ініціює в системі запуск однакових навчальних дій, тобто стани проблеми в класі еквівалентності є педагогічно еквівалентні. Всі коректні рішення проблеми не порушують жодне з обмежень. Порушене обмеження сигналізує наявність помилки, яка відповідно є наслідком неповних і некоректних знань.

Згідно ідеї Олсона [3] модель студента розглядається як сукупність порушеним ним обмежень. Введемо інтерпретацію даного підходу на основі аналізу структури та функціонування типової тьюторної системи. Нехай під час навчальної сесії з системою працює група з N студентів: S_1, S_2, \dots, S_N в якій зареєстровано L - навчальних курсів. $\{Course_l\}_{l=1 \dots L}$.

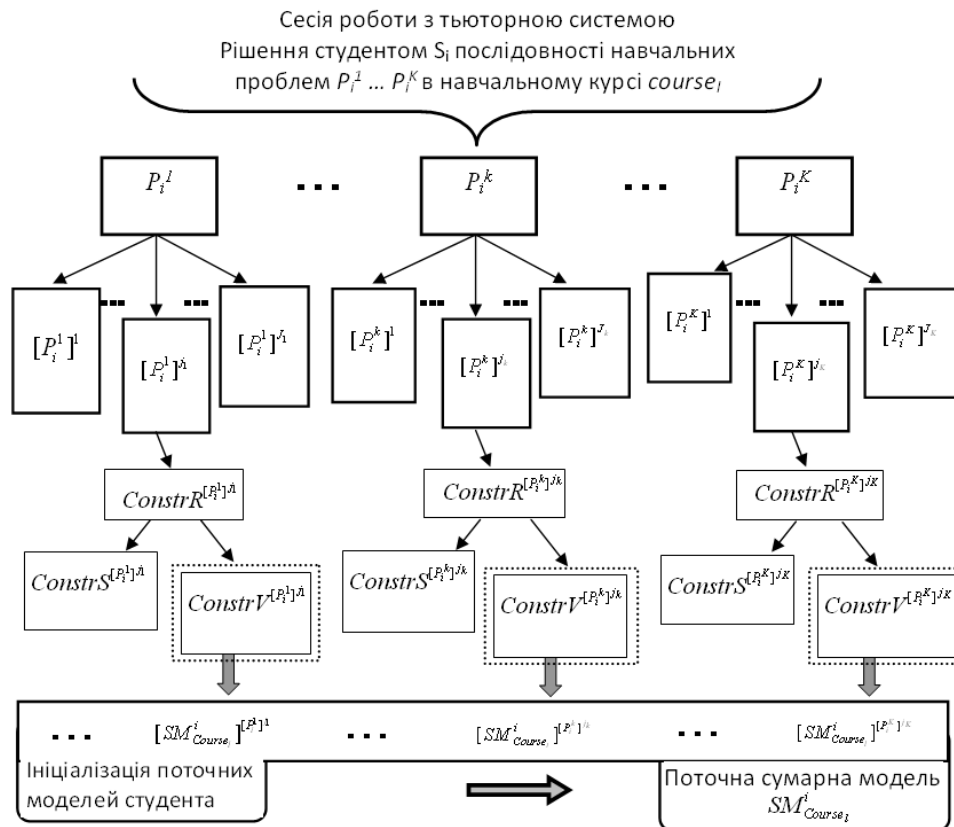


Рис. 3. Схема формування поточної сумарної моделі студента

Нехай i -й студент під час сеансу роботи працює з навчальним курсом $Course_l$ і вирішує в даному курсі послідовність навчальних проблем $[P_{course_l}^1]_i, [P_{course_l}^2]_i, \dots, [P_{course_l}^K]_i$, де $[P_{course_l}^i]_i \in P_{course_l}^{set}$, де $P_{course_l}^{set}$ - визначена множина навчальних проблем для курсу $Course_l$. В свою чергу, згідно початкової ідеї навчальна проблема P_i^i розпадається на J - станів. При вирішенні j - стану навчальної проблеми $[P_{course_l}^i]_i^j$ система вибирає певну множину релевантних обмежень $ConstrR^{[P_i^i]_i^j} \subset Constr_{set}^{P_{set}}$, де $Constr_{set}^{P_{set}}$ - загальна множина обмежень накладених на множину навчальних проблем $P_{set} = \{P_i\}, i = 1 \dots n_1$. Оскільки кожне релевантне обмеження повинно задовольнятися або порушуватися, то для j - стану навчальної проблеми $[P_i^i]_i^j$ матимемо співвідношення $ConstrR^{[P_i^i]_i^j} = ConstrS^{[P_i^i]_i^j} \cup ConstrV^{[P_i^i]_i^j}$. Тоді деяке виділене обмеження $c \in ConstrR^{[P_i^i]_i^j}$

приймає значення з двохелементної множини $\{c^{satisf.}, c^{viol.}\}$, де $c^{satisf.} \in ConstrS^{[P_i^1]^j}$, $c^{viol.} \in ConstrV^{[P_i^1]^j}$.

