Using a Hybrid Intelligent Information Systems Approach for Advertising Video Generation

Sergey Chernobrovkin, Igor Latkin, Marina Belyanova, Yuriy Gapanyuk
Bauman Moscow State Technical University
Moscow, Russia
gapyu@bmstu.ru

***Abstract*—In recent years, the effectiveness of traditional banner advertising on the Internet has been declining; users tend to pay less attention to the text and graphic banners. This leads to the fact that advertising agencies are forced to come up with sophisticated mechanics to attract users. The trend of recent years is the development of personalized advertising offers in social networks. In this article, we talk about the architecture of the system we developed for a unique advertising project, the purpose of which is to analyze available information from the user profile in the VK social network and generate a personalized video based on the received data. At almost all stages, we use machine learning methods to extract implicit profile information and filter data. The system architecture was created on the basis of the proven approach of Hybrid Intelligent Information Systems (HIIS).**

I. Introduction

The purpose of the developed system is to generate a custom video for a particular user using information from his or her social profile. This is done for advertisement purposes to encourage users to use a particular product. One of the key components is an analysis of the user's profile picture because it had to be used directly in a publicly available video, so a photo has been passed through a set of filters and transformers to be included in the resulting video. Also, an extensive profile analysis has been performed to determine the user's personality accurately.

Problems that have been solved in the developed system include face recognition, head pose detection, inappropriate content filtering, sentiment analysis, and object detection in photos.

Despite the fact that a large number of publications in the field of marketing are devoted to video advertising, we could not find publications that would discuss in detail the technical aspects of video advertising systems being developed. There are descriptions of approaches for generating video based on audio [1], [2], but this approach is not a direct analogue of the problem we are solving.

It should be noted that this system was created in real terms; therefore, instead of experimenting with datasets, libraries with pre-trained algorithms were used whenever possible. The system architecture was designed on the basis of the proven approach of Hybrid Intelligent Information Systems (HIIS) [3].

The developed system can be simultaneously considered as a HIIS and an information-analytical system [4].

II. The Developed System as HIIS

According to [3], a HIIS should combine the elements of the system based on soft computing and conventional system based on data and knowledge processing. The generalized HIIS architecture includes the following components:

- The environment.
- The subconsciousness module.
- The consciousness module.
- The boundary model of consciousness and subconsciousness.

The subconsciousness module (MS) is related to the environment in which a HIIS operates. Because the environment can be represented as a set of continuous signals, the data processing techniques of the MS are mostly based on neural networks, fuzzy logic, and combined neuro-fuzzy methods.

The consciousness module (MC) is based on conventional data and knowledge processing, which may be based on traditional programming, workflow, or rule-based approaches.

The boundary model of consciousness and subconsciousness is intended for deep integration of modules of consciousness and subconsciousness and represents an interface between these modules with the function of data storage. The data is a complex ontology that is used by both the consciousness and subconsciousness modules. The main task of the subconsciousness module is to recognize elements of ontology from the environment. If we consider the consciousness module as a kind of expert system, then the recognized elements of the ontology can be considered as elements of the operating memory of the expert system that trigger the corresponding rules. Depending on the goals of the system, rules can generate output information for the user or signals for the subconscious module that have the desired effect on the environment.

The proposed HIIS concept is considered as a generalized approach that should be adapted to create information systems in specific subject areas. The architecture of the proposed system based on the HIIS concept is presented in Fig. 1.

A feature of the environment in the developed system is that the environment is a heterogeneous source: texts, images, data from the social network. The current implementation of the system uses only the VKontakte social network; further, in the article, we use the abbreviation VK.

The subconsciousness of a system is a set of modules based on machine learning models and designed to extract

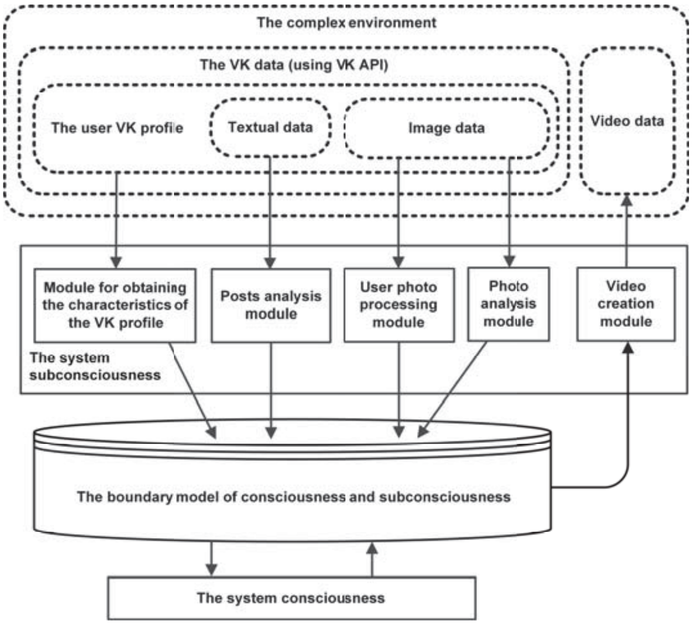

Fig. 1. The developed system as HIIS

the necessary data from the environment. Some of this data is used to verify user information, and some to form the final video.

