IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/45AI/2025/4531325/
Vol. 2025 (20), Article ID 4531325

Voice-Driven C2: A Multilingual Generative AI System for Real-Time Geospatial Interaction in Military Operations*

Paweł PIECZONKA and Marcin KUKIEŁKA

Faculty of Cybernetics, Military University of Technology,
00-908 Warsaw, Kaliskiego 2 Street, Poland

Correspondence should be addressed to: Paweł PIECZONKA, pawel.pieczonka@wat.edu.pl

* Presented at the 45th IBIMA International Conference, 25-26 June 2025, Cordoba, Spain

Abstract

This paper presents the concept of an innovative Command and Control (C2) system for exploring digital multilayer maps with voice commands, using generative artificial intelligence and modern data science methods. The goal of the system is to accelerate decision-making within the Observe-Orient-Decide-Act (OODA) loop by transforming naturally spoken commands into actions on digital maps. The proposed system allows users to query terrain and objects, calculate distances and plan optimal routes, and overlay dynamic operational layers (e.g., weather, logistics, intelligence) - all in real time using voice commands. The solution is multilingual, scalable and field-ready, and its architecture ensures integration with existing NATO C2 data infrastructures and processes. Special emphasis has been placed on interoperability, security and speed. The use of a voice interface is expected to increase operators' situational awareness, reduce cognitive load and significantly accelerate key decision-making in dynamic operational conditions.

Keywords: Voice-controlled C2 systems, Generative AI in defense, NLP, GEOINT, OODA loop optimization

Introduction

Today's military operations are characterized by an enormous volume of information and the need to interpret it instantly in order to make the right decisions. In a C2 environment, where data is constantly coming in and time is a critical factor, even small improvements to interfaces can affect mission success. Traditional operation of digital maps - via mouse, keyboard or touch - can sometimes be time-consuming and distract the operator. The OODA loop [1], describes the decision cycle (observe - orient - decide - act) and is commonly used to analyze how to improve decision-making in military operations. Accelerating the implementation of the OODA loop with the support of artificial intelligence has recently become the subject of intensive research and deployment in the armed forces. NATO is also emphasizing modernization of command and communications systems, integrating new technologies into existing C2 structures.

In recent years, there has been rapid development of voice interfaces in civilian applications - from assistants in smartphones, to infotainment systems in vehicles, to home smart speakers. In the context of geo-information, the first attempts to use voice for map navigation have emerged, indicating the potential for making GIS systems more accessible and intuitive to use. However, military requirements significantly outweigh civilian ones in terms of accuracy, security and operational context. Research shows that despite interest in the concept of voice control in GIS systems, existing solutions do not fully exploit the semantics of user commands or integrate with complex

Cite this Article as: Paweł PIECZONKA and Marcin KUKIEŁKA, Vol. 2025 (20) "Voice-Driven C2: A Multilingual Generative AI System for Real-Time Geospatial Interaction in Military Operations " Communications of International Proceedings, Vol. 2025 (20), Article ID 4531325,
https://doi.org/10.5171/2025.4531325

tasks. In other words - there is a gap between the capabilities of today's voice assistants and the needs of military geo-information systems.

In parallel, there have been tremendous advances in the field of artificial intelligence - especially in the area of generative models and natural language processing (NLP). Transformers have revolutionized the understanding and generation of language by machines, as exemplified by the BERT or GPT-4 models. These models can interpret the user's intentions and generate consistent natural language responses. Importantly, they are increasingly being used to interact with specialized data, such as natural language queries to databases or GIS systems. The introduction of large language models (LLMs) into geodata analysis allows for more intuitive retrieval of spatial information and automatic summarization of results, which can ease the burden on analysts and speed up the extraction of knowledge from map data. Still, the integration of generative AI with geospatial data in the realities of military operations remains a poorly explored domain.

