

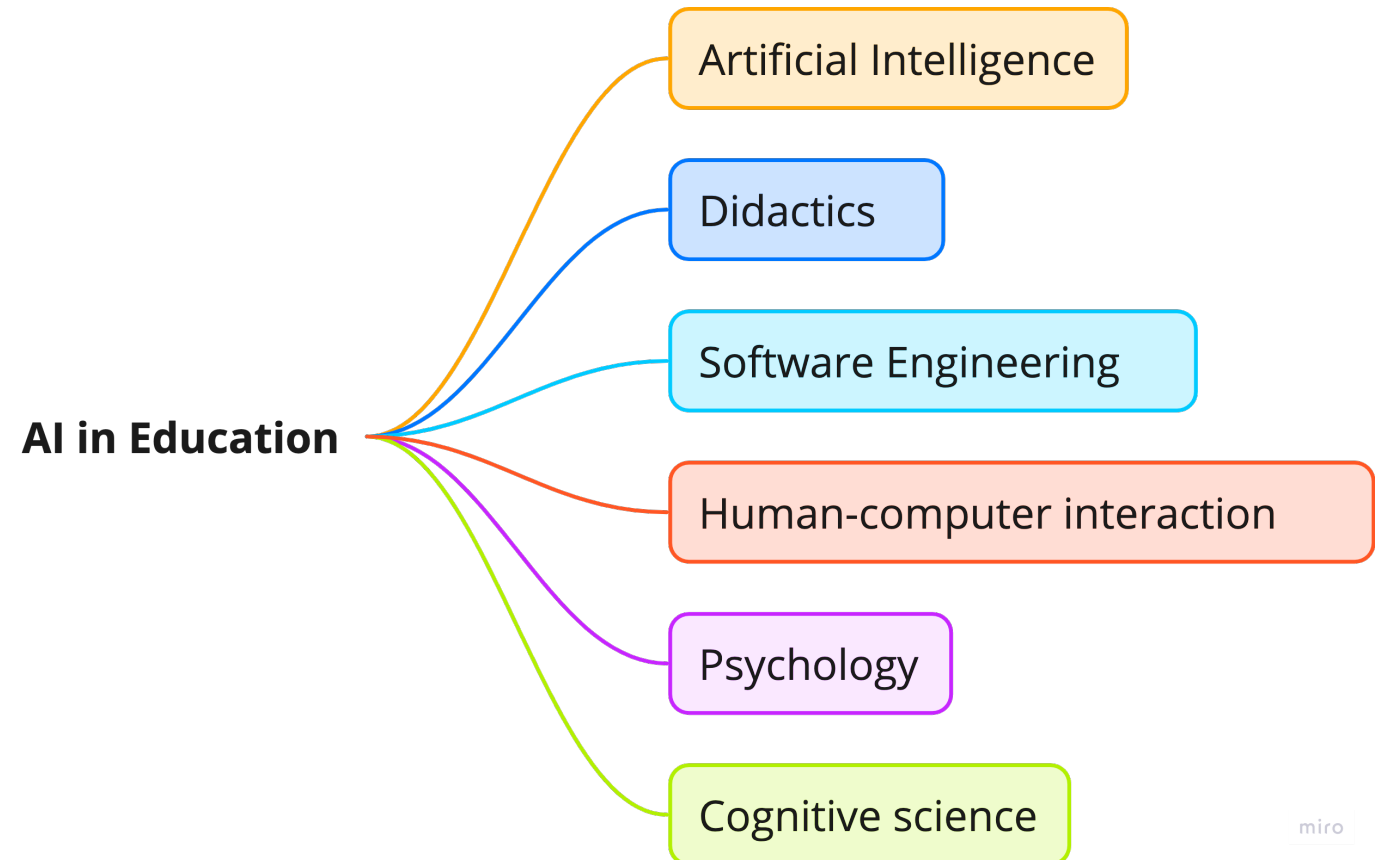
# INTELLIGENT EDUCATIONAL WEB SYSTEMS WITH INTERACTIVE CONCEPT MAPS

Sergiy Tytenko, PhD

Associate professor at American University Kyiv

# ARTIFICIAL INTELLIGENCE IN EDUCATION

The focus:  
**Development of interactive and adaptive learning environments**



---

# Related systems

Recommender systems

Search systems

Learning Management Systems

Help systems

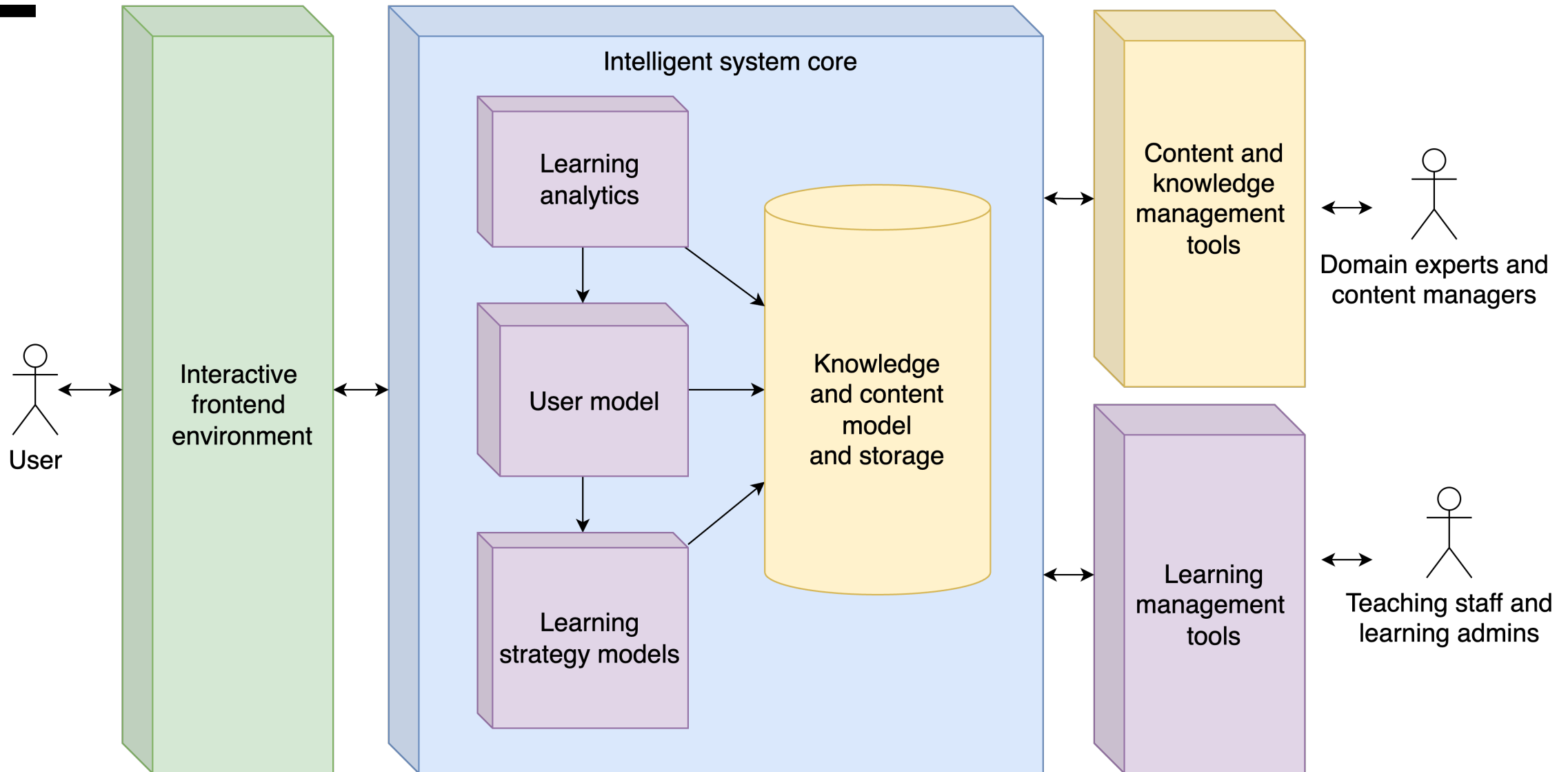
Higher education systems

Adaptive educational systems



# **DIRECTIONS IN ADAPTIVE EDUCATIONAL SYSTEMS AND AI**

# High-level architecture of intelligent web-based learning system



# Intelligent Tutoring Systems (ITS)



ITS are computer-based systems that provide personalized instruction and feedback to students. They adapt the learning materials and pace to match the individual student's abilities and progress.