The "User photo processing module" is used to search for a suitable user photo in VK and its subsequent processing for insertion into the video. The "Module for obtaining the characteristics of the VK profile" is designed to separate individual characteristics from the user profile using the VK API.

The "Posts analysis module" is designed to extract characteristics from the user's text messages using the VK API.

The "Photo analysis module" is designed to extract characteristics from user's photos using VK API.

The "Video creation module" is designed to generate the final video based on the previously received and analyzed data.

The boundary model of consciousness and subconsciousness is implemented as in-memory storage intended both for intermediate storage of output results of modules included in the system's subconsciousness and for storing processed data used for video generation.

The system consciousness module reads from the storage the output data of the modules included in the subconsciousness of the system, processes the data, and writes into the storage the processed data used to form the video. The principles of operation of the consciousness module are discussed in detail in the following section.

III. The System Consciousness as an Information-Analytical System

According to [5], the "Information-Analytical System" is "an automated system which carries out the storage, processing, analysis, and provision of information in a user-friendly form." The following formal definition is given in [4]: $IAS = \langle IAS_{DATA}, IAS_{PROC}, IAS_{INTR} \rangle$, where IAS – information-analytical system; IAS_{DATA} – the data and knowledge component of information-analytical system, responsible for the "storage of information"; IAS_{PROC} – the information processes component of information-analytical system, responsible for "processing and analysis of information"; IAS_{INTR} – the user interface component of information-analytical system, responsible for "input and provision of information in a user-friendly form."

For us, component IAS_{PROC} is especially crucial for organizing information processes and component IAS_{DATA} for aggregating data received from the subconsciousness of the system.

A. The Processes Description

According to [4], the IAS_{PROC} component is based on the metagraph process approach. The two main concepts of the metagraph model are metavertex and metaedge.

The metagraph metavertex: $mv_i = \langle \{atr_k\}, MG_f \rangle$, where mv_i – metagraph metavertex; atr_k – attribute, MG_f – metagraph fragment.

The metagraph metaedge: $me_i = \langle v_S, v_E, \{atr_k\}, MG_f \rangle$, where me_i – metagraph metaedge; v_S – source vertex (metavertex) of the metaedge; v_E – destination vertex (metavertex) of the metaedge; atr_k – attribute, MG_f – metagraph fragment.

In order to combine the data metagraph model and metagraph agent model, the concept of an "active metagraph" is proposed: $MG^{ACTIVE} = \langle MG^D, AG^{MG} \rangle$, $AG^{MG} = \{ag_i\}$, where MG^{ACTIVE} – an active metagraph; MG^D – data metagraph; AG^{MG} – set of metagraph agents ag_i, attached to the data metagraph.

The "metagraph process" may be considered as a metagraph metaedge based on active metagraph. An active metagraph is used instead of data metagraph as a process node: $PROC_i = \langle v_S, v_E, \{atr_k\}, MG_f[mv_{node} \equiv MG^{ACTIVE}_{node}] \rangle$.

The example of the metagraph processes description is represented in Fig. 2. The vertices and metavertices used for data descriptions are shown with circles. Active metagraphs $mg*$ corresponding to the metagraph process elements are shown with rectangles. The metagraph processes $PROC_1$ and $PROC_2$ are shown with double rectangles. The dashed link shows the call of nested metagraph process $PROC_2$ from the active metagraph element mg_{15}. The directed edges show the relationship between metagraph process elements. The undirected edges show the relationship between data elements or the relationship between data elements and active metagraphs. The input data for the mg_{11} element are shown as metavertex mv_1, which contains vertices v_{31}, v_{32} and connecting them edge e_{31}. The connection between the metavertex mv_1 and the process element mg_{11} is performed using the edge e_{41}. The attribute $mg_{11}.in = true$ means the semantic of input data. Similarly, using the edges e_{42}, e_{43}, e_{44}, the metavertices mv_2 and mv_3 are connected with active metagraph elements mg_{11} and mg_{12} as input-output data.