In response to these challenges, the paper presents the concept of a voice-driven map system designed to support command in military operations. The system combines speech recognition, advanced NLP using LLM, GIS algorithms and natural language generation. This fusion of technologies is expected to enable commanders to interact with the map more naturally and quickly - asking questions by voice and receiving immediate results on the map and concise text reports. The paper goes on to describe the concept of the system and its key functionalities, the proposed architecture with technical details (including the implementation of the backend in Python and the frontend in React with OpenStreetMap maps), and discusses the use of large language models to understand user commands. The whole is complemented by aspects of multilingualism, interoperability with NATO infrastructure and scalability of the solution.

System concept

The main idea behind the proposed system is to create a voice interface that will allow operators to interact with dynamic, multi-layered geospatial data in real time. This means that a commander or analyst will be able to ask a question about the map or issue a command - using natural language - and the system will interpret the intention and perform the appropriate operations on the map. Thus, we are removing the barrier between user intent and digital action, speeding up the transition from Observe and Orient to Decide and Act in the OODA loop.

Key planned capabilities of the system include:

- **Voice search for objects and regions -** The user can ask for a specific object (e.g., “find the nearest airport”) or region (e.g., “show Alpha-3 sector”) using natural language. The system will recognize the speech and mark the indicated place or objects meeting the criteria on the map.
- **Real-time route planning and measurements -** Based on commands such as “Determine the best route from point X to Y avoiding hills,” the system will find the optimal route on the map and calculate the distance and travel time. This will allow quick analysis of route options without manually clicking multiple points.
- **Dynamic overlay of operational layers -** The system will allow you to turn on and off different data layers (e.g. weather forecast, own and enemy forces, logistic situation) by voice. For example, the command “Add weather layer” will superimpose current weather data on the map, making it easier to orient oneself in the field and plan operations.
- **Multilingual support -** By using multilingual models, the system will understand commands in different languages used by NATO allied forces. A French- or Polish-speaking operator will be able to communicate with the system in his or her language, and results will be consistent regardless of the input language.
- **Generate summaries and text reports -** Every interaction with the map can be automatically summarized. The system will generate short reports (e.g., “Route from point A to B with a length of 5.4 km, travel time of 10 minutes, avoids a blocked bridge”) or a synthesis of the situation in the indicated area. Such reports can serve as documentation of operations or quick reports.

The introduction of the above functionalities into the standard C2 process is intended to minimize manual steps and reduce cognitive "friction" - the operator does not have to switch attention between thinking about the problem and operating the tool, but communicates with the system in the most natural way, namely by voice. As a result, decisions can be made faster and with less stress on decision-makers. What's more, integrating such a voice interface into existing command procedures will make it possible to use it without disrupting existing workflows - the operator still has a traditional map and GUI, but gains an additional layer of voice interaction.

Proposed system architecture

The system architecture is designed to be modular, scalable and compliant with NATO interoperability standards. Figure 1 shows the main components of the system and the data flow between them. The following subsections discuss the role of each module and the technologies used.

