# 2. Формалізовані та неформалізовані знання в системі культури

**Метою вивчення теми** є ознайомлення студентів з методичним інструментарієм трансформації знань для фахівців галузі видавництва.

## 2.1. Поняття і загальні особливості формалізованих і неформалізованих знань

В умовах сучасного постіндустріального суспільства як у виробничих процесах, так і в процесах навчання на робочому місці відбувається інтенсивна робота людей з різними видами знань як сукупністю відомостей у конкретній предметній області. Знання створюються як окремими людьми, так і цілими колективами. Але без індивідуумів неможливо створити знання. Організація всього лише підтримує творців і забезпечує їм необхідні для створення знання умови. Таким чином, процес створення організаційного знання, тобто знання, яке існує в організації, має на увазі розвиток організаційними методами знання, створеного індивідуумами, і виділення його в якості частини системи організаційного знання. Цей процес відбувається в рамках «спільноти взаємодії», яка перетинає внутрішньо- і міжорганізаційні рівні і межі.

Основою формування організаційного знання являються формалізовані та неформалізовані знання.

 Формалізовані знання визначаються їхньою здатністю бути швидко обробленими за допомогою відповідних алгоритмів і в результаті бути відображеними у вигляді звітів.

Формалізовані знання задаються як суворі судження, які утворюють формальну систему. F=, A-алфавіт або набір символів, у тому числі будується пропозицію формальної системи. P-синтаксичні правила побудови речення, X-аксіома (тотожно правильна пропозиція), I-семантичні правила (правило побудови нових речень з аксіом (теорем)).

 Основу неформалізованих знань становлять думки, ноу-хау та вміння працівників. Неформалізовані знання зазвичай представлені в описово-словесній формі і зазвичай не використовують будь-який математичний апарат. Важко застосовувати їх автоматично. Такі знання звичайно коректні, приблизні тощо. і часто засновані виключно на інтуїції та досвіді фахівців.

Завдання, засновані на використанні неформалізованих знань, називаються неформалізованими завданнями.

 До неформалізованих завдань відносяться завдання, що мають хоча б одну з властивостей:

алгоритмічне рішення не відоме;

завдання не можуть бути визначені в кінцевому підсумку числової форми завдання, причому правило дії над символами остаточно не визначено;

мету розв'язання задачі не можна виразити точно у вигляді цільової функції.

Особливого значення неформалізовані знання набувають в предметній області видавництва та поліграфії, в якій, наприклад, для верстання складного видання може використовуватися близько 50 формальних правил верстання (які знаходяться в різного роду стандартах) і понад 100 неформальних правил (тобто неформалізованих знань), які являють собою перелік рекомендацій стосовно мистецтва створення видання. У якості прикладів таких складних видань, які потребують неформальних правил верстання, являються видання з доповненою реальністю, видання, стилізовані під старину, 3D-видання.

 У випадку предметної області видавництва та поліграфії під терміном «знання» слід розуміти твердження про те, якою повинна бути та чи інша властивість фрагменту видання.

Для освоєння нової технології верстання видання фахівець де факто буде використовувати стандарти видавничо-поліграфічної галузі. Але внаслідок наявності великої кількості неформалізованих аспектів створення видання, які не прописані в стандартах, фахівець буде вимушений шукати відповідні неформальні вимоги на блогах і форумах. Цей процес пошуку та консультування на блогах і форумах звичайно затягується на кілька місяців, що значно уповільнює технологічний процес створення видання. Крім того, навіть швидке отримання фахівцем відповідних неформалізованих знань верстання видання має наслідком те, що ці отримані неформальні рекомендації так і залишаються «в голові» фахівця і не заносяться в інформаційну систему видавництва внаслідок відсутності методики трансформації неформалізованих знань у формалізовані правила та судження.

Отже, для ефективного здійснення навчального процесу на робочому місці в системах навчання мають бути відображеними як формалізовані, так і неформалізовані знання. Внаслідок цього актуальним завданням є формування методики трансформації знань з неформалізованих у формалізовані знання в процесі навчання на робочому місці

Аналіз епістемологічного аспекту формування знань в процесі навчання на робочому місці знайшов свого відображення в роботі Майкла Полані [1], який описав відмінності, які існують між знанням неформалізованим (tacit knowledge) і формалізованим (explicit knowledge). Неформалізовані знання (або «неявне») − приватне і залежне від ситуації і тому насилу піддається формалізації і поширенню. Формалізоване, або кодифіковане, знання може бути передано засобами формальної, систематичної мови.