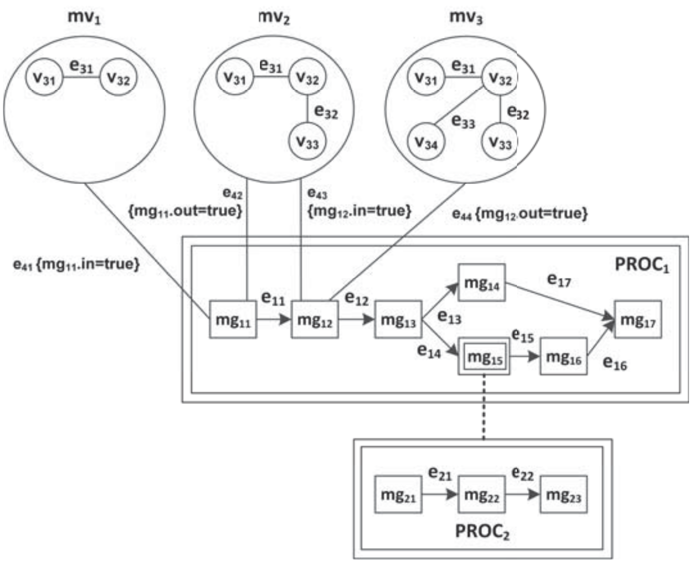

Fig. 2. The example of the metagraph processes description

The generalized process of the developed system operation is represented in Fig. 3. The process is shown as a sequence of steps that perform semantic enrichment of the resulting metavertex. The resulting metavertex is stored in the boundary model of consciousness and subconsciousness. To avoid cluttering the Fig. 3, the resulting metavertex data is shown as a simple outline instead of a complex graph.

1) The **Primary dialog** includes initial communication with the user and the choice of a VK profile for analysis.
2) After **calling the "Module for obtaining the characteristics of the VK profile,"** the profile characteristics are added to the resulting metavertex.
3) After **calling the "User photo processing module,"** the user photo processed and ready for including into the video is added to the resulting metavertex.
4) After **calling the "Posts analysis module,"** the extracted from text user psychological characteristics are added to the resulting metavertex.
5) After **calling the "Photo analysis module,"** the extracted from photos user psychological characteristics are added to the resulting metavertex.
6) The collected data is processed and aggregated for video creation purposes.
7) After **calling the "Video creation module,"** the generated video is sent to the user.

The data processing details are described in the following subsection.

B. The Data Processing Description

The classical multidimensional data model, proposed by Edgar F. Codd, allows working with numerical data (measures) binding them to the hierarchical taxonomies (dimensions) [6]. The multidimensional data model is a core for OLAP (online

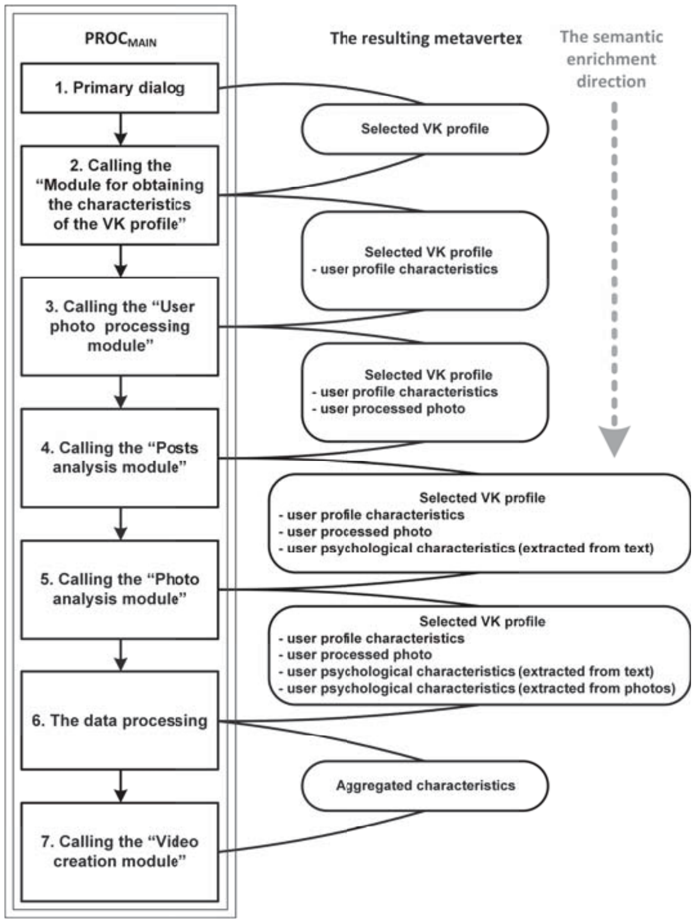

Fig. 3. The generalized process of system operation

analytical processing) information systems and is used in a significant number of information-analytical systems.

But the multidimensional model is oriented for numerical measures usage. Textual or object-oriented information is not considered for use as measures. This may be noted as a limitation of a classical multidimensional model.

An approach called "Graph OLAP" is currently being developed [7]. This idea is somewhat similar to ROLAP (Relational OLAP) approach, but instead of the relational model, the graph model is used. Thus, the "Graph OLAP" approach is an attempt to adapt the standard multidimensional model to graph data. This approach inherits the main problem of the classical multidimensional model. It does not allow changing the type and structure of the data in the aggregation process. To overcome this problem, it is proposed to use the metagraph approach [4].