Have strict style of instructing. Have a full control over the learning process



Were especially popular in early years of AI in education

# Adaptive learning web environments



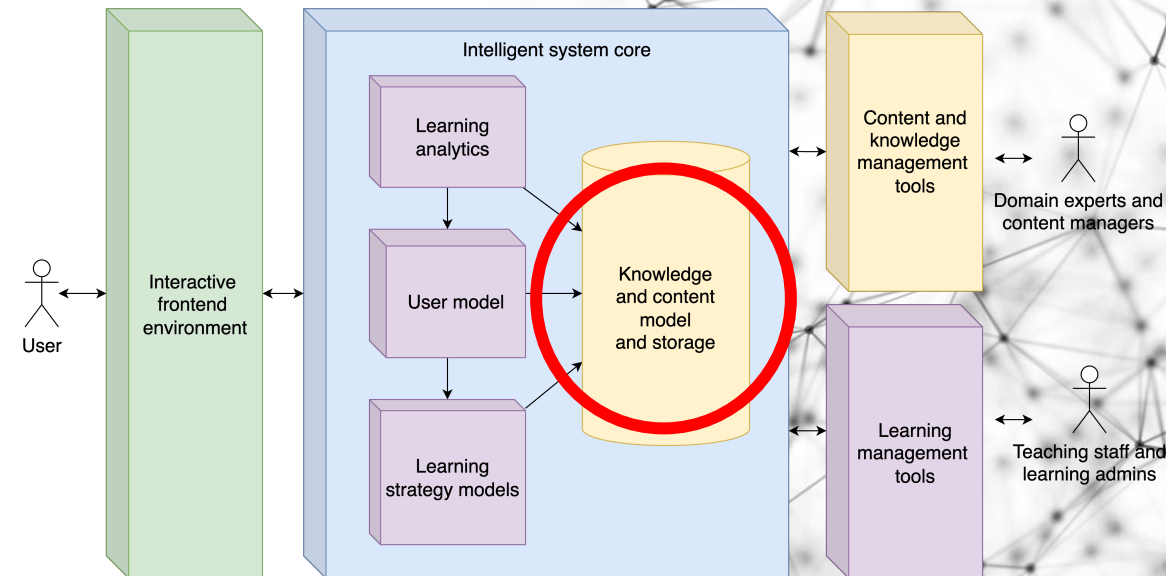
Web systems with aim at supporting learners in acquiring knowledge and skills in a particular learning domain. The goal is to enhance the individual learning process with respect to speed, accuracy, quality and quantity of learning.



Tends to give more freedom to user compared to ITS.

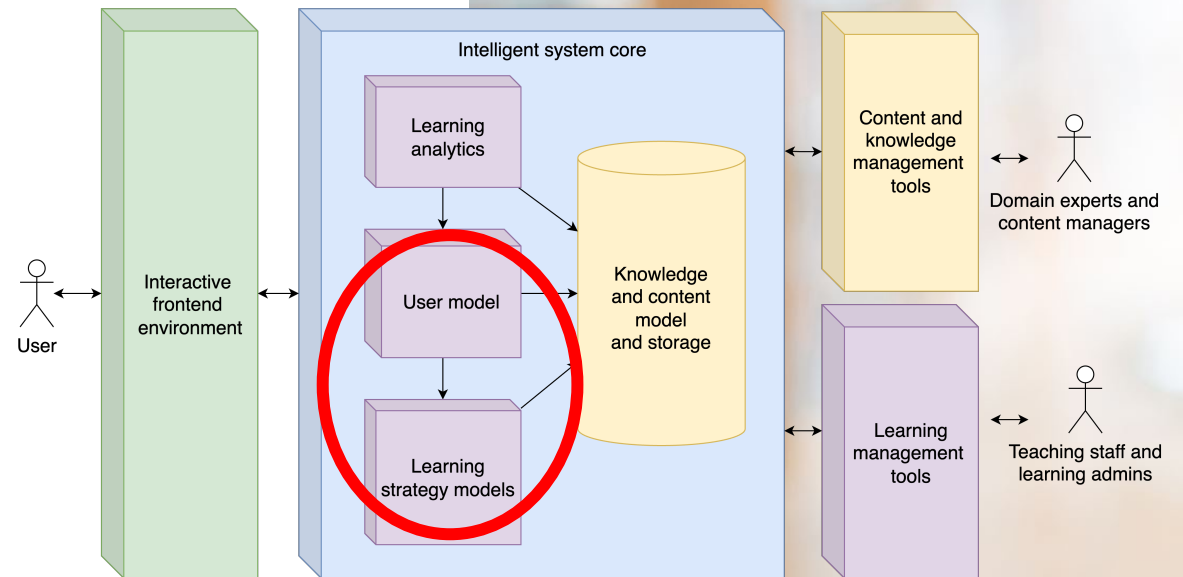
# Domain knowledge and learning content representation

- Knowledge modeling for educational systems
- Semantic modeling of learning content
- Ontological modelling
- Model knowledge in order to present it to humans



# User and Teaching Modeling

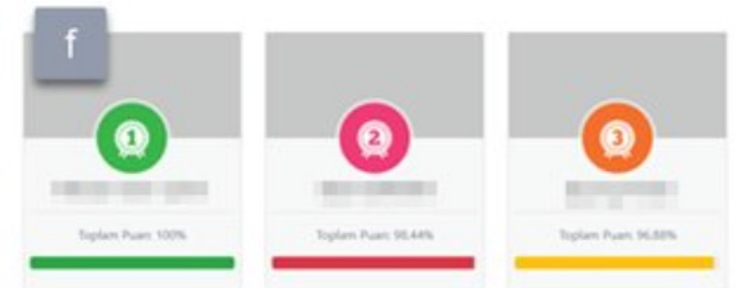
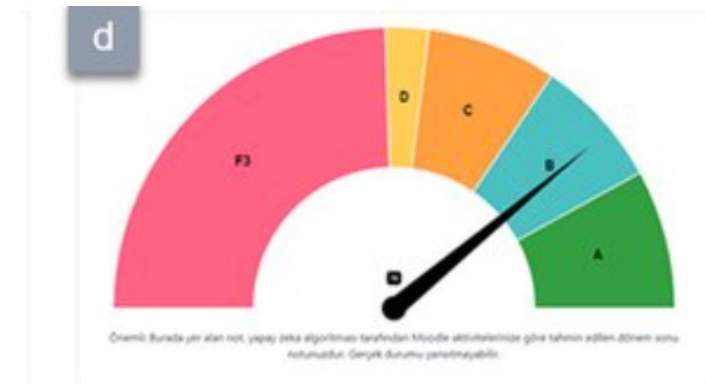
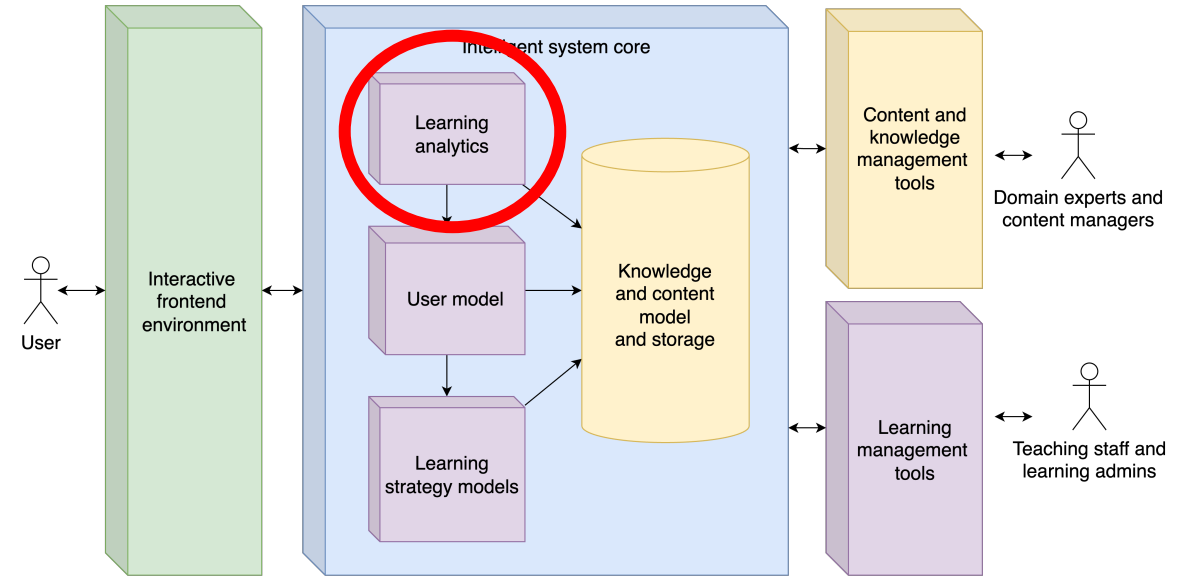
- Models of learners, including open learner models
- Intelligent tutoring, modelling of learning and teaching strategies