Неформалізоване знання включає когнітивні і технічні елементи. Когнітивні зосереджені в тому, що Джонсон-Лейрд [3] називав «інтелектуальними моделями», в рамках яких люди, створюючи аналогії в свідомості і маніпулюючи ними, отримують робочі моделі світу.

Неформалізовані знання створюється «тут і зараз» в специфічному, практичному контексті і мають властивості, які у роботі [4] називаються Бейтсоном аналоговими. Поширення неформалізованого знання серед індивідуумів за допомогою спілкування − процес аналоговий, що вимагає певної «одночасної обробки» складнощів, що містяться в предметі обміну. Формалізоване ж знання відноситься до подій минулого або об'єктам («там і тоді») і не залежить від контексту. Його властивості названі у дослідженні [4] Бейтсоном цифровими. Згідно цього дослідження Бейтсона, особливістю формалізованого знання з точки зору способу передачі виступає його фіксація у відповідних формальних і нормативних документах. Неформалізоване знання передається у вигляді обміну думками та судженнями. Щодо способу виникнення слід зазначити, що Бейтсон вважає джерелом появи неформалізованих знань досвід і інтуїцію експертів. Формалізовані знання виникають у формі інструкцій і формальних вказівок і рекомендацій. Способами існування формалізованого знання, згідно роботи Бейтсона, виступають звіти, стандарти, нормативні документи та бази даних. Неформалізовані знання існують у формі частних думок експертів.

Складність вирішення даних завдань визначається необхідністю пошуку форми опису різних видів неформалізованих знань. На основі вибору форми опису має бути сформованим метод формалізації знань для їх трансформації у відповідні формальні судження та правила. Реалізація даного методу формалізації знань повинно відбуватися на основі використання відповідних програмних засобів автоматизації процесу перетворення знань. Формалізовані знання потім зберігаються в інформаційній системі видавництва у вигляді відповідних баз знань. Трансформація знань відбувається у формі здійснення їх формального опису та створення шаблонів суджень, правил і рекомендацій.

Трансформація знань здійснюється шляхом перетворення неформалізованого знання у формалізоване.

Зокрема, в системах електронного навчання на робочому місці для автоматизації навчального процесу слід формалізувати:

а) отримання знань;

б) набуття компетентностей;

в) оцінювання якості навчання.

Для розроблення системи трансформації знань ставляться наступні задачі:

1) занесення формалізованих знань у бази знань;

2) трансформація неформалізованих знань у формалізовані;

3) автоматизація перевірки правильності засвоєння знань на основі автоматичної перевірки правильності зверстаного матеріалу видання формальним і неформальним правилам, які були попередньо занесені в систему електронного навчання та використані верстальником при обробці видання.

Схематично процес освоєння та використання компетентностей в процесі створення «продукту», в нашому випадку видання, поданий на рис. 3.



Рис. 3 **Процес освоєння та використання компетентностей в процесі створення «продукту» (тобто видання)**

Сутність елементів і зв’язків, наведених на рис.1 міститься в наступному. Фахівець проводить роботу стосовно створення певного продукту, в процесі якої враховуються об’єкти предметної області з їх зв’язками та властивостями. В свою чергу, ці об’єкти предметної області, їх зв’язки та властивості знаходять своє відображення у відповідних особливостях створеного продукту. В процесі цієї роботи використовуються знання про те, як перетворювати, змінювати об’єкти, зв’язки та їх властивості для певного фрагменту предметної області. Ці знання можуть бути як формалізованими, так і неформалізованими. Для зручного та оперативного використання неформалізовані знання мають бути перетворені у формалізовані шляхом використання методики трансформації знань. Ця методика повинна враховувати необхідність формалізації компетентностей, знань, умінь на робочому місці, якими володіє відповідний фахівець.

**2.2. Моделі трансформації знань з формалізованих в неформалізовані і навпаки.**

Ідея «трансформації знання» в якійсь мірі схожа на модель ACT (adaptive control of thought) з галузі когнітивної психології. Відповідно до цієї моделі, для розвитку когнітивних навичок все декларативне знання, що є знанням формалізованим, має бути перетворено в процедурне знання, тобто знання неформалізоване. Поняття декларативного знання й формалізованого знання та поняття неформалізованого знання й процедурного знання є тотожніми.