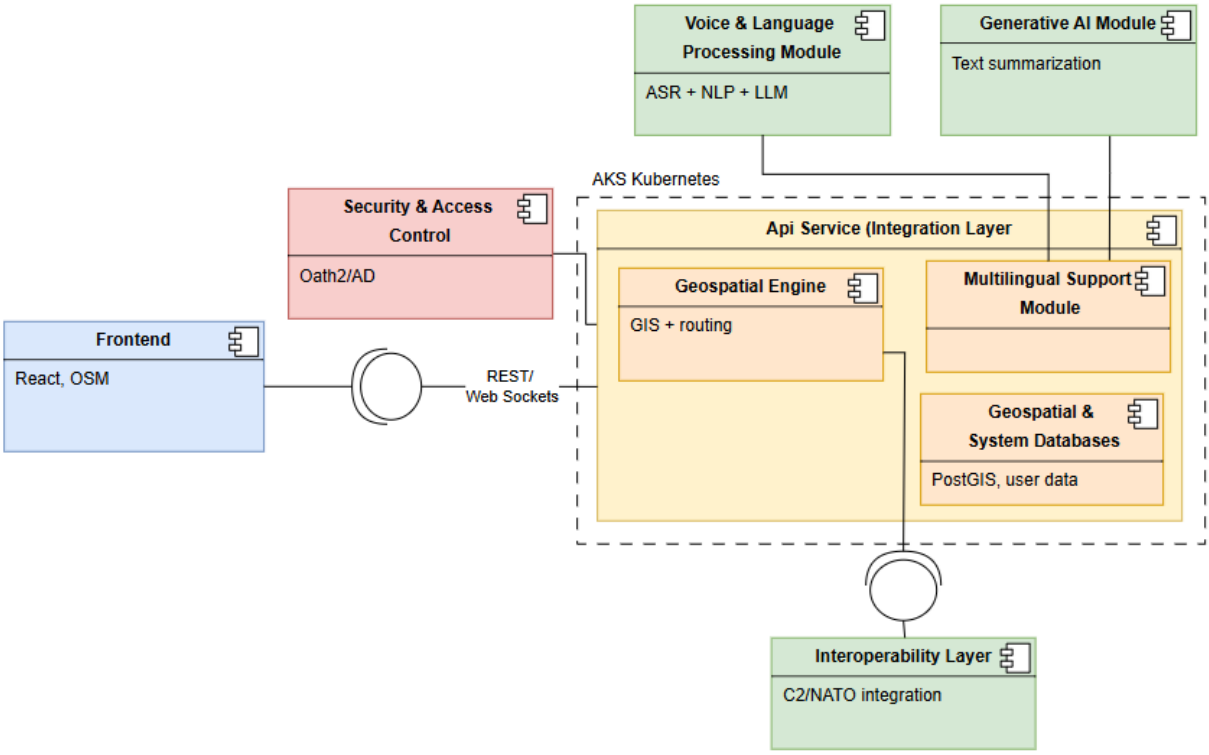

Figure 1. Component architecture of the system.

The spoken voice command is processed by a speech recognition module (ASR) and then by an NLP layer based on a large language model to interpret the intent. The GIS geo-information engine performs the desired action on the map (e.g., routing, searching for an object) and returns the result, which can be summarized as a text report by the generative AI module. The system integrates with external data sources (e.g., intelligence or weather data) and maintains interoperability with C2 infrastructure through a dedicated data exchange layer. Multilingual support is provided through multilingual NLP models, supporting speech recognition and command understanding regardless of the user's language. The data exchange mechanism based on web sockets presented in Figure 1 allows for real-time data visualization on the frontend without the need to refresh the application.

A. Voice interface (ASR)

The first part of the processing chain is the voice interface, responsible for capturing a spoken command and converting it into text. Modern Automatic Speech Recognition (ASR) models such as OpenAI Whisper and

Mozilla DeepSpeech were used here. The Whisper model has been trained on a huge dataset (680,000 hours of speech recordings in multiple languages) and achieves high resistance to noise and recognition errors. As a result, the interface is able to operate in field conditions, in the presence of audio interference, and still transcribe commands correctly. An important aspect is the fine-tuning of the ASR model to military specifics - taking into account specialized vocabulary, code names and acronyms used in the armed forces. By fine-tuning the model on such a corpus or using domain-specific vocabularies, the system will correctly recognize, for example, a command containing unit cryptonyms or mission-specific landmark names. The text string (speech transcription) obtained from ASR is then passed to the NLP layer [2]. It is worth noting that the ASR module can operate locally (at the edge of the network) - on a field device or mobile server - which ensures operation even with limited access to the cloud and increases security (we do not transmit raw audio over the network).