# Learning Analytics Dashboards

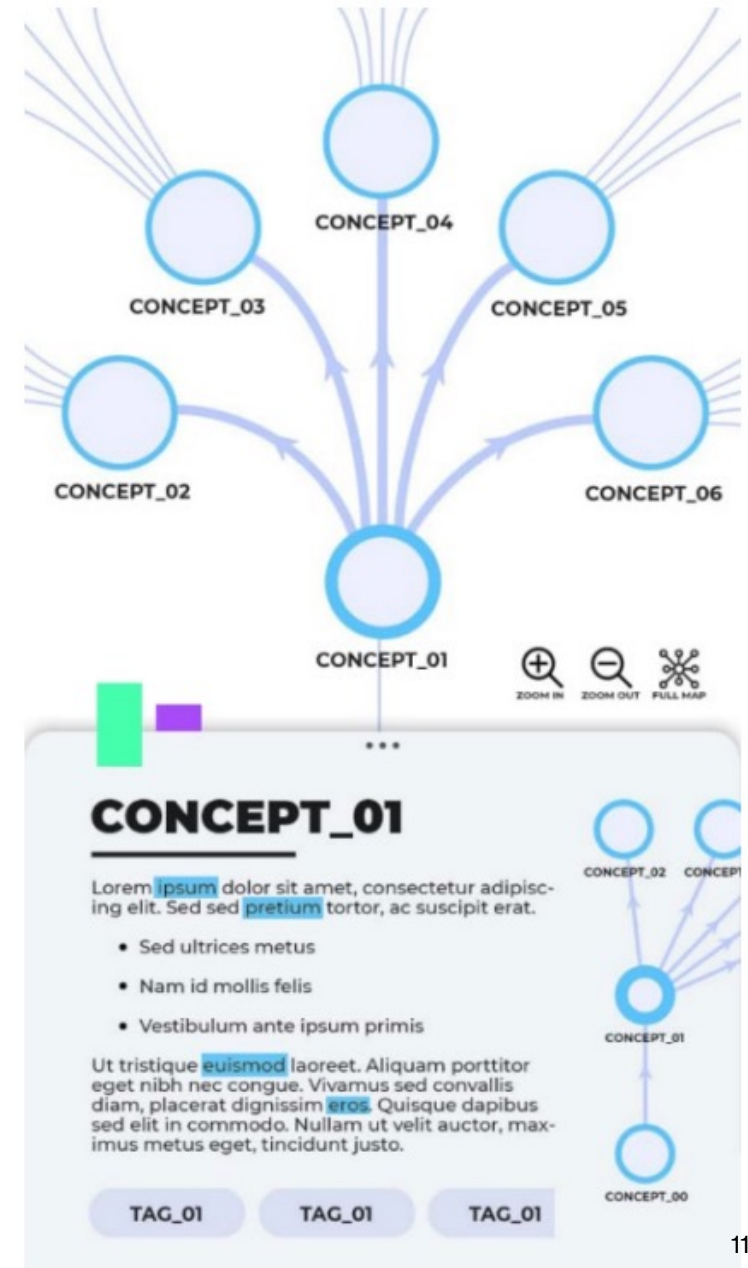
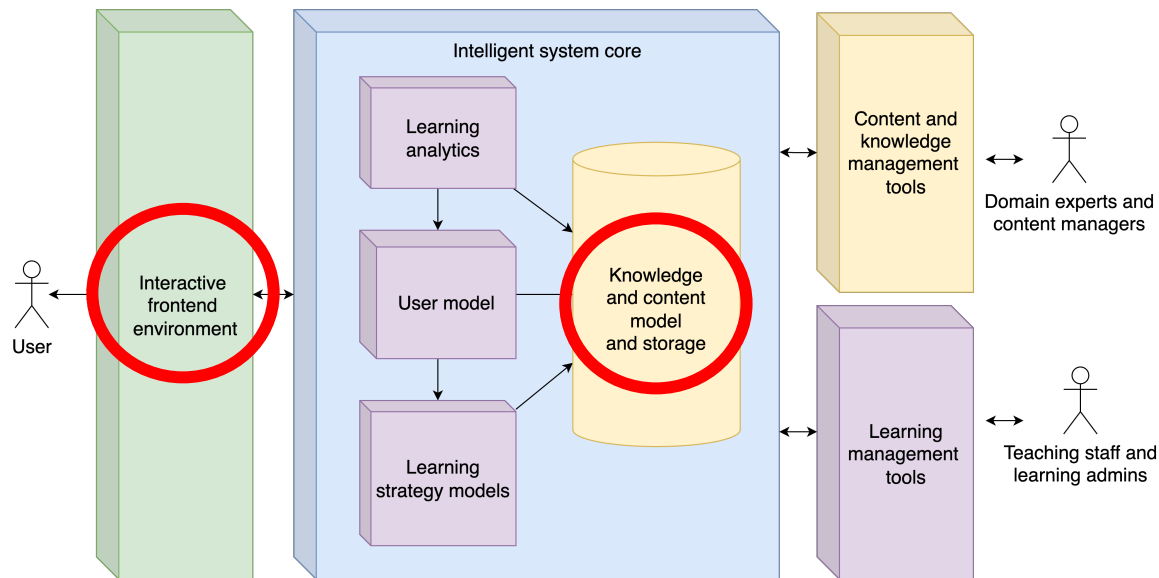
- Learning analytics dashboards are tools or platforms used in the field of education to collect, analyze, and present data related to student performance and learning outcomes.
- These dashboards offer a visual representation of data to educators, administrators, and students, helping them make informed decisions about teaching and learning

Akçapınar, G., Hasnine, M.N. (2021). Development and Evaluation of a Student-Facing Gamified Learning Analytics Dashboard. In: Sahin, M., Ifenthaler, D. (eds) Visualizations and Dashboards for Learning Analytics. Advances in Analytics for Learning and Teaching. Springer, Cham. [https://doi.org/10.1007/978-3-030-81222-5\\_13](https://doi.org/10.1007/978-3-030-81222-5_13)



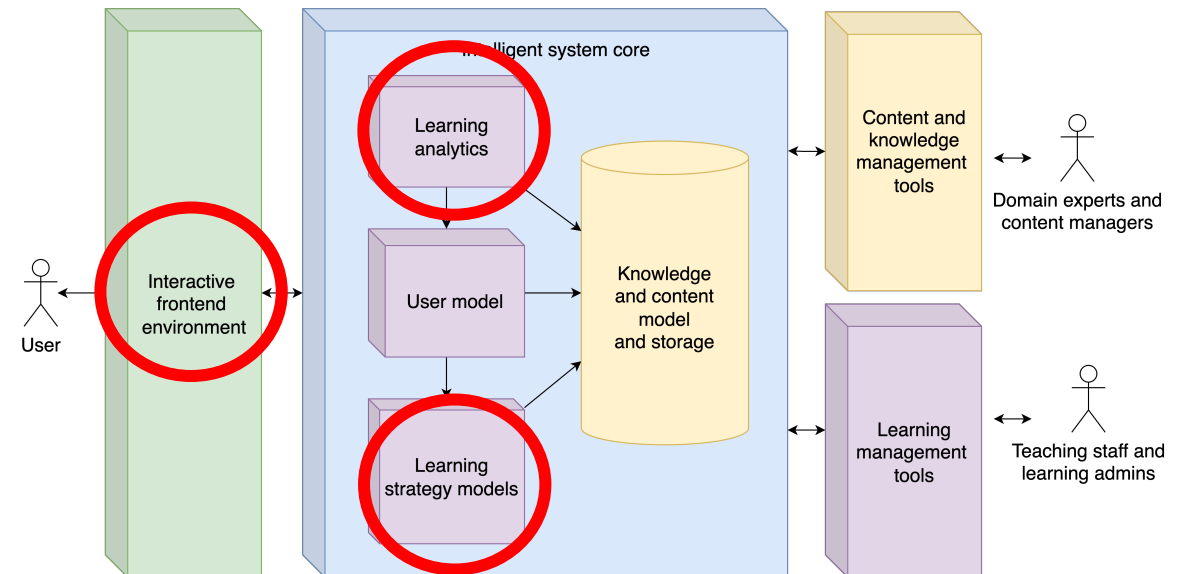
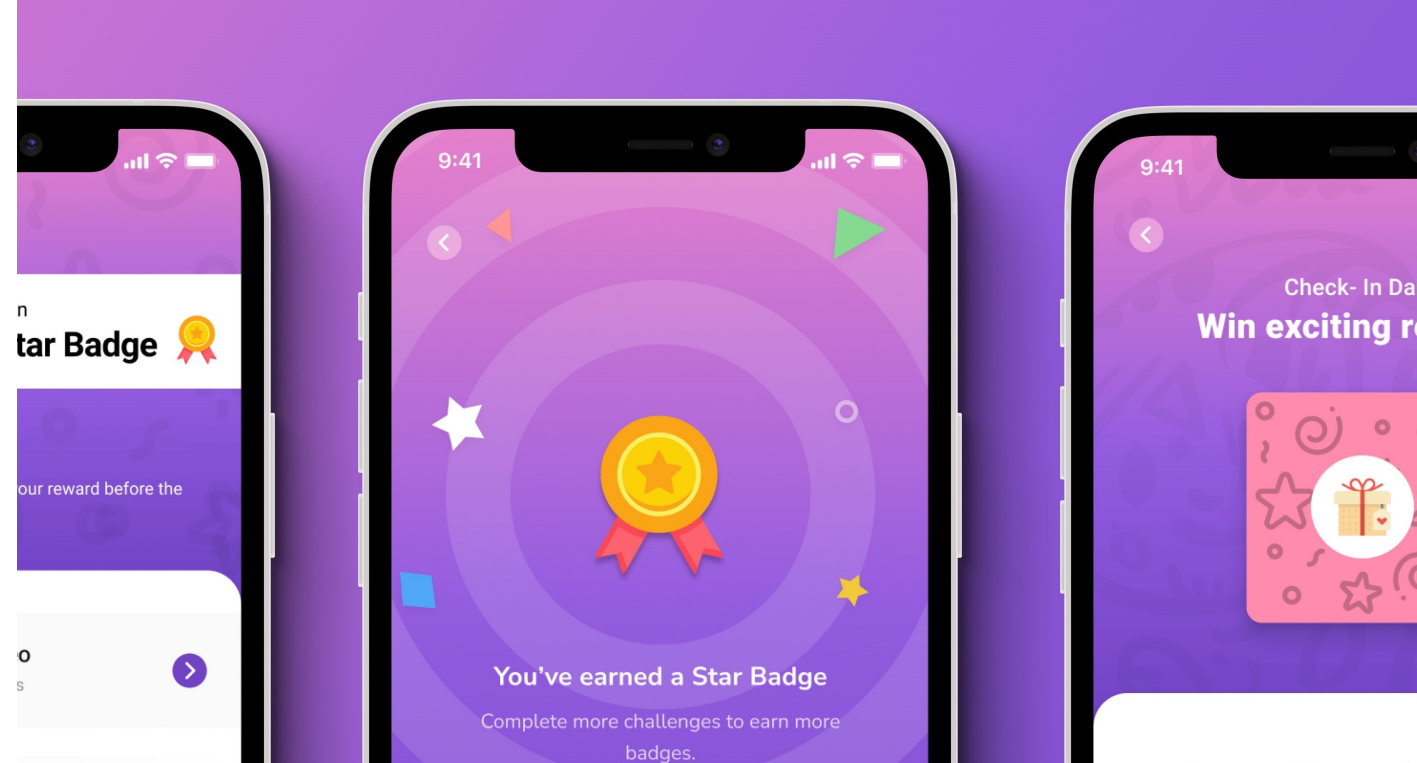
# Interactive content presentation

- Innovative human-computer interaction for learning systems



# Gamification in e-learning

- Gamification serves to make learning more engaging and interactive.
- Such systems incorporate game elements into educational content.



