

Applying of artificial intelligence in the tasks of separation the university's IT landscape into functional modules for microservices and microkernels.

Piskunov P.I.¹, Prokhorov I.V.¹

(corresponding e-mail - pa.piskunov@gmail.com). ¹NRNU MEPhI, Russia, Moscow, Kashirskoe hwy, 31



Piskunov P.I.



Prokhorov I.V.



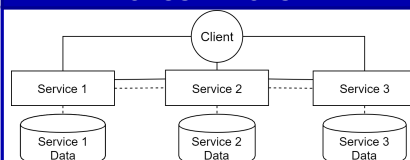
ABSTRACT #1

In our time, there are less and less "small" information systems. They more and more resemble complex information landscapes. Therefore, the question arises not of the creation of separate parts of the IS, but of their interaction and integration, especially in the case of performing heterogeneous tasks and working in a wide subject area.

Typically, this task is solved by an IS architect, or by several major developers using design methodologies.

But in our world there are many similar systems and there are general ideas for their design, and the number of little things that need to be taken into account is already difficult to fit in the head of one person. Why not apply AI in this case, which just has advantages in solving such unstructured problems.

MICROSERVICES #2



External interface (API, GUI)

Internal highspeed interface of data and command transfer

Fig 1. Microservice structure

Benefits:

- Independent service development and deployment
- Scalability
- Reducing design errors and their impact
- Relation of system structure with its logic

Difficulties

- Degeneration (with initially qualitative separation)
- Large number of connections and system elements
- The need to control the emergence of inseparable links between services and their new partitions

WHY AI #3

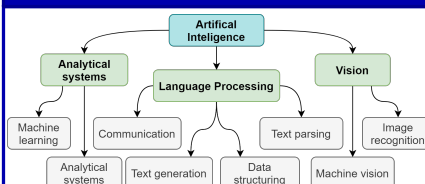


Fig 2. Some of AI subpartitions

Using AI in the separation and redesign tasks can be useful because:

- Program code is quite structured representation of the object
- Redesigning can be represented as clustering or classification
- The task of redesign can be variable and require some intuition
- Working with incomplete data
- Strongly correlated inputs
- Many types of input data
- Some memory may be required to optimize the process when updating the system

Brain Inspired #5

Different senses >> different input formats >> different programs or algorithms for parsing / loading data

Different parts of the brain >> differences in structure for the wide problem solving >> different approaches and analysis algorithms for different stages

Decision making >> an assessment of many factors >> assemblies of models and an assessment of their relevance to the task



Fig 3. Brain sections

Tab 1. The analogy of redesigning and ordinary thinking

	Stage		
	Data Collection	Analysis	Conclusion
Elements of the thinking and work of the human body	Vision (processing of image details), hearing (conversion of air vibrations into data), identification of pathogens	The formation of associations, analogies in memory, the formation of labels, grouping objects by their value	Analytical thinking, higher nervous activity, intuition (like unconscious thinking)
Existing Algorithms and possible methods	Converting architecture descriptions into machine-readable formats, code analysis, representing the system as a graph, etc.	Classification, clustering, decision trees, graph optimization, neural networks	DSS, fuzzy logic, self-organizing maps, memory neural networks
Generalized redesign task	Collection of requirements, system properties	Assessing individual modules, connections	Generation of conclusions, unambiguous recommendations and actions

UNIVERSITIES #4

Why might this be especially applicable to universities?

- A large number of departments and parallel activities
- Lots of input and output data in unstructured formats and from various sources
- A high degree of standardization of the educational process at the state level
- Long data life cycle and distinct milestones
- Often the lack of full-fledged permanent development teams
- The users do not have additional specialization in working with the information system
- Long-term support for system modules and irrelevant documentation

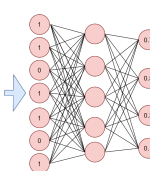
USED APPROACH #7

- **Space** – separation of individual pieces of programs by services at the level of code and functions
- **Object** – method \ backend function containing some scripts \ SQL queries
- **Technology**
 - **to collect properties** – manual structured documentation, nesting and relationships of tables in views database tools, searching for table names by code with SQL highlighting
 - **for analysis** – neural networks and cluster analysis algorithms

Some average sql request

```
select * from student stu
INNER JOIN group gr
ON gr.groid = stu.groid
INNER JOIN man m
ON stu.manid = m.manid
LEFT JOIN hostel_student h
ON stu.manid = h.manid
LEFT JOIN education_plan PLE
ON PLE.PLEID = GR.PLEID
```

Property	Value
Group	1
Student	1
Assessment sheet	0
Hostel	1
Man	1
Grades	0
Education plan	1



Suitability for report module
0.7
Suitability for student individual plan module
0.5
Suitability for diplom printing module
0.3
Suitability for stipend module
0.1

Fig 4. Separation preparation and design conveyor

PIPELINE #6

In order to create a pipeline of redesign, we first have to choose what we are working on, our data space, on the basis of which our system will do the redesign.

This is similar to how a person in childhood forms the concept of objects of the surrounding world. When solving specific problems, we can prompt him on what to pay attention to.

Tab 2. The analogy of redesigning and ordinary thinking

Stage	Question
Definition of target space	What and why do we want to split up?
Definition of separation objects and their properties	What elements will we work on?
Choice of technology, algorithm or AI type	What is best suited for analyzing the structure and objects we work with?
Data conversion and analysis	How do we present our objects so that our AI can understand?
Conclusions	What we can decide based on the results of the analysis?

CONCLUSIONS #8

- Good separation of the system into services can extend the life of the system and reduce the cost of maintaining it
- The modern level of development of AI makes it possible to shift many of the tasks of redesigning IS onto the shoulders of a computer
- The use of modularity, including modular AI, allows you to combine existing developments and increase the autonomy of the solution gradually, without a sudden increase in the complexity of one AI structure

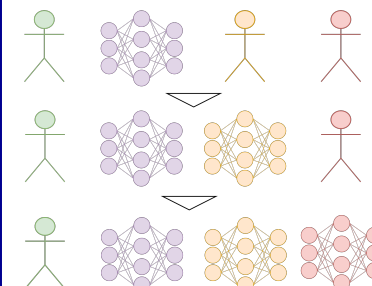


Fig 5. The future goal