 Існують чотири способи трансформації знань (рис. 4) [2], які стосуються самих знань, а не людини як носія знань та способів поведінки:

1) з неформалізованого в неформалізоване - соціалізація;

2) з неформалізованого в формалізоване - екстерналізація;

3) з формалізованого в формалізоване - комбінація;

4) з формалізованого в неформалізоване - інтерналізація.



Рис. 4. **Чотири моделі трансформації знання з формалізованого в неформалізоване і навпаки**

Організаційне знання створюється та використовується в процесі навчання на робочому місці по спіралі, починаючи з індивідуального рівня, і при розширенні взаємодії проходить рівні секції, відділу, сектору та організації (рис. 5). Прикладом організаційного знання на рівні відділу є правила верстання видання та його перевірки. Ці правила використовуються у межах редакційно-видавничого відділу.



Рис. 5 **Спіраль створення та використання знання в процесі навчання на робочому місці**

Наведемо приклад формалізованих і неформалізованих знань для задачі верстання видання.

Формалізоване знання – «В текстах типу «Х» слід вживати на початку абзацу буквицю виду «Y».

Неформалізоване знання – «В текстах типу «Х» рекомендується (або бажано) виділяти буквицю (якщо вона є) іншим кольором (яким – не вказується)».

Наступний шар неформалізованого знання – «Якщо видання дитячого спрямування, то краще буквиця «Y» з кольором «назва кольору»; якщо текст історичного спрямування, то буквиця вибирається відповідно з особливостями передачі кольором певної емоції або почуття, яке несе в собі фрагмент тексту».

Процес трансформації знань є досить складною процедурою, тому виникає необхідність у пошуку способів його автоматизації, які дадуть можливість здійснення швидкого і якісного перетворення, при цьому витративши на сам процес автоматизації найменшу кількість часу та зусиль.

Дану задачу можна вирішити двома способами:

1. створити спеціальні програмні засоби, що описують кожне неформалізоване знання;
2. знайти форму опису знань для того, щоб на їх основі автоматично формувати програмні засоби для створення шаблонів або перевірки готових формалізованих знань.

У першому випадку необхідно проаналізувати всі неформалізовані знання і для кожного виду таких знань створити програмний засіб, що здійснює трансформацію. Даний підхід передбачає тільки описання кожного знання програмною мовою та задання перевірки за допомогою операторів умов. Але, з іншого боку, таких програмних засобів буде велика кількість і на написання програмного коду, що описує всі правила буде витрачено багато часу.

Оскільки метою автоматизації є прискорення всіх процесів, необхідно розглянути інші способи автоматизації, що дають можливість виграти час на підготовці видань.

Що стосується другого способу, його суть полягає в описі думок, ноу-хау та вмінь у вигляді формалізованих правил деякої формалізованої мови і в автоматичному перетворенні вже формалізованих правил у програмний код.

Даний спосіб є більш швидким і автоматизує сам процес створення програмних засобів формалізації знань, тому його застосування є оптимальним для вирішення задачі трансформації знань.

Першим етапом створення такого програмного засобу є формалізація наявних знань.

У будь-якому судженні присутня конкретна думка. Вона виступає формою вираження дійсності у свідомості людини. Ця форма являє собою яке-небудь висловлювання про предмети їх властивості і стан, а також про відносини між ними.

Дане твердження дозволяє сформулювати робочу гіпотезу наукового дослідження: правило конкретної предметної області, представлене у вигляді судження, може бути формалізоване на основі використання апарата логіки суджень.

Кожне судження складається із трьох основних елементів суб'єкта, предиката і зв'язки. Кожний із цих членів судження обов'язково є або мається на увазі у всіх судженнях.

Розподіл неформалізованих знань на складові частини згідно з логікою суджень, дозволить надалі виділити основні суб'єкти в цих правилах і їх властивості, що у свою чергу стане основою для трансформації даних знань.

**2.3. Метод формалізації знань**

Проведене емпіричне дослідження дало можливість сформулювати метод формалізації знань, що дозволяє представити останні у вигляді придатному для автоматизованої перевірки коректності трансформації.

 Метод формалізації знань складається з таких етапів.

1) Виявлення суджень про конкретну предметну область;

а) Виявлення вимог конкретної предметної області стосовно формування конкретних видів неформалізованих знань;

б) Виявлення суджень у розрізі предметної області.

2) Дескриптивна формалізація знань.

Дескриптивна формалізація знань припускає прямий опис об'єктів за допомогою термінів. Мета даного етапу формалізації – представлення даних у компактному, більш точному й однозначному вигляді. У цьому випадку, у якості термінів обрані терміни апарата логіки суджень.