<https://www.customerglu.com/blogs/epic-meaning-engagement-learn-how-to-increase-customer-engagement-with-gamification>

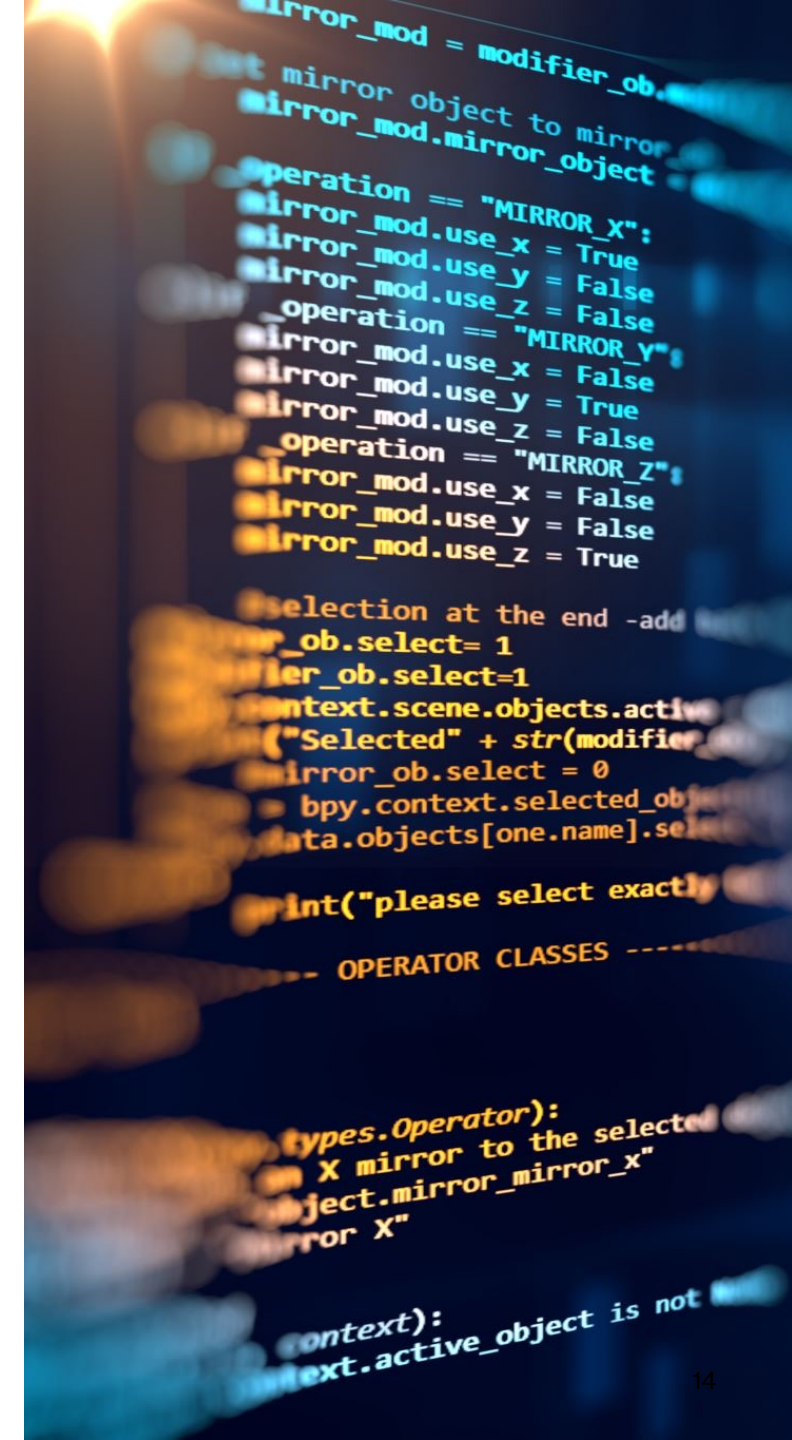
# Natural language processing tools

- NLP-powered systems are used for tasks like automated essay grading, chatbots, and virtual teaching assistants.
- They can assess and provide feedback on written and spoken language skill
- Learning content analysis



# Knowledge assessment

- Automated creation of knowledge assesment tools
- Quiz generation
- Automated check of text based assignemnts
- Automation of coding assignments



## Research focus:

# Systems to support self-regulated lifelong learning

Professional learning  
and growth

Professional  
relevance

Focus on applied  
knowledge

Identifying and  
consideration of the  
student's personal  
experience and  
knowledge

Constructivist-style  
of learning with self-  
regulation

Importance of  
interdisciplinary  
knowledge

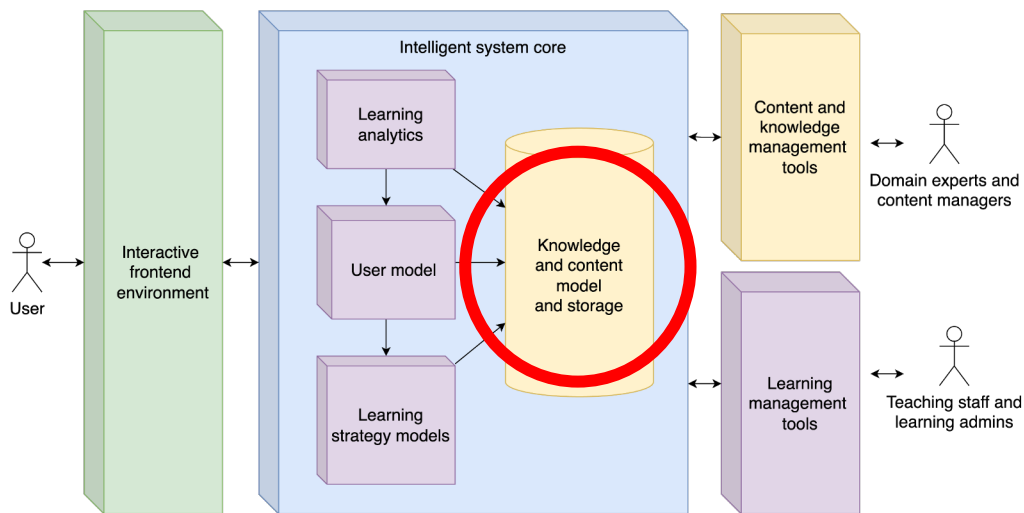
*O. Gagarin, S. Tytenko. Research and analysis of methods and models of lifelong learning intelligent systems, Research Bulletin of NTUU "Kyiv Polytechnic Institute", no 6 (56), pp. 37-48, 2007*