Моделювання студента означається як процес збору релевантної інформації *RelevantInformation* для ідентифікації та представлення рівня знань студента *KnowledgesLevel*. В ідеальному випадку модель студента повинна відображати його знання, та стратегії навчання, що найбільш підходять його індивідуальному профілю, навчальним інтересам, мотивації до навчання.

$$StudentModel^{ideal} \cong StudentKnowledges_{set} \cup \{LearningStrategies_i\} \Big| StudentProfile^{Active}$$

На основі таких параметрів система може вибирати оптимальний стиль навчального матеріалу (контенту) з використанням навчальних переваг та адекватному ступеню концентрації уваги в кожній навчальній сесії: $Session^j = \sum_i \{LearningContent\}_i \Big| LearningPreferences_{i2}$.

Задача побудови моделі студента *StudentModel* є складною і трудомісткою задачею у зв'язку з великою кількістю складових. Деякі дослідники відзначають наявність елементів в емпіричній моделі студента *EmpiricalModel* які важко формалізуються математичними засобами. Зокрема, деякі дослідники рекомендують одержувати інформацію необхідну для побудови моделі студента від самого суб'єкта навчання в процесі інтерактивного діалогу, наприклад при першій сесії навчання в процесі залогінення. Інші дослідники рекомендують виведення таких знань в системі на основі неявних ознак та запитань і одержання неявної моделі *DeducedModel*. Складним питанням є також питання верифікації одержаної моделі *ModelVerification*, перевірки її на несуперечливість *ConsistencyCheck*. Важливою ідеєю є також введення чіткого ранжування елементів та складових одержаної моделі по критерію їх практичної значимості *ImplementationRange*.

Основна ідея яка лежить в необхідності побудови та підтримки моделі студента полягає в генерації системою коректуючих педагогічних впливів на основі активних профілів суб'єктів навчання одержаних з релевантних моделей:

$$StudentModel_i^{relev.} \Rightarrow \{ActiveStudentProfile_i\} \Rightarrow \{PedagogicalActions_i\}$$

При реєстрації нового студента S_i в тьюторі блок моделювання студента присвоює даному студенту деяку шаблонну прототипну модель, яка відповідає базовому рівню засвоєння попереднього навчального курсу або навчальних курсів які є необхідними для розуміння та успішного засвоєння поточного навчального курсу $Course_l$, $l=1...L$, де L - кількість навчальних курсів зареєстрованих та внесених в базу даних тьюторної системи. Таку початкову модель студента можна розглядати як деяку ідеальну модель студента стосовно навчального курсу $Course_l$. Позначатимемо її як $SM_{course_l}^{IDEAL}$. Оскільки під час інтерактивного вивчення поточного навчального курсу i - й студент буде вирішувати деяку скінчену послідовність навчальних проблем $Course_l \rightarrow P_i^1, P_i^2 \dots P_i^k$, то доцільним є формування підсумкової моделі i -го студента як сукупності моделей по визначених вимірах: по навчальних курсах, по проблемах, що вирішуються в кожному з навчальних курсів та по станах проблем із послідовності навчальних проблем курсу. Тобто ми матимемо складену підсумкову модель i -го студента SM_i^{sum} для навчального курсу $Course_l$ $SM_{Course_l}^i$ яка буде містити складові, що відповідають частковим моделям утвореним на основі досвіду студента щодо вирішення списку поточних проблем $[SM_{Course_l}^i]^{P_i}$. В свою чергу кожна з таких складових міститиме підмоделі утворені на основі аналізу порушених обмежень при задоволенні виділених станів проблеми $[SM_{Course_l}^i]^{P_i^j}$. Кінцевий результат можна представити формулою:

$$SM_i^{sum} = \left\{ SM_{Course_l}^i \right\}_{l=1...L} = \left\{ \left\{ [SM_{Course_l}^i]^{P_k} \right\}_{k=1...K} \right\}_{l=1...L} = \left\{ \left\{ \left\{ [SM_{Course_l}^i]^{P_k^j} \right\}_{j=1...J} \right\}_{k=1...K} \right\}_{l=1...L}$$