Оскільки всі обрані неформалізовані знання є судженнями, необхідно в даних судженнях виділити основні елементи.

Дескриптивна формалізація знань відбувається у спосіб, який наведено далі.

а) Виділяємо суб'єкти, предикати, зв'язки та квантори суджень.

Суб’єкт – це то, про що говориться в судженні, предикат виражає значення про ознаку предмета думки, тобто те, що говориться про суб'єкт судження, зв’язка – виражає відношення, що встановилося в судженні між суб'єктом і предикатом, а квантор указує, відноситься судження до всього або до частини об'єму поняття, що виражає суб'єкт.

**б) Визначаємо вид судження (просте або складне).**

При цьому, якщо судження можна розкласти тільки на поняття, виходить воно є простим судженням, якщо судження складається з декількох вихідних суджень, з'єднаних у рамках даного складного судження логічними сполучниками, виходить воно є складним судженням.

**в) За допомогою логічного аналізу визначаємо вид складного судження.**

Якщо суджень більше одного, уточнюємо в яких логічних відносинах (кон'юнкція, диз'юнкція, імплікація, еквівалентність або заперечення) перебувають між собою судження, з урахуванням пріоритету логічних операцій. Виписуємо судження та логічні операції між ними, формуючи логічний вираз.

Наприклад, складне судження може бути представлено в такому вигляді:

{просте судження} І {просте судження} АБО {просте судження}

**3) Логічна або дедуктивна формалізація знань.**

Дана формалізація передбачає відображення загальних взаємозв'язків між поняттями або судженнями за допомогою дедуктивно-упорядкованих систем символів. Дедуктивна формалізація дозволяє уточнити та систематизувати змістовне представлення конкретних видів знань.

Даний етап формалізації складається з підетапів, що описані далі.

**а) Побудова алфавіту або складання вихідних знаків і символів.**

Формуємо словники суб'єктів, предикатів, кванторів і зв'язок.

Словник суб'єктів (словник формується шляхом вибору унікальних суб'єктів судження й опису їх формалізованою мовою).

Словник предикатів. Даний словник будується за допомогою розчленовування предикатів на елементи: властивості, відносини, значення властивостей.

При цьому формулювання предикатів повинно будуватися в стверджувальній формі. Словник відношень між властивістю і значенням (повний перелік) представлений у табл. 1. Значення незавжди є порівнюваними (наприклад, колір). В цьому випадку варто залучувати інструментарій неформальної логіки.

Таблиця 1

**Словник відношень**

|  |  |
| --- | --- |
| **Відношення** | **Опис природньою мовою** |
| Greater | Більше |
| Less | Менше |
| Equal | Рівно |
| Greaterorequal | Більше або рівно |
| Lessorequal | Менше або рівно |
| Notequal | Не рівно |

**б) Формування правил побудови виразів на підставі обраного алфавіту.**

Після того, як сформовані словники, на їхній підставі будується опис усіх видів знань, для цього необхідно розробити структуру опису. Структура опису неформалізованих знань, розроблена згідно апарата логіки суджень, представлена на рис. 6.

Така структура неформалізованих знань дозволяє описати їх форму так, щоб при формалізації не було втрачено жодного елементу з судження.

У процесі розробки механізму автоматизованої перевірки коректності трансформації знань, був сформульований ідеальний кінцевий результат: механізм повинен автоматично генерувати програмний код для перевірки конкретного знання з його наступним виконанням і генеруванням звіту про виявлені невідповідності.

Суб’єкт

Предикат

Квантор

Зв’язка

Властивість

Відношення

Значення

Опис знання

Рис. 6. **Структура опису неформалізованих знань**

 Для реалізації механізму із зазначеною вище функціональністю, він повинен містити в собі наступні елементи:

1. база формалізованих знань конкретної предметної області;
2. перетворювач – елемент механізму, що дозволяє описати алгоритм перетворення формалізованого знання у фрагмент програмного коду;
3. виконавець – програма, що генерує програмний код для перевірки знання на відповідність заданому користувачем набору правил предметної області, що і забезпечує його наступне виконання з формуванням звіту про виявлені невідповідності.

Структура розробленого механізму представлена на рис. 7.

Розглянемо більш докладно кожний елемент даного механізму.

База формалізованих знань. Для автоматизованого перетворення правил конкретної предметної області в програмний код, перші повинні бути представлені у формалізованому вигляді.