**ONTOLOGY-ORIENTED EDUCATIONAL  
WEB SYSTEM WITH USE OF  
INTERACTIVE CONCEPT MAPS**

---

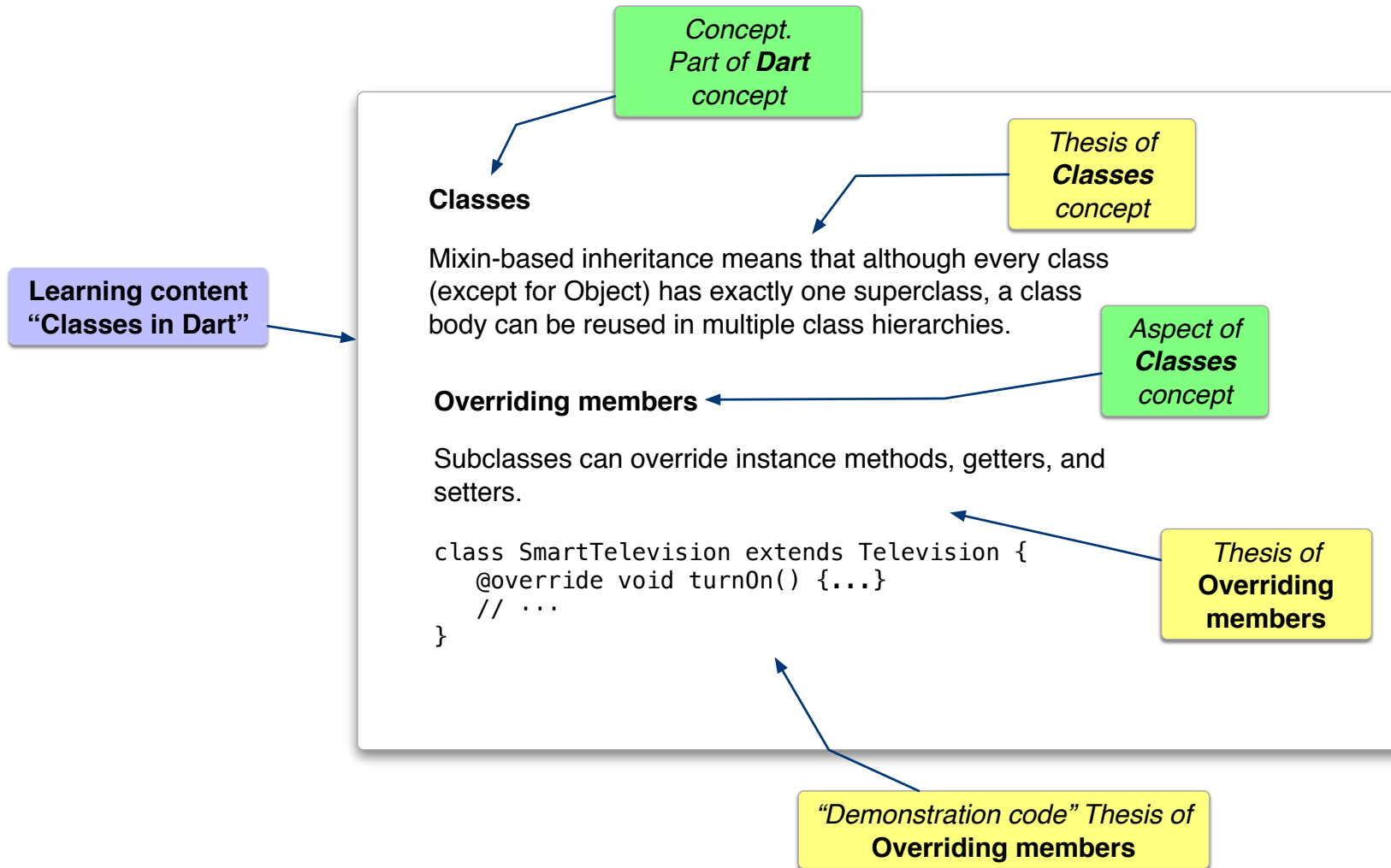
# Concept-thesis model of learning domain representation



Tytenko S. V CONSTRUCTION OF DIDACTIC ONTOLOGY BASED ON THE ANALYSIS OF CONCEPT-THESIS MODEL ELEMENTS. *Naukovi visti NTUU - KPI*. 2010, Vol. 2010 Issue 1, p82-87.

- It is proposed to implement semantic formalization of educational content by decomposing key elements of text and media into two basic structural units which are **concepts** and **theses**.
- *This approach involves the processing of educational content by an expert using a user-friendly web interface. As a result, we get a lot of concepts, as well as key information to them.*
- Concept-thesis data are used for:
  1. Automatic test construction and **knowledge assessment**,
  2. Construction of a **semantic summary**, and subject area **thesaurus**,
  3. Automatic construction of **didactic ontology**
  4. Interactive **concept maps**.

# Concept-thesis model of learning domain representation



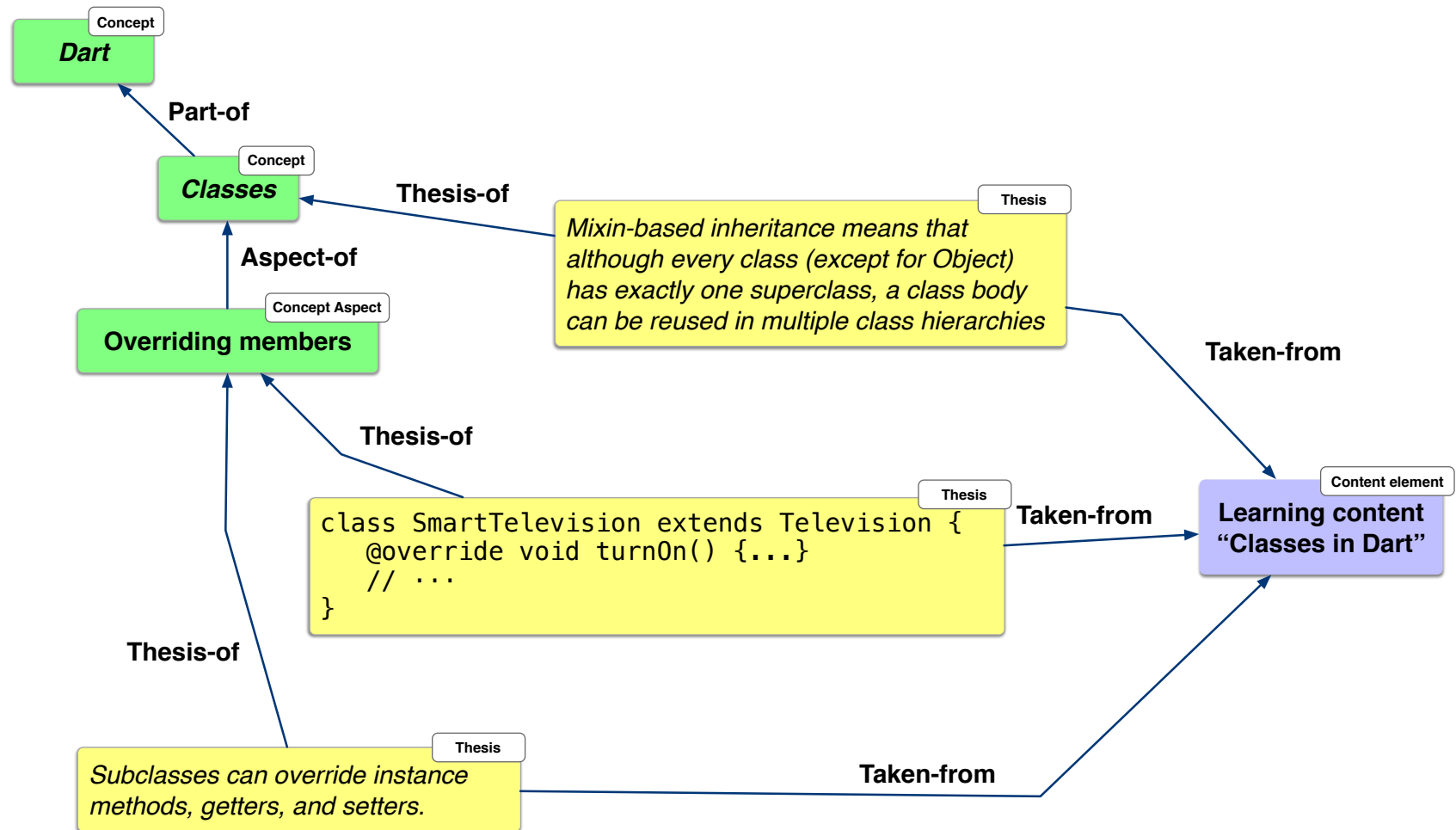
$$C = \{c_1, \dots, c_{n1}\}$$

$$T = \{t_1, \dots, t_{n2}\}$$

$$CT: T \rightarrow C,$$

$$TC: C \rightarrow 2^T$$

# Result fragment of ontology



# Concept relations: structural

## - Aspects

- $Aspects = \{c \mid c \in C \text{ isAspect}(c) = true\}$
- $AspectOf : Aspects \rightarrow C$

$$PartOf(c') = \left\{ c : c \in C \wedge \exists t' : \left( \begin{array}{l} t' \in TC(c') \wedge \\ TClass(t') = tRelPartOf \wedge \\ RelCT(t') = c \end{array} \right) \right\}$$