According to [4] the multidimensional data model may be represented in the form of a metagraph. At the same time, this approach allows storing in hypercube facts and aggregate not only numerical values but any complex data structures. This allows working with data, knowledge, situations, processes descriptions represented in the form of metagraph.

The aggregation example is represented in Fig. 4. There

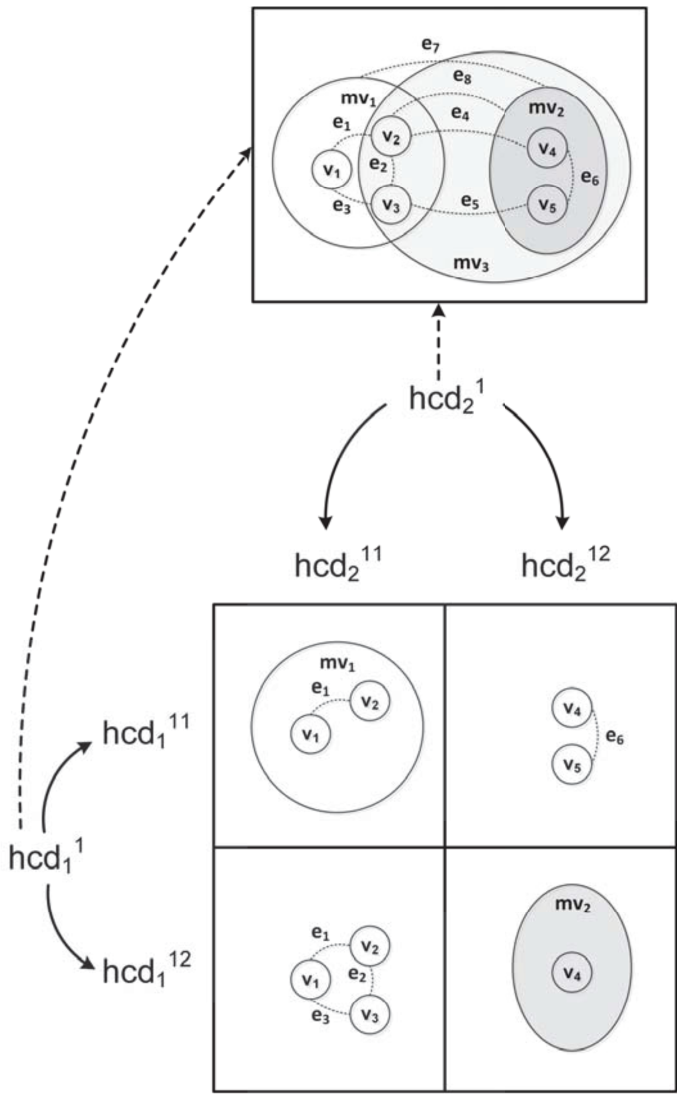

Fig. 4. The example of the metagraph aggregation

is a simple hypercube with two dimensions hcd_1 and hcd_2. The hypercube facts correspond to the hypercube dimension element combinations $hcd_{111} - hcd_{211}$, $hcd_{111} - hcd_{212}$, $hcd_{112} - hcd_{211}$, $hcd_{112} - hcd_{212}$ are lower-level hypercube facts. The combination $hcd_{11} - hcd_{21}$ corresponds to the aggregated hypercube fact. In the process of aggregation, not only quantitative characteristics change but also the metagraph structure of cells corresponding to the facts of the hypercube.

In our case, the input data for data processing is a semantically enriched resulting metavertex.

During data processing (block 6 in Fig. 3), we need to solve two problems: select an appropriate video template and get aggregated parameters for substituting into the template.

The Holland Codes model [8] is known as the Holland Occupational Themes (or RIASEC) model and is widely used in psychology. Codes are labeled as Realistic (Doers), Investigative (Thinkers), Artistic (Creators), Social (Helpers), Enterprising (Persuaders), and Conventional (Organizers) based on what. Example of Holland codes distribution for particular text input is provided in the Fig 5 [9]. They are officially used for students future occupation assessment and for staff recruitment. The target video templates correspond to the Holland codes.

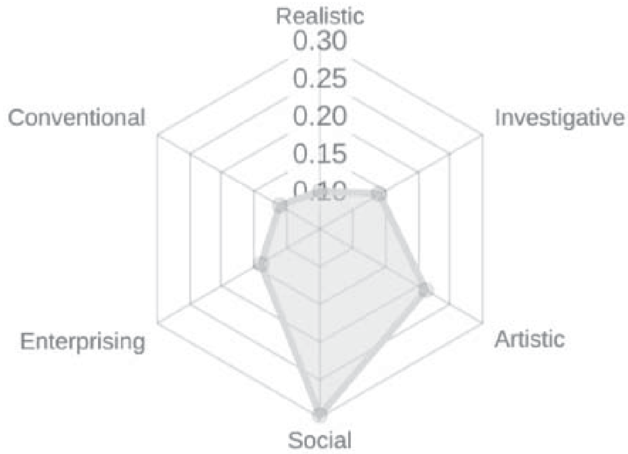

Fig. 5. The example of the Holland codes distribution

Data processing and aggregation are performed using rule-based metagraph agents. The result of processing is Holland's dominant code for a given user, as well as the aggregated data required to generate the video.