Використання формалізації дозволить виявити основні форми, якими можуть бути описані правила тієї чи іншої предметної області, і надалі працювати вже з елементами отриманих форм. У якості елементів формалізованих правил доцільно використовувати: суб'єкт правила, властивості суб'єкта, а також взаємозв'язки між суб'єктом і його властивостями.



Рис. 7. **Структура механізму автоматизованої перевірки формалізованих знань на відповідність правилам конкретної предметної області**

Після того, як правила формалізовані природньою мовою, необхідно описати їх формалізованою мовою.

Формалізована мова являє собою будь-яку сукупність, деяким чином спеціалізованих мовних засобів з (більш-менш) точно фіксованими правилами утворення «виразів» (синтаксис формалізованої мови) і приписування цим виразам певного змісту (семантика). У такому вживанні термін «формалізована мова» не припускає ніяких спеціальних обмежень ні на синтаксичну структуру, ні на семантичні правила, ні на призначення такої мови.

 Під формалізованою мовою в логіці розуміють інтерпретоване числення, тобто деяку формальну систему разом з її інтерпретацією. Використання формалізованої мови – характерна риса математичної логіки, яку часто і визначають як «предмет формальної логіки, досліджуваний за допомогою побудови формалізованих мов» [4].

Аналіз різних видів мов формалізації, дозволив виявити, що найбільше підходить для вирішення даного завдання мова розмітки XML.

Вибір даної мови для формалізації правил предметної області обумовлений її достоїнствами у порівнянні з іншими мовами:

1. XML – мова розмітки, що дозволяє стандартизувати вид файлів-даних, використовуваних комп'ютерними програмами, у вигляді тексту, зрозумілого людині;
2. XML підтримує Юнікод;
3. у форматі XML можуть бути описані такі структури даних як записи, списки та дерева;
4. XML – це самодокументований формат, який описує структуру й імена полів так само як і значення полів;
5. XML має строго визначений синтаксис і вимоги до аналізу, що дозволяє йому залишатися простим, ефективним і несуперечливим. Одночасно із цим, різні розробники не обмежені у виборі виразних методів (наприклад, можна моделювати дані, поміщаючи значення в параметри тегів або в тіло тегів, можна використовувати різні мови та нотації для іменування тегів і т. д.);
6. XML – формат, заснований на міжнародних стандартах;
7. ієрархічна структура XML підходить для опису практично будь-яких типів документів, крім аудіо та відео мультимедійних потоків, растрових зображень, мережних структур даних і двійкових даних;
8. XML являє собою простий текст, вільний від ліцензування і яких-небудь обмежень;
9. XML не залежить від платформи;
10. XML не накладає вимог на порядок розташування атрибутів в елементі та вкладених елементів різних типів, що суттєво полегшує виконання вимог зворотної сумісності;
11. існує стандартний механізм перетворення XSLT, реалізації якого вбудовані в браузери, операційні системи, веб-сервери та системи управління знаннями. Внаслідок цього опис знань на основі формату XML може бути легко та швидко адаптований для використання в різних системах управління знаннями (наприклад, в редакторі онтологій Protégé);
12. XML підтримується на низькому апаратному, мікропрограмному і програмному рівнях у сучасних апаратних рішеннях.

Перетворювач. Даний елемент механізму, призначений для опису алгоритму перетворення формалізованого знання у фрагмент програмного коду. Слід зазначити, що для перетворювача умови необхідності і достатності будуть забезпечені в тому випадку, коли є відомою семантика всіх форм опису правил конкретної предметної області та для кожної з них буде описаний алгоритм генерування програмного коду. Також слід зазначити, що можливість генерування програмного коду відповідного до об'єктної моделі тієї або іншої предметної області й (або) використання різних мов програмування забезпечується шляхом створення і використання різних перетворювачів.

У якості перетворювача формалізованих правил може виступити мова перетворень XML-документів – XSLT. Вибір мови XSLT для задачі автоматизованої перевірки формалізованих знань на відповідність правилам конкретної предметної області обумовлений наступними обставинами. Мова XSLT є декларативною, а не процедурною. Замість визначення послідовності операторів, що виконуються, ця мова визначає правила, які будуть застосовуватися під час перетворення. Саме перетворення ведеться по фіксованому алгоритму. Елементи в просторі імен XSLT (що звичайно мають префікс xsl:) трактуються як інструкції і мають спеціальну семантику, яка вказує на те, як вони повинні інтерпретуватися. Одні призначені для додавання вузлів у результуюче дерево, інші є керуючими конструкціями. XSLT здатний перетворити формалізовану мову XML у скриптову мову.