B. Natural language processing layer (NLP)

The recognized text command goes to the Natural Language Processing module, whose task is to understand the user's intent, interpret the context and extract key parameters of the command (e.g., place names, units of measurement, types of objects). Here we make use of transformative language models, in particular large models like GPT-4 and specialized BERT-type models adapted to information extraction tasks [3]. Thanks to their powerful context-understanding capabilities, these models are able to distinguish the nuances of geographic queries - for example, to recognize whether the word “Jaguar” in a given utterance means the name of a locality, the codename of an operation or the type of vehicle. The LLM is used here for syntactic-semantic analysis of the command and can operate in a few-shot mode (using appropriately designed prompts/prompts) or through specialized training toward map-based interactions. For example, the system can use prompt engineering, providing the GPT-4 model with context: “The user gives map-related commands. Your task is to formulate computer-understandable GIS instructions based on the query.” Based on this, the model will generate, for example, a GIS query structure (pseudo command language) or simply a well-parameterized query to the geo-information engine. The NLP layer also takes care to take into account the context of the dialogue - if the user asks another question referring to the previous one (e.g., “is this bridge passable?” after previously asking about bridges on the river), the model will use the interaction history to correctly interpret the pronoun “this”. The implementation of this can be done by maintaining the state of the conversation and providing the model with, for example, the last N commands along with the results in the form of a history.

Worth noting is the use of a large language model as the centerpiece of this layer. The LLM not only classifies intentions (e.g., search for an object vs. change map view), but can immediately generate a response in text form - which is used further down the line (generative module). In this way, one model can perform two functions: understanding commands and generating summaries, which simplifies the architecture and ensures language consistency of the system. The prototype implementation uses the GPT-4 model available through the OpenAI API to interpret commands with high efficiency, but consideration is being given to implementing an open-source model trained on a similar task for independence from external services.

C. Geoinformation engine (GIS)

At the heart of the system is a geo-information engine that translates an interpreted command into an operation on map data. This engine integrates with a geographic information system (GIS) - this could be an existing platform like ArcGIS Enterprise or an open source QGIS/PostGIS stack, or it could directly use data from OpenStreetMap (OSM). The implementation layer assumes the use of Python APIs of these tools (e.g. ArcPy for ArcGIS or equivalents in QGIS) or dedicated geoengineering libraries (such as GeoPandas, Shapely for geometric operations, NetworkX/OSRM for routing, rasterio for raster analysis, etc.). The engine's task is to perform a map action according to the user's intention: it may be to zoom the map view to the indicated region, search for objects that meet the criteria (e.g., all bridges on a given river), add a new data layer to the view, take a distance or area measurement, generate a terrain profile along the route, etc. [4]

The GIS engine has access to spatial databases and web services. When using OSM as the basis for topographic data, the system can use OSM APIs (e.g., Nominatim for geocoding place names and tile servers for map underlay) and tools such as OSRM (Open Source Routing Machine) to calculate routes along OSM roads [5]. This allows

commands like “calculate route” to be executed instantly on current road data. In addition, the engine can integrate data from sensors and military sources - for example, current troop positions from Blue Force Tracking, threat intelligence information or weather forecasts. This data is treated as additional layers and can be stored in an internal GIS database (e.g., PostGIS) or retrieved through an interoperability layer (described further below).

An important element is the business logic in the GIS engine, which is responsible for spatial inference. For example, for the query “Which areas are impassable after recent rainfall?” the system can use data on rainfall intensity and overlay it on a landform map to indicate valleys at risk of flooding. Such complex queries require a combination of multiple data layers and decision rules - the GIS engine accomplishes this through a sequence of geoprocessing operations. The advantage of using existing GIS platforms is the ability to use off-the-shelf analysis tools (e.g., terrain analysis, buffering, layer intersections) and optimize them.

The result of the geo-information engine is passed back to the presentation component - typically this will be: (a) an updated map view (e.g., drawn route, marked objects, attached layer), (b) possibly numerical or descriptive data that constitute the response (e.g., route length, list of found objects with attributes). This data will then be used by the generative module to generate a feedback message to the user.