Holland's dominant code was determined based on the psychological characteristics of the user extracted from the texts and photos of the user's profile in VK (output of blocks 4 and 5 in Fig. 3).

In the next sections, we will look at the work of modules that are parts of the subconsciousness of the system.

IV. The Module for Obtaining the Characteristics of the VK Profile

The main purpose of this relatively simple module is to extract attributes from the user's profile using VK API. The extracted attributes are saved to the resulting metavertex.

The extracted attributes include processing of the following information: user's sex, city, country, education, spoken languages, subscriptions, various counters (such as audio and video files, friends).

V. The User Photo Processing Module

The process of analyzing the user's profile photo consisted of several stages, such as:

1) Fetch the user's profile photos.
2) Find exactly one face in the picture.
3) Determine the head pose.
4) Make sure that the photo does not have inappropriate content.
5) Prepare this photo for including in the video.

It was extremely important in this project that the human's head has been directed straight to the camera, i.e., a person has been looking straight, because the final video is dependant on this fact. For face recognition library [15] has been used. It determines if there is a face in the picture and if the person is one of the famous people, in which case such a photo is becoming not valid. There are three possible directions of head rotation: yaw, pitch, and roll like in Fig. 6.

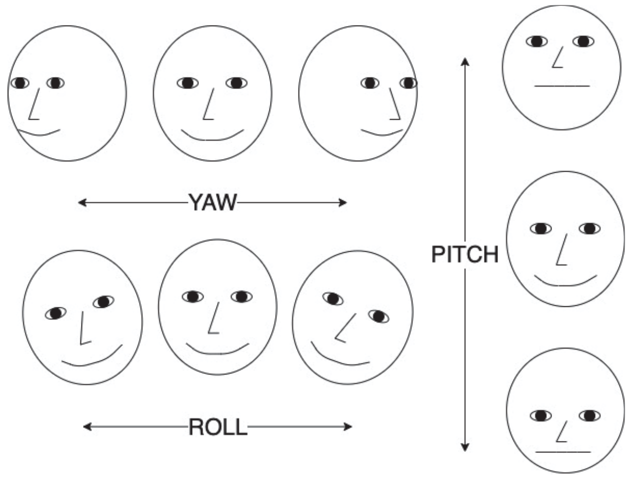

Fig. 6. Directions of head rotation

Big values of yaw and pitch are not valid for the current task, but roll rotations can be easily fixed by rotating an image in the opposite direction than the roll angle. To determine head pose, we used shape predictor from dlib [16] to find face landmarks coordinates then using solvePnP method from OpenCV, we determined the rotation matrix and the translation vector of the head in the picture. After that, using method described in [17], we transformed the resulting rotation matrix to two sets of Euler angles (which represent yaw, pitch, and roll).

If everything determined correctly on previous steps, then a photo is transferred to NSFW analysis using the AWS Rekognition platform. However, it has limitations. In our service, a face is cut out from a photo, but the service allows us to find inappropriate content on the entire photo. Therefore, artificially made, for example, in Photoshop, incorrect photos can still be allowed by the system.

VI. The Posts Analysis Module

The purpose of this module is to extract Holland's dominant code for a given user based on user's posts.

To solve this problem, we used the method described in [9]. The authors of this paper propose a neural network topology and a training dataset for determining the Holland codes from textual information. Texts of user's posts were provided as the input and the resulting output was the distribution of Holland codes.

The authors pose the problem of determining Holland's codes as a regression problem. The outputs of the neural network are the weights for each Holland's code for a given user.

To determine the dominant Holland code, it is sufficient to take the code with maximum weight. However, as the "photo analysis module" is used to correct weights, all the resulting weights for the "posts analysis module" are saved into the storage.

VII. The Photo Analysis Module

The purpose of this module is to determine the number of photos that contain different objects. This module is an experimental one and is used to correct the Holland code weights based on the subject of the photo.

An example of a correction rule: "If most of the user's photos show animals, multiply the weight of code 'Social' by the correction factor 1.2."

The COCO dataset [10] was used for module implementation. Currently, most object detection algorithms have models that are pre-trained on this dataset.

When implementing the module, we experimented with algorithms Mask R-CNN [11] (implementation [12]) and YOLOv3 [13] (implementation [14]).

In our case, the Mask R-CNN algorithm worked about three times longer than YOLOv3 algorithm. The quality of the algorithms was comparable. Therefore, the YOLOv3 algorithm was used to solve the problem.