## - Part-of, is-a and instance-of relations

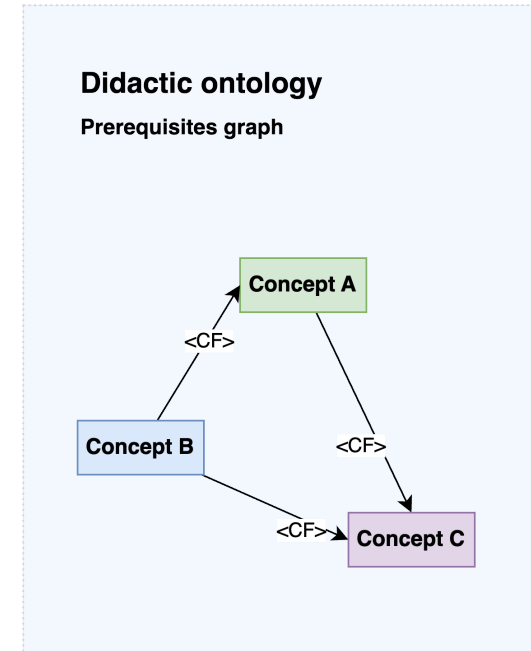
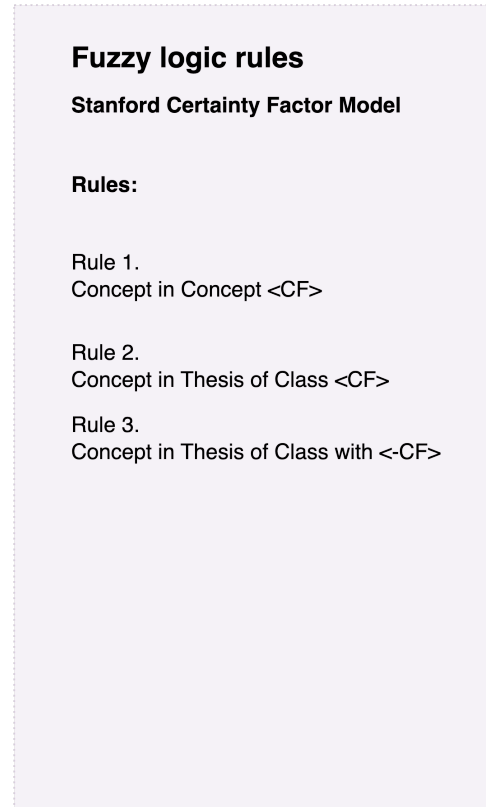
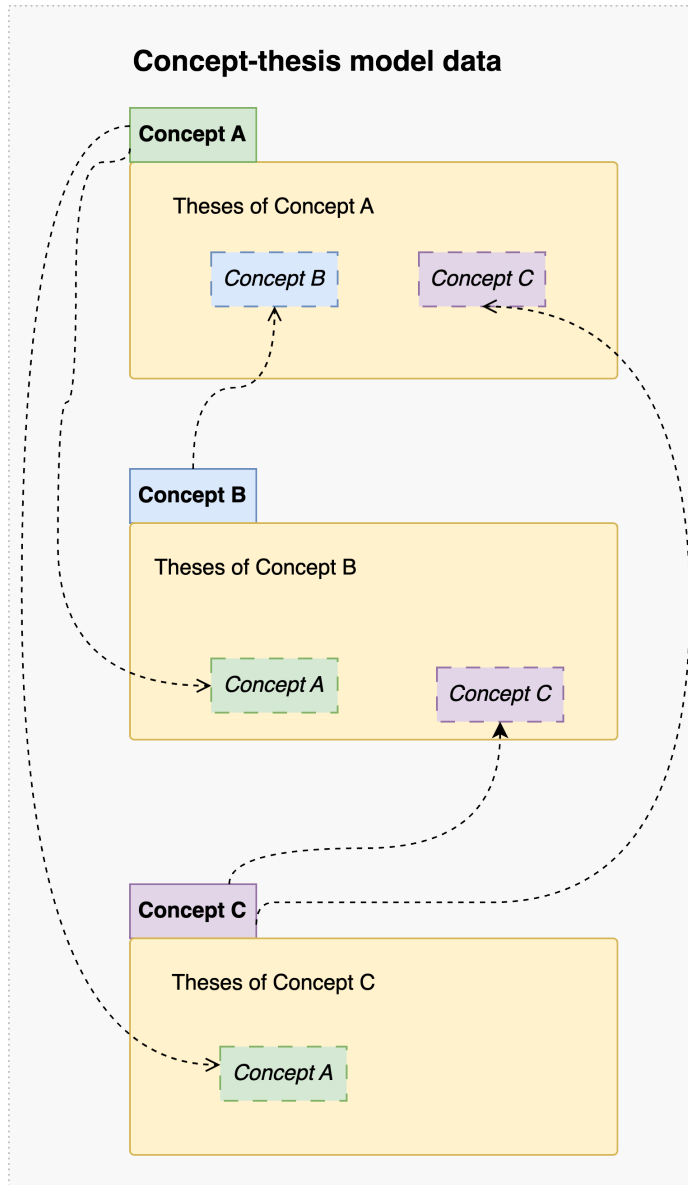
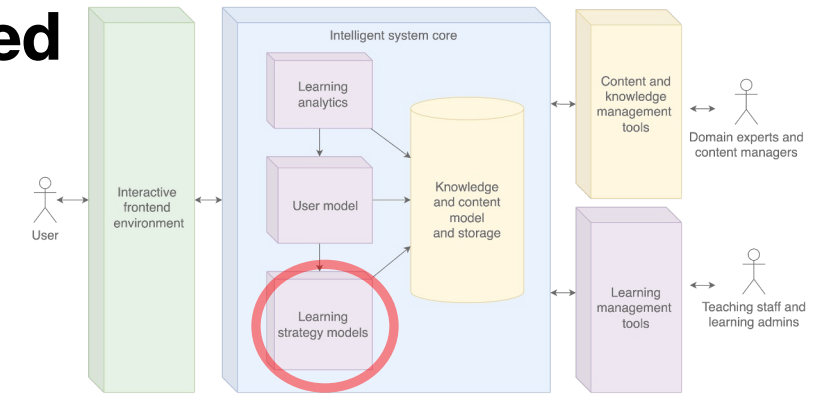
- $PartOf : C \rightarrow C,$
- $IsA : C \rightarrow C,$
- $InstanceOf : C \rightarrow C$

$$IsA(c') = \left\{ c : c \in C \wedge \exists t' : \left( \begin{array}{l} t' \in TC(c') \wedge \\ TClass(t') = tRelIsA \wedge \\ RelCT(t') = c \end{array} \right) \right\}$$

$$InstanceOf(c') = \left\{ c : c \in C \wedge \exists t' : \left( \begin{array}{l} t' \in TC(c') \wedge \\ TClass(t') = tRelInstanceOf \wedge \\ RelCT(t') = c \end{array} \right) \right\}$$

# Concept relations: didactic. Automatically generated

- Fuzzy logic based reasoning to define prerequisite concepts



Tytenko S. V CONSTRUCTION OF DIDACTIC ONTOLOGY BASED ON THE ANALYSIS OF CONCEPT-THESIS MODEL ELEMENTS. *Naukovi visti NTUU - KPI* . 2010, Vol. 2010 Issue 1, p82-87.