D. Integration of generative AI (summary module)

The generative component is responsible for automatically creating summaries, messages and reports based on user interaction with the system and the results of spatial queries. It uses the capabilities of a large language model (the same one used in the NLP layer or a separate specialized one) to generate a concise textual description. For example, if an operator asks for a terrain analysis, the system might return a message along the lines of: “The area around point Alpha is mountainous terrain - average elevation 520 meters above sea level, no paved roads within 2 km. Traffic conditions: difficult.” This type of automatically generated information allows you to quickly get an idea of the situation without the user interpreting the raw data himself [6].

The generative module can perform several roles:

- **Summary of query results -** for example, after searching for bridges along the patrol route, the system will summarize: “3 bridges were found, all with a load capacity of more than 30 tons, two of them are controlled by own forces.”
- **Generate alerts/reports -** if the analysis reveals a situation that requires attention, the system can generate a message-style alert to the commander. E.g., after a weather layer has been applied: “Caution: heavy precipitation forecast for Bravo area within 2 hours may reduce visibility and maneuverability.”
- **Documentation creation -** upon request, the system can prepare a full report of the session (which areas were reviewed, what decisions were made), which is useful for debriefing and post-operational analysis.

On the technical side, language generation is implemented by the same NLP model (GPT-4) [6] with appropriate instructions (e.g., “Create a report for the commander based on the following map data...”) [7]. This takes advantage of the model's contextual knowledge of the world and its ability to formulate text that resembles the style of military reports. Alternatively, a slightly smaller language model specialized in generating specific formats of statements (such as situation reports) can be used. Generative AI allows personalization of the response style - depending on the rank of the recipient or the type of operation, the report can have a different tone and detail.

What's important is that the user remains in the loop - it's up to the operator to use the generated summary or to analyze the data on the map himself. The system does not impose conclusions, but only suggests them, which is important from the point of view of trust in AI in a military environment. Any message generated can also be backed up with source data (for example, the system can show which data was used to generate a flood warning). Such transparency increases the reliability of the generative module.

E. Multilingual processing module

To make the system usable in a multinational environment, a multilingualism module is provided that integrates models for translating and understanding multiple languages. In practice, this means that both the ASR and NLP stages use multilingually trained models (e.g. Whisper supports multiple languages at once, and the NLP layer

uses models like XLM-R or mBART that understand texts in different languages). The use of the XLM-R model (XLM-RoBERTa) makes it possible to achieve a high level of command comprehension even for languages less represented in the training data. This allows operators from different countries to work together, issuing commands each in their own language, with the system interpreting all of them correctly.

Technically, the module takes care of, for example, recognizing the language of utterance (which Whisper can do automatically [4]) and redirecting to the appropriate NLP model (if separate models for different languages were used). It is also possible to use an intermediate translation layer - for example, translating statements into English and operating on a single English-language model - but the direct use of multilingual models is more efficient and reduces errors. The prototype implementation plans to use the Hugging Face Transformers library with models such as face/mbart-large-50 for possible command translation and XLM-R for syntactic analysis in a multilingual view. This will provide language support for most European languages used in NATO, including Polish, English, French, Spanish and many others.

F. Interoperability and data security layer

The C2 system cannot operate in isolation - it must work seamlessly with existing command platforms and battlefield sensors [8]. Therefore, the architecture includes an interoperability layer, responsible for exchanging data with external sources and publishing results to other systems. It implements interfaces in accordance with NATO standards, such as C2 Information Services (C2IS) or MIP Baseline protocols for exchanging situational data. For example, information on the position of one's own troops can arrive in APP-6(B) or ADatP-3 format, reconnaissance data in the STANAG 4559 standard, and reconnaissance video in the STANAG 4609 standard - the interoperability layer translates these streams into a form understandable by the GIS engine (e.g., converting positional metadata to geo-layers) [9]. This ensures that our system can serve as an overlay to existing C4ISR networks, taking data from them and supplementing it with voice interaction.