The default values of the Holland codes are provided by the "posts analysis module." The dominant Holland code is determined after applying correction rules.

VIII. The Video Creation Module

The purpose of this module is to render the result video file personalized for a particular user. As an input, it accepts a user name, aggregated data, and a prepared profile picture. These data are passed to the render process, which renders an Adobe After Effects project by substituting corresponding layers and composition with proper texts and images, and the output file is uploaded back to the user.

In order to make the final video as personalized as possible, we were faced with a difficult task - to cut out the user's face from the most successful profile photo, adjust the position of the face, based on the previously obtained rotation data, and merge the resulting one with the frame on which the actor's photo is located. In the After Effects project we created a special layer for face, into which we transfer the photo, receiving the final frame with the user's face instead of the original actor's face as shown in Fig. 7.

Several methods have been tried to solve the problem. The first approach was to segment the face in the photo, cut out the segmented area, adjust the size, position, and color, and add the final frame to the layer. There are many ready-to-use solutions for solving the problem of determining a face in a photo, for example: OpenCV with a Haar cascade [18], dlib with segmenting a face with landmarks. For segmentation of objects on the photo - model FCN-8s-VGG [19].

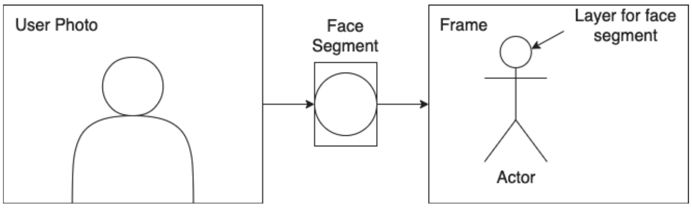

Fig. 7. The principle of layering photo on video

We focused on face segmentation, since it was important that hair or other objects in the background in the original photo did not fall into the frame. Using the obtained coordinates of the face, a mask was formed and the face was cut from the photo using this mask for further processing.

However, there were problems when processing the resulting segment of the photo. In particular, when adjusting the photo size for insertion or correcting skin color in the photo, the resulting frame did not always look natural. Sometimes it was seen that the face seemed to be layered on top of the frame, despite the normal proportions. Also, sometimes there was a problem of incorrect segmentation when part of the neck fell into the final frame.

Because of these problems, it became clear that cutting out a face from a photo and completely transferring it to the frame so that it looked realistic would not work. Therefore, we decided to apply the method of "mixing" the original face of the actor and the face of the user segmented in the photo. This method is detailed in [20]. The main idea is to segment faces in two photos, select anchor points, form 3d models of faces by points, and combine them. The disadvantage of such a scheme is that the user's original photo appears in the final frame with distortion and is "mixed" with the actor's original photo. We made this assumption, since it was necessary to obtain the most natural look of the frame with user's photo.

Therefore, in the final system, this particular method was applied, and the videos turned out to be realistic in more cases than in the first version.

The result is shown in Fig. 8

Fig. 8. The example of final video-frame

IX. Project Deployment

The entire system, including all of the components (except for the actual render services) were deployed in the Kubernetes cluster. Each service is a Python application packed inside a Docker container. Applications communicate with each other via a system bus implemented on top of the RabbitMQ server. The actual diagram of services is displayed in the Fig. 9. Every component displayed in a rectangle is a service deployed to the cluster. The idea is to be able to scale each service independently. So if, say, Sentiment SVC is becoming a bottleneck of the whole processing pipeline, we can easily scale it up and process requests faster.

Arrows on the diagram represent the data flow between services, and all these communications happen in the RabbitMQ server. The communication between service A to service B means that service A creates a task with destination set to service B and puts it into the system bus. Service B waits for messages directed to itself, executes it, and may or may not create another task for another service. Components:

1) ChatBot. User communicates with the entire system via chatbot in VK social network.
2) Photo Analyzer. Subsystem responsible for the extraction of an appropriate user's photo. This component consists of several subservices, which are actually independent.
   a) AvaFetcher. Service downloads all available profile picture photos from the user's page.
   b) FaceFind. Searches for one and only one face in the image. If an image does not contain faces or contains more than one face, it gets rejected.
   c) NSFWFind. Service detects any inappropriate content in the image.
   d) CelebFind. Service detects if the face in the image is the face of someone well known, so if this is the case image also gets rejected.
   e) FaceSegm. Service created a final image that would be inserted into the resulting video.
3) Profile Analyzer. Subsystem responsible for analyzing a user's profile and essentially choosing one of the AfterEffects projects to render.
   a) Grammar SVC. Analyzes user's posts to be grammatically correct.
   b) Sentiment SVC. Analyzes user's posts for their sentiment value.
   c) Animal SVC. Analyzes user's images to find any images with animals.
   d) Smile SVC. Analyzes user's images to find smiling persons.
4) Video Farm. Service responsible for communicating directly to After Effects and render the video file. This is the most resource-intensive point of the system, basically the overall speed of video creation is related to it. However, this subsystem can scale horizontally, so the parameters of the servers and their number were selected to meet the bandwidth requirements of the entire system.