Виконавець. Виконавець являє собою програму, яка з погляду користувача, забезпечує перевірку трансформованого знання на відповідність правилам предметної області. Основними функціями даної програми є:

* формування в процесі взаємодії з користувачем набору правил і вимог предметної області, яким повинне відповідати трансформоване знання;
* вибір перетворювача, відповідного до інформаційної системи, та вибір виду трансформованого знання, що вимагає перевірки на відповідність правилам предметної області;
* генерування програмного коду перевірки трансформованого знання на відповідність правилам предметної області;
* здійснення перевірки трансформованого знання за допомогою згенерованого програмного коду і формування звіту про виявлені невідповідності.

Однією з найбільш важливих функцій даного елемента є генерування програмного коду.

Генерування програмного коду перевірки трансформованого знання на відповідність правилам предметної області може здійснюватися за допомогою будь-якої мови програмування. У результаті аналізу різних мов програмування для реалізації цього завдання була обрана скриптова мова Javascript. Вибір даної мови в рамках даного дослідження обумовлений наступними обставинами:

1. Javascript забезпечує програмування без ризику дестабілізувати систему. На відміну від плагінів, скрипти інтерпретуються, а не компілюються, неправильно написаний скрипт виведе діагностичне повідомлення, а не приведе систему до краху;
2. Javascript створює виразний код. Ця скриптова мова має власний проблемно-орієнтований набір команд, і один рядок скрипта може робити те ж, що десятки рядків, написані традиційною програмною мовою. Як наслідок, на скриптовій мові може писати програміст дуже низької кваліфікації;

3) Javascript створює можливості кросплатформеності, тобто виконання у браузерах під різними операційними системами.

Для того, щоб прискорити сам процес автоматизації необхідно створити програмний засіб, який буде автоматично генерувати необхідні скрипти для перевірки трансформованих знань.

Для вирішення завдання потрібне створення програмного засобу, який буде формувати файл зі скриптом з формалізованих правил предметної області і перетворювача необхідно вибрати мову програмування, здатну здійснити дане перетворення. У якості такої мови може виступити мова програмування C#.

Таким чином, об'єднання описаних компонентів дозволить створити новий механізм для автоматизованої перевірки трансформованих знань на відповідність правилам конкретної предметної області і дасть можливість удосконалити процес трансформації знань. Застосування розробленого механізму можливо як для перевірки, так і для створення шаблонів неформалізованих знань із урахуванням правил конкретної предметної області. Впровадження даного механізму в електронне навчання на робочому місці дозволить скоротити часові витрати на трансформацію і перевірку знань, а також забезпечити високу якість цього процесу, при цьому автоматичне генерування правил для перевірки знань скорочує також час на розробку прототипу системи перевірки трансформованих знань на основі даного механізму.

Даний прототип розроблено на основі запропонованого механізму автоматизованої перевірки трансформованих знань на відповідність правилам конкретної предметної області.

Основні елементи такого механізму це:

1. база формалізованих технічних правил перевірки трансформованих знань на відповідність правилам конкретної предметної області;
2. перетворювач;
3. виконавець.

Розглянемо більш докладно структуру перетворювача. Оскільки в якості мови для перетворення обрана мова XSLT, перетворювач будується згідно із правилами побудови шаблонів-XSLT. Структура розробленого перетворювача представлена на рис. 8.

Шаблон для опису суб’єктів

Шаблон для опису предикатів

Шаблон для опису функцій

Шаблон для опису властивостей

Шаблон для опису відношень

Шаблон для опису правила

Шаблон для опису значень

Рис. 8 **Структура перетворювача**

Наступний етап – розробка програмного засобу, який буде взаємодіяти з базою формалізованих правил предметної області, перетворюючи дані правила в скрипти для перевірки видань.

Даний програмний засіб повинен мати наступну функціональність:

1. вибір трансформованого знання для перевірки;
2. вибір системи перетворення (у цьому випадку припускає вибір відповідного перетворювача у форматі XSLT);
3. вибір виду знання;
4. формування набору правил для перевірки трансформованих знань на відповідність правилам конкретної предметної області (відбувається шляхом вибору виду знання, а також вибору додаткових параметрів перевірки, описаних у форматі XML);
5. перетворення Xml-формату у формат Javascript;
6. створення файлу, що містить згенерований код для перевірки трансформованого знання;
7. відображення повідомлення користувачеві про помилки в трансформації знання.