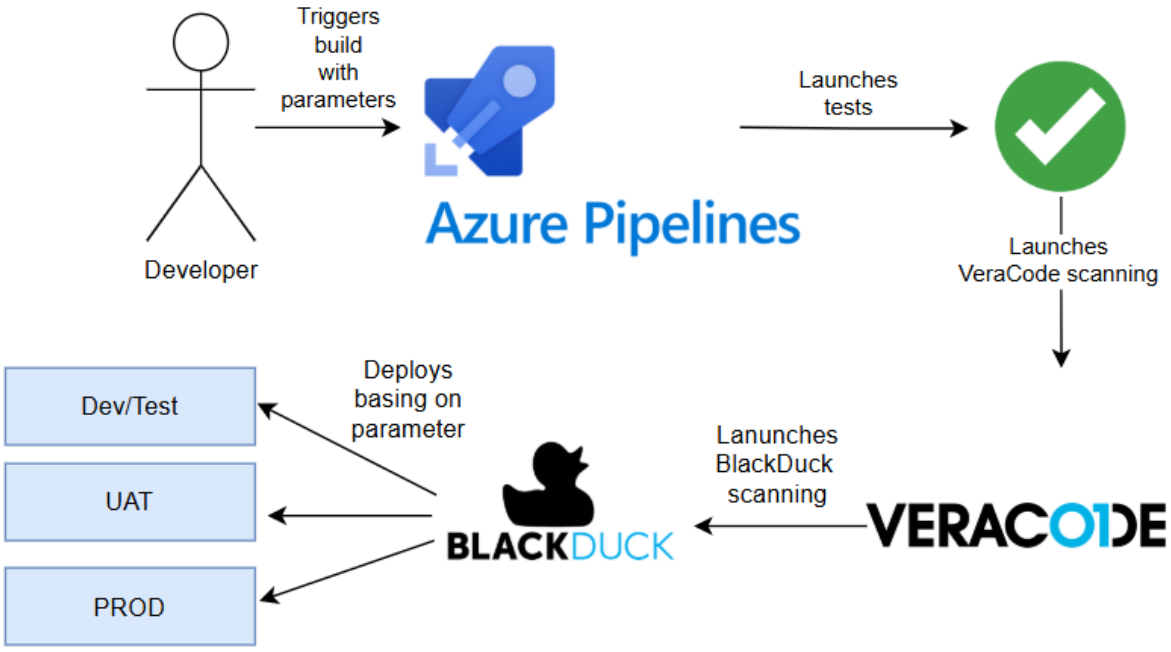

Figure 2. High-level Voice-Driven C2 system implementation architecture.

The individual steps of the CI/CD process shown in Figure 2 represent a multi-stage verification of the components responsible for the correctness of the processed data.The user (operator) can use a field device (e.g., a tablet or computer with a microphone) with an application web interface (React frontend with embedded OpenStreetMap). Voice commands are sent over a secure channel (encrypted, e.g. HTTPS over a 4G/5G network or radio link) to an application server running in a private cloud or on command center infrastructure. There, a Python backend with AI and GIS modules (the so-called Voice C2 Service) resides. The backend processes commands and

communicates with both external NATO C2 systems (via an interoperability layer providing standards-compliant APIs) and geo services (e.g., retrieves current map data from OSM databases or map servers). The whole thing can be deployed in a container environment (e.g. Kubernetes) to ensure scalability and high availability. If connectivity to the central cloud is lost, the system can operate locally at the edge of the network (edge) with limited functionality - this ensures continuity of operations under battlefield conditions.