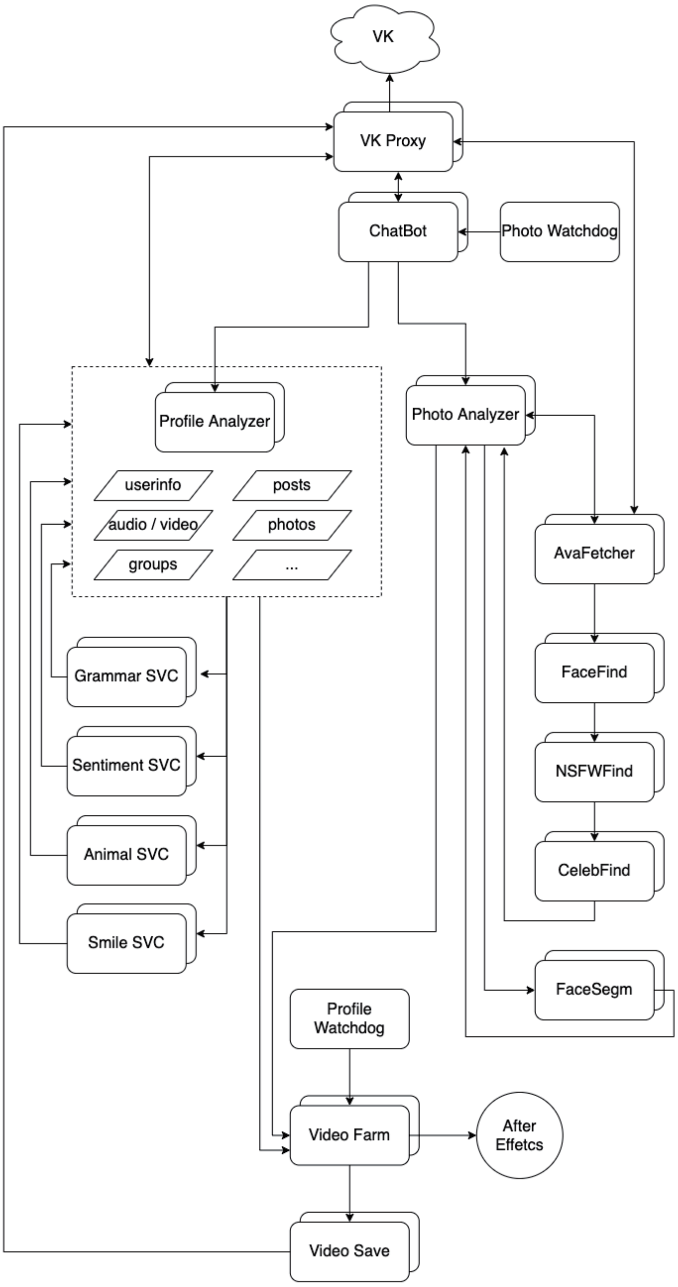

Fig. 9. System component diagram

5) Video Save. Service responsible for uploading the resulting video back to VK. Profile Watchdog and Photo Watchdog services were needed to track progress of each processing pipeline to watch if the pipeline is stuck on any stage (because of any kind of failure - network issues, application crash, etc.) and to restart the failed stage so the pipeline could eventually succeed.
6) VK Proxy controls requests rate limit to VK.

The only services that are not deployed in the Kubernetes cluster are the Video Farm and Video Save, because for the Video Farm service we need direct access to After Effects and this could be achieved on Windows servers.

Video creation is the most resource-intensive task, as shown in Fig. 10.

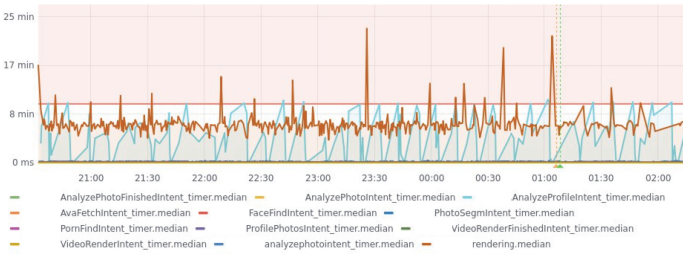

Fig. 10. System Services timings

We have stabilized the generation time by managing the resources and traffic supplied by advertising to the project. The user's waiting time is shown in Fig. 11 and meets the specified requirements.