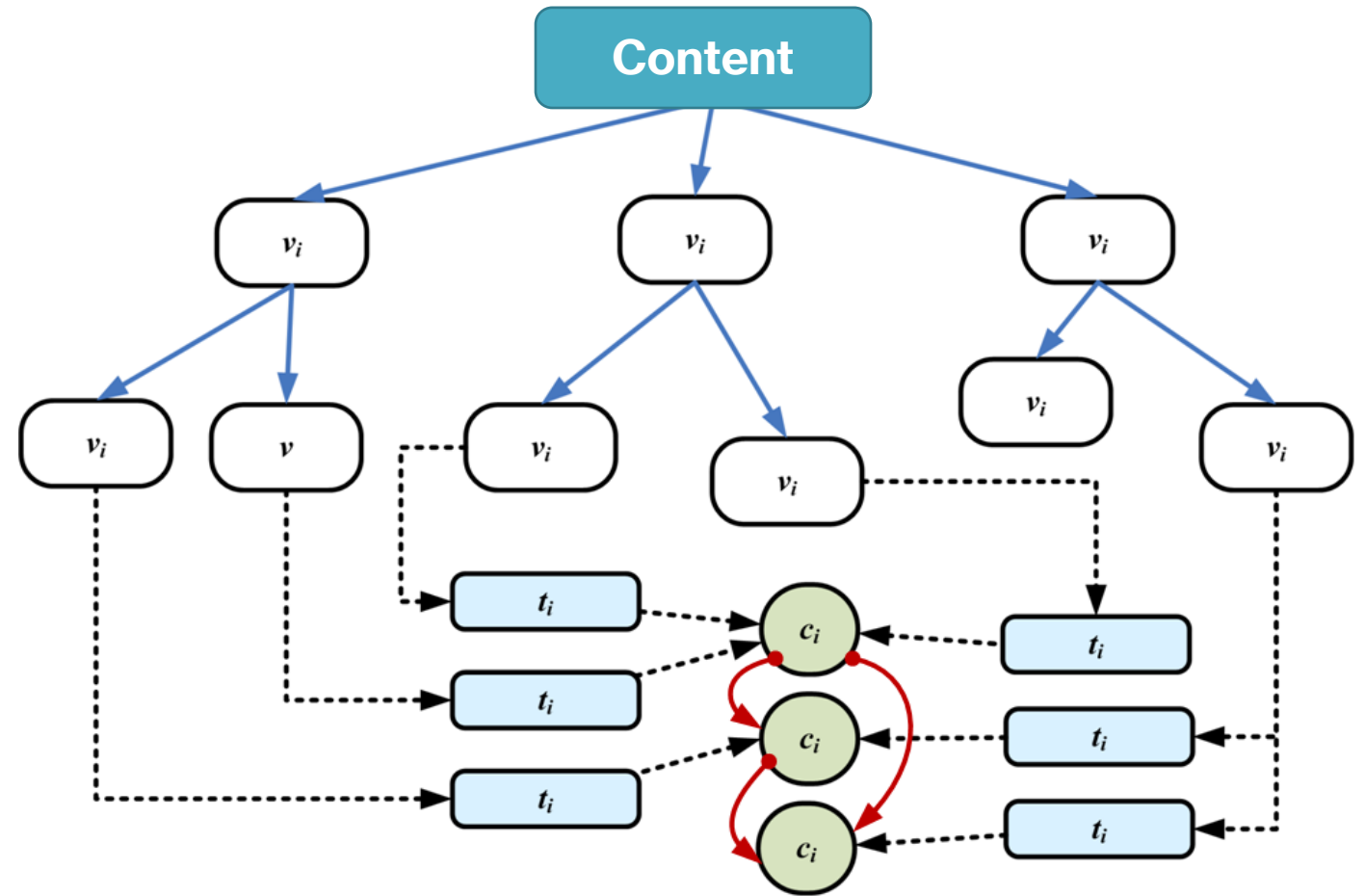
# CONTENT HIERARCHY AND CTM

$$V = \{v_1, \dots, v_{n_0}\}.$$

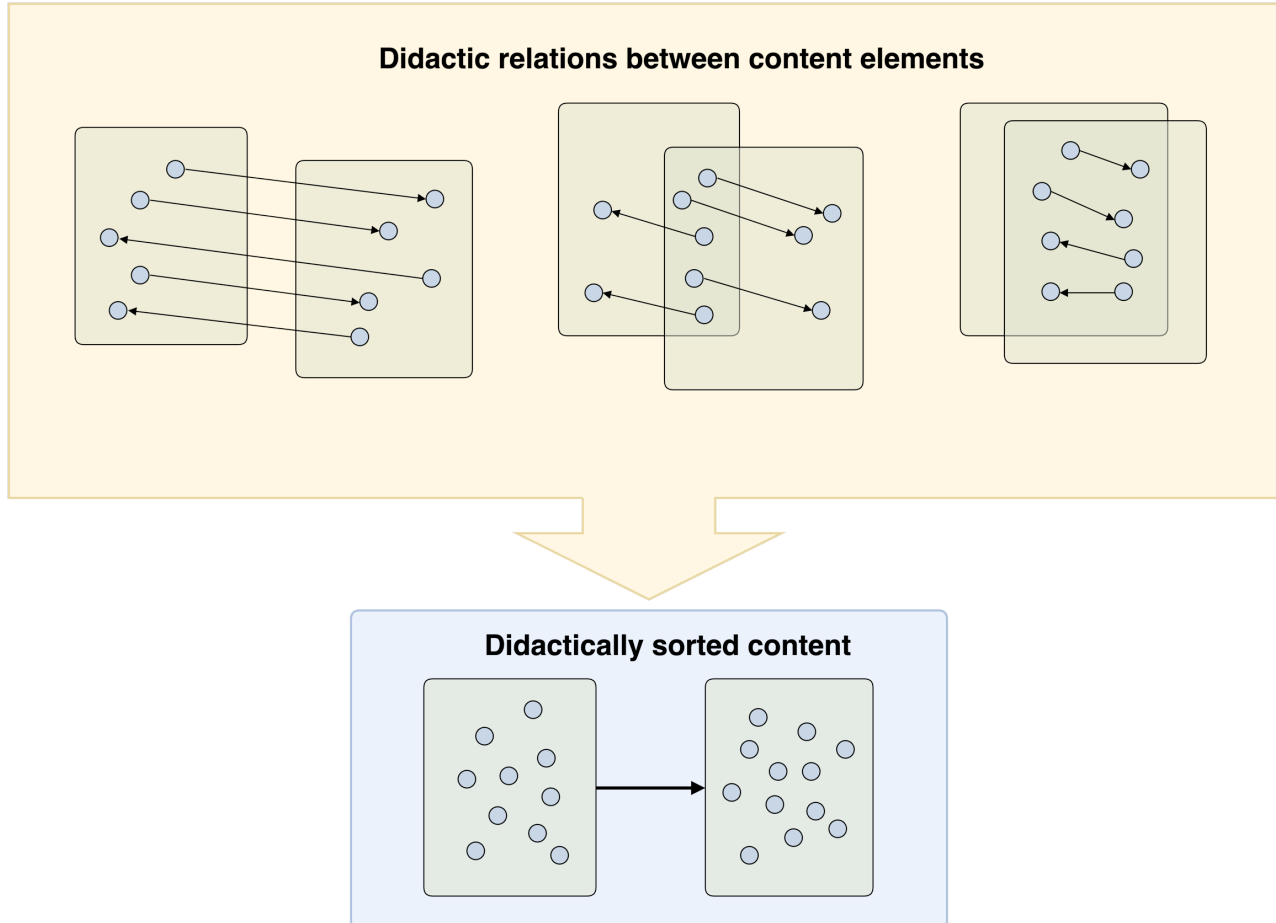
$$Ch : V \rightarrow 2^V$$

$$TV : V \rightarrow 2^T$$

$$CV(v) = \{c : TV(v) \cap TC(c) \neq \emptyset\}.$$



# Intelligent content ordering

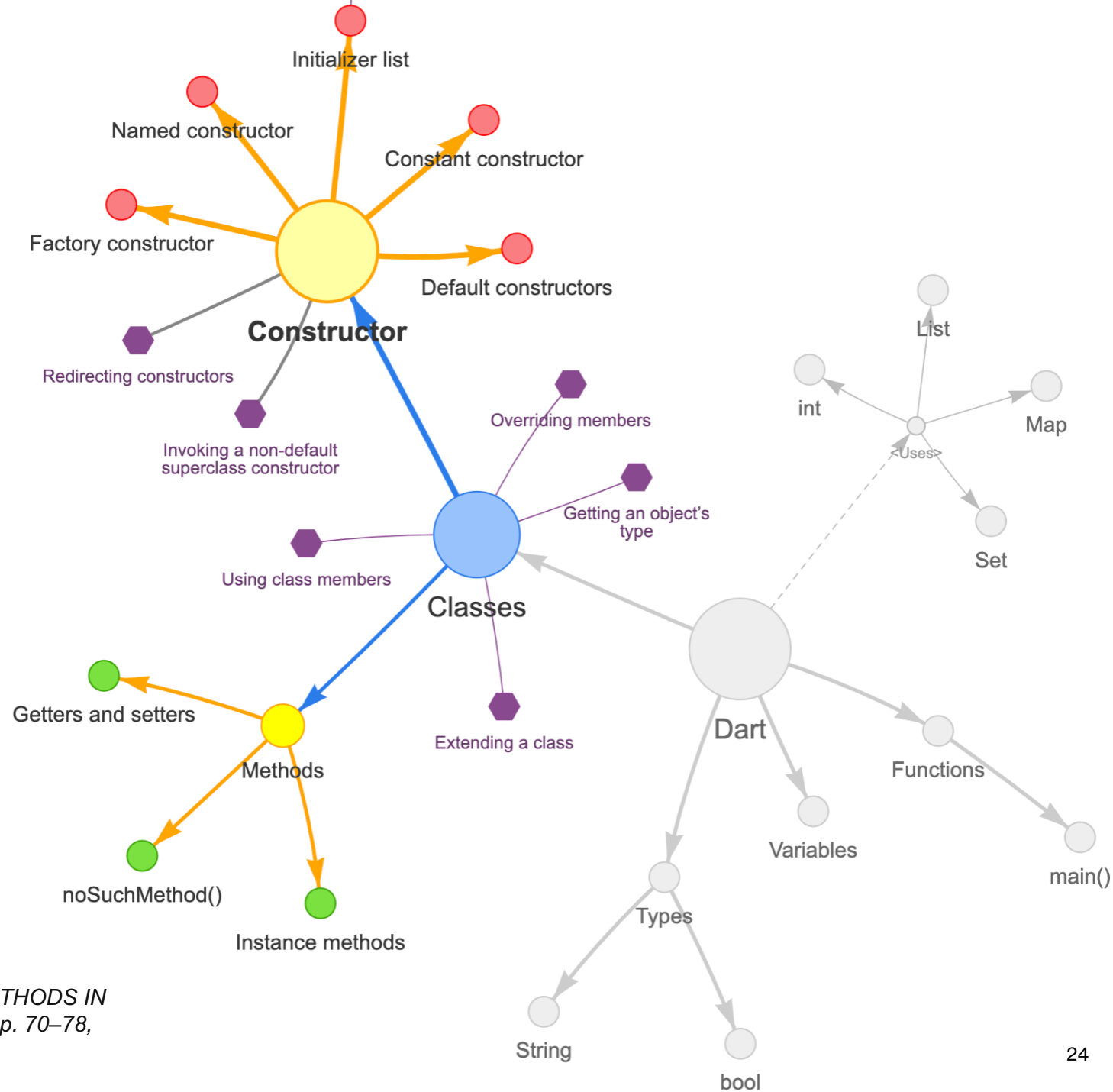


## Algorithm of automatic ordering of individual content

1. Indexing of content with concepts according to the subject area
2. Construction of a transitive closure of the ontology graph using a modification of the Floyd-Warshall algorithm
3. Rule-based analysis of the relationship between the concepts of content with Stanford fuzzy inference model.
4. Sorting content sections using the topological sorting algorithm.

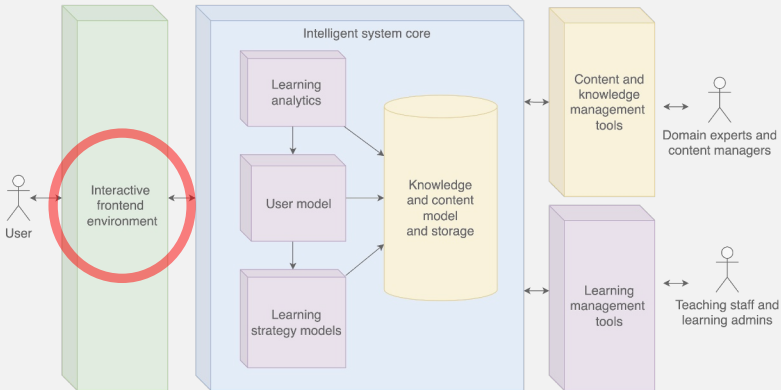
# Interactive concept maps

- Concept maps
- Knowledge graphs
- Ontologies
- Semantic networks
- Mind maps



# Interactive concept maps on Semantic portal

- [www.semantic-portal.net](http://www.semantic-portal.net)



**Chosen content filters  $V_f$**

**Filter map by subject**

Dart Tour | Type system | Fixing common type problems | Libraries | Effective Dart | Packaging | Tools and Techniques

**Tune filter with child subjects**

A basic Dart program | **Important concepts** | Keywords | **Variables** | **Built-in types** | **Functions** | Operators | Control flow statements | Exceptions | Classes | Generics