The interoperability layer also ensures information security. In military systems, it is crucial that sensitive data is not leaked and that only authorized users have access to the system. The architecture provides for user authentication and authorization mechanisms (e.g., integration with military directory services and PKI), and communication between components is encrypted. Geographic data and query results can be marked with clauses (unclassified, confidential, etc.) depending on the source and content - the system will respect security policies, deciding, for example, whether it can include classified information in a response to a given user. The use of containers and service isolation further enhances security: AI modules can be isolated from the external network to prevent unauthorized access, and any AI models used in the system can be deployed on closed infrastructure (without communication with the public Internet).

G. Scalability and reliability

The designed system is conceived as cloud-native, which means it is easy to scale horizontally as workloads increase. The use of containerization (Docker) and orchestration (e.g. Kubernetes) makes it possible to run multiple instances of the backend server handling commands from different users in parallel. The microservice architecture allows you to scale individual components independently - for example, in a situation of intensive use of speech generation you can raise more instances of the ASR module, and for complex spatial analysis allocate more resources to the GIS engine. Load-balancing mechanisms will direct traffic to the appropriate instances, ensuring smooth operation of the system even with many simultaneous requests.

Reliability (robustness) is key in command applications - so the architecture provides for redundancy of critical components. Servers can be duplicated in different locations (e.g., in the command center and backed up in the tactical cloud) so that the failure of one does not stop the system. The ASR module and key models can run locally on edge devices offline in case of loss of connectivity - for example, if a patrol loses contact with a server, the commander's tablet can still recognize basic commands and perform simple operations on a pre-loaded map. Once connectivity is restored, any new data (e.g., query results) synchronizes with the central server. In addition, the AI models used have been selected for fault tolerance - Whisper shows resistance to various accents and interference, and GPT-4 can correct ambiguities in speech through conversational feedback. If the system misunderstands a command, it can ask the user to clarify instead of returning an error.

By using cloud technologies, the system can be deployed flexibly: entirely in a secured private cloud (e.g., on NATO infrastructure), partially on field servers (e.g., mobile staff servers) or even on a single standalone laptop for testing purposes. In each of these cases, the component architecture remains consistent - only the scale and deployment configuration changes.

Prototype implementation

As part of the prototype work, a preliminary implementation of the system was prepared in the following technology stack: the backend was realized in Python language (which facilitates the integration of AI models and GIS libraries), while the frontend as a web application in React technology. Communication between the frontend and the backend is via REST API (with HTTP/HTTPS protocol). The frontend uses the Leaflet library with a map layer based on OpenStreetMap - this provides an interactive map base on which query results (e.g. routes, object markers) are displayed. Leaflet also allows for easy overlay of additional WMS/WMTS layers, which was used to represent data such as satellite images or vector maps from military servers.

A microphone handling module has been implemented on the browser side - using the Web Speech API available in modern browsers, a user's speech is recorded, which is then sent to the backend (either as an audio stream or after initial conversion to text by the browser mechanism). Due to greater reliability and the need to tune for

military vocabulary, the ultimate plan is to send the raw audio to the backend and use the Whisper model there for speech recognition.

The Python backend has been organized into services corresponding to the modules of the architecture. The FastAPI framework was used, which facilitates the exposition of REST endpoints and asynchronous handling - for example, one of the /voiceCommand API methods accepts audio data or command text and performs all the described steps (ASR -> NLP -> GIS -> generative AI), returning the result in the form of a JSON structure (containing possible response text and data for visualization on the map). The AI modules were implemented using the PyTorch and Hugging Face Transformers libraries - this allowed loading the Whisper model (version large) and using the GPT-4 API through the OpenAI library. The GIS engine is based on the GeoPandas library combined with the PostGIS geodatabase engine; in addition, the OSRM routing service running in the container was integrated (which enabled fast route calculations based on OSM data).

Integration with external C2 systems has been simulated by simple REST services generating test data (e.g., random positions of own units). The target version plans to use real interfaces - e.g., connecting to the JCATS simulation system or the MIDB (Modernized Integrated Database) via MIP-compliant adapters [10].