Інтерфейс прототипу системи автоматизованої перевірки трансформованих знань на відповідність правилам конкретної предметної області наведено на рис. 9.



Рис. 9. **Інтерфейс прототипу інформаційної системи автоматизованої перевірки трансформованих знань на відповідність правилам конкретної предметної області**

Цей розроблений прототип являє собою фрагмент системи електронного навчання для фахівців предметної області видавництва та поліграфії. База знань для цієї системи формується фахівцем e-learning, який опрацьовує неформалізовані та формалізовані знання.

Дана функція прототипу була реалізована за допомогою створення додаткових класів на мові C#, які стали шаблоном для формування нових правил предметної області користувачем.

Після того, як користувач обирає основні правила предметної області, такі як: суб’єкт, предикат, зв’язка і квантор, він може створити правило, не вдаючись до деталей роботи даного механізму. Коли правило буде створено, його можна легко додати до бази. Потім запустити генерування скрипту і перевірити трансформовані знання на відповідність правилам конкретної предметної області. Вікно програми, у якому створюється нове правило, наведено на рис. 10.



Рис. 10**. Створення нового правила**

Отже, створення нового правила за зазначеною схемою є шаблоном для формування нових правил предметної області користувачем.

## 2.4. Приклад застосування методики формалізації знань для галузі видавництва та поліграфії.

## У якості приклада застосування запропонованої методики формалізації знань в процесі навчання на робочому місці в умовах галузі видавництва та поліграфії наведемо створені на її основі методичні рекомендації з формалізації правил верстання видань та перевірки макету на відповідність даним правилам. Користувачем даних методичних рекомендацій виступає дизайнер-верстальник. У якості правил предметної області в даному прикладі являються правила верстання видань.

Так, в результаті аналізу правил верстання видань були сформовані рекомендації щодо їх формалізації та перевірки макету на відповідність даним правилам.

1) Виокремлення правил верстання видань з-поміж іншої інформації відносно підготовки та випуску видань.

Кожен нормативний документ містить в собі досить різноманітну інформацію щодо видань. Такі документи можуть містити загальні технічні вимоги, терміни та визначення понять, нормативні посилання, класифікації видань, правила приймання, методи контролювання, транспортування та зберігання видань, гарантії виробника, вимоги безпеки та охорони навколишнього середовища.

Правила верстання видань завжди знаходяться у розділі технічні вимоги, якщо такий вказано в нормативному документі. Але даний розділ може також містити й вимоги до фотоформ, до поліграфічного виконання, матеріалів, пакування та маркування. Тому для того, щоб виокремити правила верстання видань, необхідно виявити характеристики таких правил, які допоможуть виділити їх серед інших.

Характеристики, за якими можна виявити правила верстання видань, представлені нижче.

а) Кожне окреме правило верстання видань співвідноситься з одним суб’єктом, який належить до множини таких параметрів видань:

формат сторінки складання;

розмір середника (середників) при багатошпальтовому складанні;

кількість рядків у полосі (шпальті);

гарнітура, кегель, інтерліньяж, накреслення, відбивка та оформлення основного тексту;

гарнітура, кегель, інтерліньяж, накреслення, відбивка та оформлення колонтитулу;

гарнітура, кегель, інтерліньяж, накреслення, відбивка та оформлення колонцифри;

розмір спуску на початкових сторінках;

розмір внутрітекстових відбивок між різними текстами;

параметри форматування символів;

параметри форматування абзаців;

характер виключання рядків.

б) Кожне окреме правило верстання видань є логічно-відокремленим реченням, що має закінчену думку.

Наприклад: текст у всьому обсязі видання, окрім заголовків, має бути відтворений шрифтами не більше ніж чотирьох гарнітур або чотирьох накреслень однієї гарнітури.

в) Кожне правило верстання видань описано в наказовому способі.

Наприклад: кегель шрифту основного тексту, відтвореного шрифтом на латинській графічній основі, має бути не менше ніж 16 пунктів.

2) Формалізація правил верстання видань.

Перед початком формалізації правил верстання видань, необхідно записати його природньою мовою в поле програми перевірки видань на відповідність правила верстання видань. Для цього необхідно відкрити програму перевірки макетів на відповідність правила верстання видань та натиснути кнопку «Створити нове правило» (див. рис. 7).

Після того, як виявлено та записано окреме правило верстання видань, необхідно його формалізувати згідно з апаратом логіки суджень. Для цього необхідно виявити у правилах верстання видань суб’єкт судження.