де SM_i^{sum} - кінцева підсумкова модель i -го студента після вивчення всієї послідовності курсів L ;

$SM_{Course_l}^i$ - модель i -го студента одержана в результаті фіксації порушених обмежень при вивченні навчального курсу l ;

$[SM_{Course_l}^i]^{P_k}$ - модель i -го студента одержана в результаті фіксації порушених обмежень при вивченні навчального курсу $Course_l$, шляхом вирішення навчальної проблеми P_k ;

$[SM_{Course_l}^i]^{P_k^j}$ модель i -го студента одержана в результаті фіксації порушених обмежень при вивченні навчального курсу $Course_l$, шляхом вирішення j -стану навчальної проблеми P_k .

Додатковою перевагою підходу CBM є можливість уникнення деталізованого вивчення та класифікації помилок студентів $StudentErrors$. Суб'єкти навчання схильні до систематичного використання деякої єдиної процедури для вирішення поточних задач $SolvingRoutine$. Кожен студент S_i застосовує одну з класу процедур $RoutinesClass^{i_1}$, які можуть бути застосовані для певного класу задач $TaskClass^{i_2}$ і стратегії вибору процедур $RoutinesSelectionStrategies$ з відповідного класу еквівалентності $Class^{ekvl}$, і залежить від поточної задачі $CurrentTask$. Таким чином середній студент $AverageStudent_i$ може знати кілька різних процедур $Routines^{i_2}$ для рішення поточної задачі і переключатися між ними. Проте для системи стратегія поведінки та мислення студента немає особливого значення, оскільки ефективність стратегії вибраної студентом $SolvingStrategy_i$ оцінюється системою з точки зору порушених обмежень $Constraints_{set}^{viol}$ на основі підсумкової моделі студента.

Висновки. В даній статті виконано формалізацію основних елементів архітектури тьюторної навчальної системи на основі обмежень. Представлено схеми структури зв'язків, що формуються блоком аналізу моделі студента та генерації педагогічних впливів які здійснюються системою при порушенні студентом обмежень релевантних до поточної навчальної проблеми. Запропоновано сумарну формальну модель студента, одержану в результаті фіксації порушених обмежень при вивченні навчальних курсів, що розглядаються як сукупність навчальних проблем.

Подальші дослідження даного напрямку будуть спрямовані на імплементацію теоретичних принципів концепції $Constraints Satisfaction Problem$ у власному проекті реалізації прототипу веб-базованого тьютора для вивчення мов програмування на основі аналізу типових помилок студентів.

Література

1. Оксамитна Л.П. Методи та засоби самоорганізації моделі знань в автоматизованих системах контролю знань та навчання: Автореф. дис... канд. техн. наук: 05.13.06 / Черкас. держ. технол. ун-т. – Черкаси, 2003. – 18 с.
2. Григорова А.А. Методи, алгоритми та технології контролю знань в системах навчання: Автореф. дис... канд. техн. наук: 05.13.06 / Херсон. держ. техн. ун-т. – Херсон, 2004. – 21 с.
3. Ohlsson S. Constraint-based student modeling // In: Greer J.E., McCalla G. (Eds): Student modeling: the key to individualized knowledge-based instruction. – 1994. – P.167-189.
4. http://en.wikipedia.org/wiki/Constraint_satisfaction

Надійшла 2.3.2009 р.

Вимоги до оформлення рукопису – <http://visniktup.narod.ru/rules/>

Рекомендовано до друку рішенням Вченої ради Хмельницького національного університету, протокол № 10 від 29.4.2009 р.

Підп. до друку 30.4.2009 р. Ум.друк.арк. Ум.друк.арк. 31,49 Обл.-вид.арк. 29,96
Формат 30x42/4, папір офсетний. Друк різнографією.
Наклад 100, зам. № _____

Тиражування здійснено з оригінал-макету, виготовленого редакцією журналу “Вісник Хмельницького національного університету” редакційно-видавничим центром Хмельницького національного університету 29016, м. Хмельницький, вул. Інститутська, 7/1. тел (0382) 72-83-63