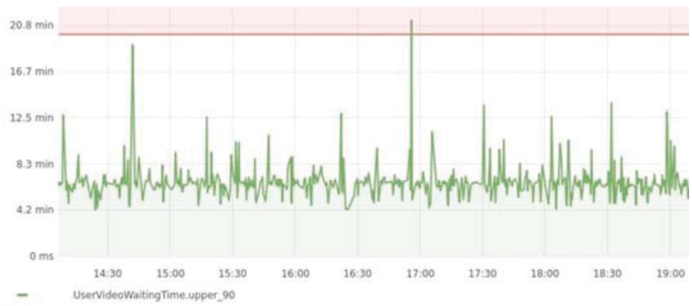

Fig. 11. User video waiting time

X. Conclusions

The developed practical system uses a combination of well-known machine learning methods, without opening new approaches to solving the listed problems. Nevertheless, the article focuses on the architecture of such a system and examines it from the point of view of the HIIS. The developed system can be simultaneously considered as a HIIS and an information-analytical system. Using the HIIS approach allows us to define the overall architecture of the system. Modules that use machine learning belong to the system's subconsciousness.

The information-analytical system consists of three components: the data and knowledge component, the information processes component, and the user interface component.

The metagraph process may be considered as a metagraph metaedge based on active metagraph. The generalized process of the developed system operation may be represented as a metagraph process. The process may be considered as a sequence of steps that perform semantic enrichment of the resulting metavertex.

The metagraph multidimensional data model allows storing in hypercube facts and aggregate not only numerical values but any complex data structures. This allows working with data and knowledge in the form of a metagraph. Data processing and aggregation are performed using rule-based metagraph agents. The result of processing is Holland's dominant code for a given user, as well as the aggregated data required to generate the video.

Thus, a proposed approach allows us to develop a personalized video advertising system based on social network user data.

REFERENCES

[1] Zhu, H., Li, Y., Zhu, F., Zheng, A., He, R.: Lets Play Music: Audio-driven Performance Video Generation. arXiv preprint arXiv:2011.02631 (2020)

[2] Guo, X., Li, J., Zhao, Y.: DanceIt: Music-inspired Dancing Video Synthesis. arXiv preprint arXiv:2009.08027 (2020)

[3] Chernenkiy, V., Gapanyuk, Yu., Terekhov, V., Revunkov, G., Kaganov, Y.: The hybrid intelligent information system approach as the basis for cognitive architecture. Procedia Computer Science Vol. 145, 143–152 (2018)

[4] Gapanyuk, Yu.: Metagraph Approach to the Information-Analytical Systems Development. In: Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, pp. 428–439. Moscow (2019)

[5] Buldakova, T.I., Suyatinov, S.I.: The Significance of Interdisciplinary Projects in Becoming a Research Engineer. In: Smirnova, E.V., Clark, R.P. (eds.) Handbook of Research on Engineering Education in a Global Context. pp. 243–253. Hershey, IGI Global (2019)

[6] Codd, E.F.: Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, E.F.Codd Associates (1993)

[7] Vaisman, A.A., Zimanyi, E.: Graph OLAP. In: Sakr, S., Zomaya, A.Y. (eds.) Encyclopedia of Big Data Technologies. Springer, Cham (2019)

[8] Nauta, M.: The development, evolution, and status of Holland's theory of vocational personalities: Reflections and future directions for counseling psychology. Journal of counseling psychology, vol. 57, no. 1, pp. 11–22 (2010). doi: 10.1037/a0018213

[9] Belyanova, M., Chernobrovkin, S., Latkin, I., Samarev, R.: Comparison of Convolutional Neural Networks and Search Based Approaches for Extracting Psychological Characteristics from Job Description. In: Lu, H., Tang, H., Wang, Z. (eds,) Advances in Neural Networks – ISNN 2019. LNCS, vol 11555, pp. 491–500. Springer, Cham (2019). 10.1007/978-3-030-22808-8_48

[10] The Common Objects in Context (COCO) dataset, https://cocodataset.org. Last accessed 21 July 2020

[11] He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. arXiv preprint arXiv:1703.06870 (2018)

[12] Project Mask_RCNN, https://github.com/matterport/Mask_RCNN/. Last accessed 21 July 2020

[13] Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767 (2018)

[14] Project YOLO-v3-Object-Detection, https://github.com/mohitwildbeast/YOLO-v3-Object-Detection. Last accessed 21 July 2020

[15] Project face_recognition, https://github.com/ageitgey/face_recognition. Last accessed 21 July 2020

[16] Dlib C++ library, http://dlib.net/. Last accessed 21 July 2020

[17] Slabaugh, G.: Computing Euler angles from a rotation matrix. https://www.gregslabaugh.net/publications/euler.pdf. Last accessed 21 July 2020

[18] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. last accessed 21 July 2020

[19] Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[20] Nirkin, Y., Masi, I., Tuan, A. T., Hassner, T., Medioni, G. (2018, May). On face segmentation, face swapping, and face perception. In 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018) (pp. 98-105). IEEE.