а) Виявлення суб’єкта судження.

Щоб виокремити суб’єкт правил верстання видань, необхідно знайти в правилі предмет думки, тобто те, про що говориться в даному правилі.

Наприклад: кегель шрифту основного тексту, відтвореного шрифтом на латинській графічній основі, має бути не менше ніж 16 пунктів.

В даному випадку мова йде про основний текст. Після того, як знайдено суб’єкт, необхідно внести його в програму перевірки макету видання.

Для цього у вікні додавання нового правила, в спадаючому списку суб’єктів вибрати той, про який йдеться в даному правилі.

Якщо, у списку суб’єктів немає даного суб’єкта, необхідно звернутися до розробників, для того, щоб додати його до бази.

б) Виявлення властивостей суб’єкта.

Після того, як виявлено суб’єкт, необхідно визначити умови за якими буде перевірятись даний суб’єкт. Щоб сформувати умову перевірки макету, необхідно виділити властивості суб’єкту, значення даних властивостей, та відношення між властивістю та значенням, які потім також будуть задані в програмі.

Розглянемо даний етап формалізації на попередньому прикладі.

В даному випадку, властивістю тексту є «кегель шрифту», значення властивості – «16 пунктів», а відношення між властивістю та значенням – «не менше».

Крім того, дане правило містить ще одну умову – шрифт має бути на латинські графічній основі, де «шрифт» – це властивість, значення властивості – «на латинській графічній основі», відношення між властивістю та значенням – «має бути».

Після виявлення цих даних, необхідно також внести їх в програму перевірки макетів видань (наприклад, PerfectPixel або Adobe Illustrator), обираючи властивості та відношення між ними зі спадаючих списків, та вводячи значення властивостей у вільні комірки значень з клавіатури. Якщо, у списку властивостей немає потрібної властивості, необхідно звернутися до розробників, для того, щоб додати її до бази.

Кожне правило верстання видань може містити декілька умов, які можна додавати в програмі за допомогою кнопки «+».

в) Виявлення кванторів суджень.

Кванторами суджень у даному випадку являються види видань. Тобто, кожне окреме правило верстання може відноситись до окремого виду видань, тому при введенні нового правилу верстання в програму перевірки макетів видань на відповідність правилам верстання видань, необхідно також обрати до якого виду видань буде відноситись дане правило.

Для цього необхідно обрати із запропонованої класифікації вид видання, яке буде перевірятись на відповідність даному правилу.

3) Перевірка макету на відповідність правилам верстання видань.

Коли нове правило верстання видань сформовано, необхідно додати його до існуючої бази правил та перевірити макет на відповідність даному правилу, або ж тим правилам, які вже додано до бази інформаційної системи автоматизованої перевірки трансформованих знань на відповідність правилам конкретної предметної області (див. рис. 1.6). Для цього необхідно натиснути на кнопку «Генерувати правило», потім на кнопку «Додати до списку». Потім необхідно перейти в попереднє вікно, обрати видання за допомогою кнопки «Огляд», систему верстання та вибрати ті правила верстання видань, на відповідність яким буде перевірятись макет і натиснути по черзі кнопки «Генерувати скрипт» та «Виконати перевірку». Після чого макет буде перевірено та представлено звіт про знайдені помилки в макеті.

Таким чином, розглянутий приклад в умовах навчання дизайнера-верстальника на робочому місці ілюструє практичну доцільність запропонованої методики трансформації знань.

Таким чином, формалізовані і неформалізовані знання становлять у навчанні на робочому місці ключову основу освітнього процесу. Формалізовані знання, передані у вигляді формальних, систематичних мовних конструкцій, дозволяють швидко й ефективно здійснювати онтологічний опис предметної області та використовувати перспективні методи та засоби обробки інформації у системах електронного навчання, зокрема методи штучного інтелекту. Неформалізовані знання дозволяють фіксувати власний досвід, використовувати вміння та практичні думки, формувати перспективні інноваційні ідеї стосовно певної предметної області електронного навчання на робочому місці. Запропонована трансформація знань дозволяє ефективно використовувати в системі електронного навчання як у цілісному навчальному комплексі як формалізовані, так і неформалізовані знання та оперативно здійснювати формування певних комплексів знань на індивідуальному, груповому, організаційному та між організаційному рівнях. Завдяки автоматизації перевірки трансформованих знань на відповідність правилам конкретної предметної області з’являється можливість швидкого й якісного перетворення знань і, відповідно, їх формалізації стосовно вимог певної предметної області.