Libraries and visibility | Asynchrony support | Generators | Callable classes | Isolates | Typedefs | Metadata | Comments

Clickable & Draggable!

search node

← Dart

*Current concept*  
**theses**

**Variables**

Store references. ∞

Everything you can place in a variable is an object, and every object is an instance of a class. ∞

---

Even numbers, functions, and null are objects. ∞

All objects inherit from the Object class. ∞

Creating a variable and initializing it

Default value

const

**Reference to learning content**

**Child concept**

**Target concept**

**Aspect of concept**

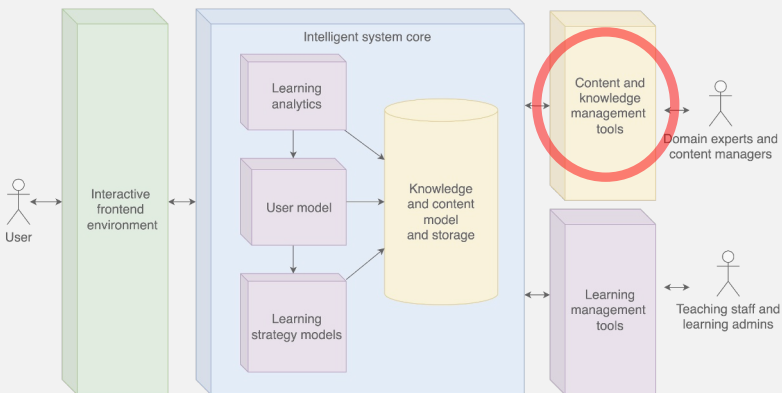
**Background concept**

**Interactive concept map**

**Information panel of concept map**

# Concept-thesis management

- Interactive tools to manage semantic entities of learning content



The screenshot shows a 'Semantic portal' interface. The main content area displays 'A basic Dart program' with a code snippet. Annotations point to various parts of the interface:

- Content**: Points to the main content area.
- Concept management panel**: Points to the right-hand sidebar.
- Added concepts**: Points to a list of concepts in the sidebar, including 'bool', 'List', 'main()', 'String', and 'Types'.
- Current concept to edit**: Points to the 'Dart' concept entry in the sidebar.

The code snippet shown is:

```
printInteger(int aNumber) // Define a function.
{
  print('The number is $aNumber.');// Print to console.
}

// This is where the app starts executing
main()
{
  var number = 42; // Declare and initialize a variable.
  printInteger(number); // Call a function.
}
```

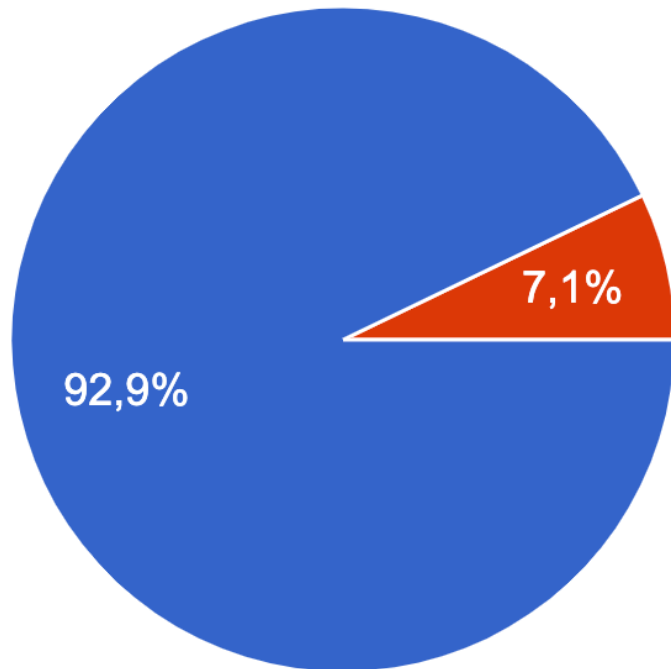
Below the code, it states: 'Here's what this program uses that applies to all (or almost all) Dart apps:'

- `// This is a comment` (A single-line comment. Dart also supports multi-line and document comments. For details, see Comments)
- `int A type` (Some of the other built-in types are `String`, `List`, and `bool`.)
- `42` (A number literal. Number literals are a kind of compile-time constant)

The sidebar on the right shows a list of concepts under 'Поняття сторінки' (8 items). Below this are buttons for '+ Нове поняття', '+ Новий аспект', and '+ Нове поняття-частина'. Under 'Поточне поняття', the 'Dart' concept is selected, showing its details in a dark blue panel.

# The results of a survey of fourth-year students in the study of Dart language

How useful was the concept map?

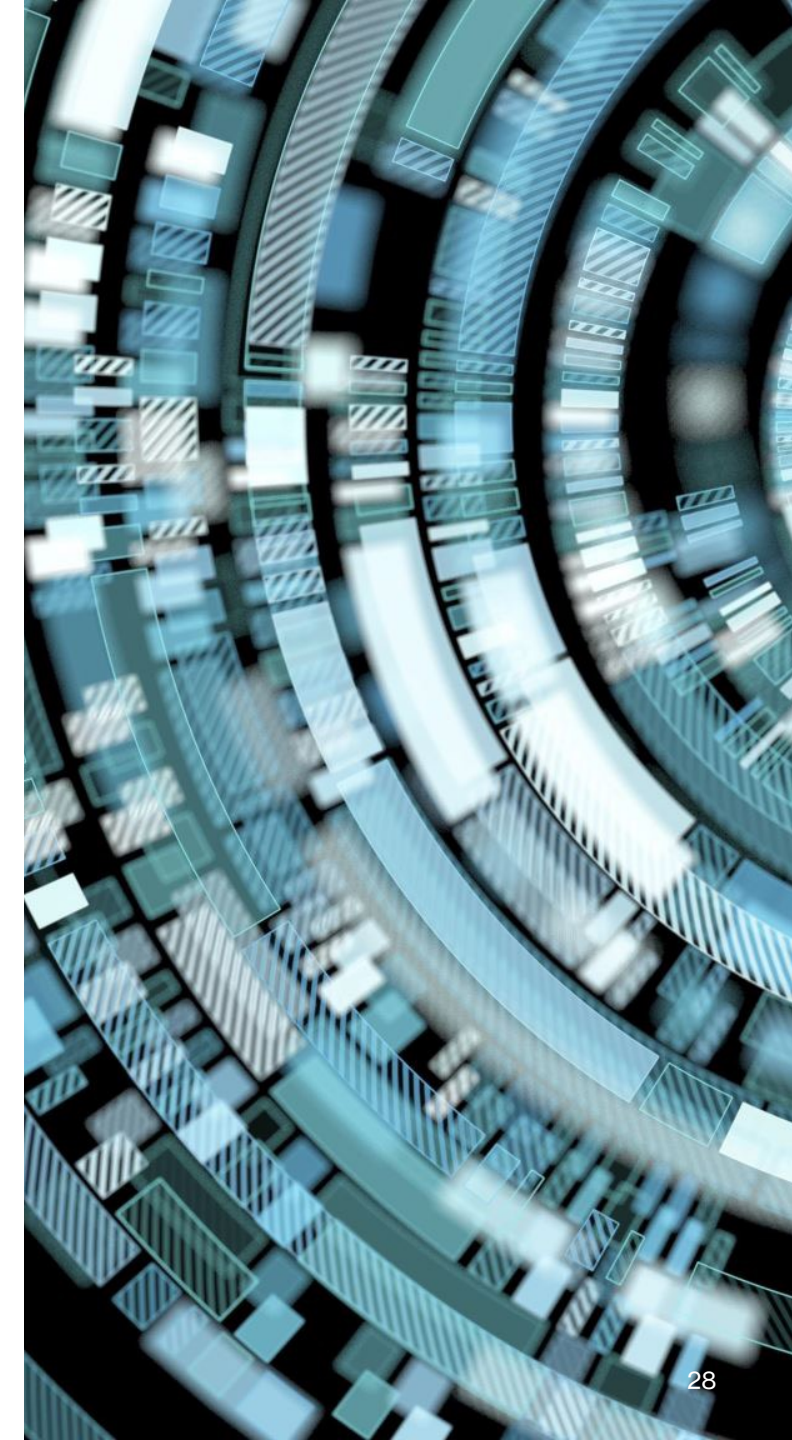


- Not useful, text format is enough — 0
- Useful, but text format is more convenient — 2
- Useful, is more convenient than text format — 26

# Further research

Large Language Models in Ontology-oriented educational web systems and Software Architecture:

- Automated extraction of concepts and theses from educational content with OpenAI API (*M. Rybak, AUK*)
- Generation of learning content with use of ontology-based course structure and OpenAI API (*A. Kakun, O. Ivchenko, AUK*)
- Exploratory learning UI based on interactive concept maps (*A. Tytenko, AUK*)
- Using smart contracts in the educational system based on the blockchain protocol (*I. Bukatych, AUK*)
- Software Architecture: Modular Monolith as a Microservices Precursor (*T. Shablii*), Teaching Software Architecture (*I. Manzhula*), Mobile Software Architecture Patterns (*D. Bilohub, M. Skrypchenko, AUK*).



Sergiy Tytenko

[sergiy.tytenko@auk.edu.ua](mailto:sergiy.tytenko@auk.edu.ua)

**INTELLIGENT  
EDUCATIONAL WEB  
SYSTEMS WITH  
INTERACTIVE  
CONCEPT MAPS**