The first tests of the prototype showed correct operation of the key paths: the system correctly recognizes commands in Polish and English, performs simple map operations (searches for objects, draws routes) and returns the generated text summary. Example scenario: on the command “Show the shortest route from the Alfa base to the Bravo point and specify the distance” - the application marked the desired route on the map and displayed the message “The shortest route from Alfa to Bravo is 12.4 km long (by local roads, expected travel time 20 minutes).” The efficiency of the decision-making process was improved - the operator did not have to manually search for points on the map or count kilometers, which shortened the “Orient” phase and allowed to move more quickly to the decision on route selection.

Conclusions

The concept of a voice-controlled C2 map system presented in the article demonstrates how the latest advances in artificial intelligence can be used to improve decision-making in military operations. By combining advanced speech recognition, natural language understanding of large models (LLM) and classic GIS technologies, the proposed solution enables commanders and analysts to quickly obtain the spatial information they need and interpret it - without tedious interface maintenance. The system is in line with the trend of digitizing the battlefield and increasing situational awareness through more natural human-machine interactions.

The system's architecture was designed with military realities in mind: it is modular and easily expandable, allowing adaptation to different theaters of operations and integration of new data sources. Compliance with NATO standards ensures that the solution can be seamlessly deployed alongside existing command tools, using and complementing data already collected (the plug-and-play principle of the C4ISR ecosystem). At the same time, attention has been paid to security and reliability issues - critical in combat applications - so that the system can be a trusted support rather than a burden to the user.

The results of the initial implementation are promising: the voice interface works smoothly in Polish and English, and the generative AI provides useful summaries of map information. Further work is planned to conduct more extensive tests with military analysts, verify the system's effectiveness in training scenarios, and optimize AI models for operation in limited hardware environments (e.g., on mobile devices with lower computing power). In addition, the addition of a text-to-speech (TTS) module is being considered so that the system can not only display, but also read aloud the generated reports - which would further close the voice communication loop and may take the operator's eyes off the screen.

In summary, the proposed Voice-Driven C2 system represents a step toward more natural user interfaces in military command systems. By reducing the time and effort required to obtain information from maps and making better use of the data (through automated analysis and summarization), the solution has the potential to increase the speed of command operations and enable more informed, data-driven decisions - which in battlefield conditions translates into an operational advantage.

Acknowledgments

The work was financed by the Military University of Technology as part of the project No. UGB 531-000023-W500-22.

Bibliography

1. Boyd, J. *A Discourse on Winning and Losing*. U.S. Air Force, 1987.
2. Zarazaga-Soria, F.J., Martín-Segura, S., de Larrinzar, J.L., et al.. *First Steps toward Voice User Interfaces for Web-Based Navigation of Geographic Information: A Spanish Terms Study*. Applied Sciences, 13(4), 2083, 2023.
3. Devlin, J., et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Proc. of NAACL 2019.
4. Vaswani, A., et al.. *Attention Is All You Need*. Proc. of NeurIPS 2017.
5. Haklay, M., & Weber, P. *OpenStreetMap: User-Generated Street Maps*. IEEE Pervasive Computing, 7(4), 12–18, 2008.
6. OpenAI (2023b). *GPT-4 Technical Report*. arXiv:2303.08774.
7. Radford A., Wook Kim J, Xu T., Brockman G., McLeavey Ch., Sutskever I: Robust Speech Recognition via Large-Scale Weak Supervision, Electrical Engineering and Systems Science , 2022.
8. NATO Communications and Information Agency (NCIA). *Modernization of Command and Control Systems*. Brussels. 2022.
9. NATO Standardization Office. *STANAG 4609: NATO Digital Motion Imagery Standard*. Bruksela, 2015.
10. William, I.O. *Geo-Spatial Analysis with Large Language Model*. International Journal of Advanced Natural Sciences and Engineering Researches, 9(1), 1–9, 2025.