2. Information technology - Artificial intelligence - Management system, International standard No. ISO/IEC 42001 (2023). https://www.gsc-co.com/wp-content/uploads/2024/08/SCAN-ISO-420012023_-Web.pdf
3. Kapoor, S., Girija, S. S., Arora, L., Pradhan, D., Shetgaonkar, A., & Raj, A. (2025). Adversarial Attacks in Multimodal Systems: A Practitioner's Survey. (pp. 1–8). https://doi.org/10.48550/arXiv.2505.03084
4. National Institute of Standards and Technology. (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf
5. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence and amending various regulations and directives (Artificial Intelligence Act) (2024, June 13). Official Journal of the European Union, L 1689, 1–50. http://data.europa.eu/eli/reg/2024/1689/oj
6. Sullivan, M. (2025, February 4). The EU AI Act's Implementation Timeline: Key Milestones for Enforcement. Transcend. https://transcend.io/blog/eu-ai-act-implementation-timeline
7. Zoting, S., & Shivarkar, A. (2025, June 2). Artificial Intelligence (AI) Market Size to Hit USD 3,680.47 Bn by 2034. Precedence Research - Market Research Reports & Strategic Consulting. https://www.precedenceresearch.com/artificial-intelligence-market

OBJECT-ORIENTED APPROACH TO SOFTWARE DESIGNING FOR CALCULATING MATERIAL NEEDS BASED ON SKETCHES IN GARMENT PRODUCTION

Murava Vladyslav
Bachelor Student
Zalutska Olha
Teacher
Didur Volodymyr
Bachelor Student
Mazurets Oleksandr
Ph.D in Engineering Science, Associate Professor
Computer Science Department
Khmelnytskyi National University, Ukraine

The increasing complexity and customization demands of modern garment production have rendered traditional methods of calculating material requirements both inefficient and error-prone, underscoring the urgent need for intelligent software solutions that bridge the gap between creative design and resource planning. In this context, an object-oriented approach to software design emerges as a particularly relevant paradigm, as it allows for the direct modeling of real-world entities – such as sketch components, fabric types, seam allowances, and pattern pieces – within the architecture of the system. By encapsulating these domain concepts into well-defined classes with clear interfaces, the software can mirror the conceptual workflow of designers and production planners, fostering a more intuitive and maintainable codebase. Moreover, as garment sketches evolve from mere hand-drawn outlines into richly annotated digital assets, the software must dynamically interpret shape, scale, and stylistic details in order to translate visual information into accurate material estimates. Object orientation streamlines this translation by enabling polymorphic handling of diverse sketch elements, so that a sleeve piece, a collar segment, or a decorative trim can each supply its own logic for calculating fabric yardage or notch requirements without entangling the core engine in brittle conditional logic.

Beyond maintainability and extensibility, the object-oriented approach confers decisive advantages in terms of collaboration and iterative development. In the fast-paced world of fashion, design teams often work in parallel on multiple style variants, necessitating a software backbone capable of incorporating new garment classes and material specifications on the fly. Through inheritance and composition, new design modules can be integrated seamlessly into the existing framework, reusing common behaviors – such as seam layout algorithms or marker optimization routines – while permitting specialized overrides for novel construction techniques or sustainable fabric blends. This modularity not only accelerates time to market by reducing the overhead of customizing the system for each collection but also supports the ongoing evolution of fashion software toward more holistic digital ecosystems, where upstream sketch data flows effortlessly into cut-planning, cost estimation, and inventory management subsystems.

Furthermore, the relevance of an object-oriented software design is magnified by the growing emphasis on sustainability and lean manufacturing within the garment industry. Waste reduction is a critical objective, and precise material calculation based on early-stage sketches can drastically minimize off-cuts and over-ordering. By representing fabric roll widths, pattern nesting rules, and yield constraints as first-class objects, the system can perform advanced optimization analyses – identifying the most efficient layouts and suggesting alternative grouping strategies – without compromising the clarity of the code. This transparency is essential not only for developer productivity but for auditability and regulatory compliance, enabling stakeholders to trace how each material decision emerges from sketch geometry and business rules. In sum, the object-oriented design of material-planning software grounded in garment sketches addresses the pressing needs of customization, agility, and sustainability in contemporary fashion manufacturing, providing a robust technical foundation for future innovations in digital apparel workflows.

The advent of convolutional neural networks has opened a transformative pathway for automating the translation of garment sketches into precise material requirement estimates, thereby bridging the traditional gulf between creative design intent and production planning [3, 4]. Convolutional layers, with their capacity to learn hierarchical spatial features, excel at extracting the subtle contours and texture cues embedded in two-dimensional fashion drawings, enabling a system to discern sleeve shapes, collar curves, seam placements, and panel proportions directly from raw pixel data [5, 6]. By training such networks on large corpora of paired sketches and actual cut patterns – augmented with ground-truth measurements of fabric consumption – the model learns to associate visual motifs with real-world material metrics, effectively internalizing the principles of marker efficiency and material yield without the need for hand-coded rules or brittle heuristics [7].

During training, early convolutional filters specialize in detecting lines and edges corresponding to sketch strokes and pattern outlines, while deeper layers aggregate these primitives into complex representations of garment modules such as bodices, pant legs, or decorative flounces [8, 9]. This layered abstraction allows the network to generalize across stylistic variations – sketches rendered with watercolour washes, fine technical diagrams, or freehand pencil strokes all feed into a unified system that can discern the underlying construction logic [10, 11]. Crucially, the CNN's receptive fields can be adapted to capture both local details – such as notch positions or darts – and global shape contexts, ensuring that material estimates account for both small trim allowances and the overall silhouette dimensions [12]. Residual connections and dilated convolutions further enhance the model's ability to correlate distant sketch elements, for instance linking a waistband indicator with the adjacent skirt panel, thus reflecting the interdependence inherent in nested marker layouts [13, 14].

The purpose of the work is designing and creating of software for calculating material needs based on sketches in garment production by object-oriented approach.

To correctly implement the functionality of the future application, you first need to design the algorithms of the software system functions according to the task formed above.

To authorize the user, you need to create a method that will accept the entered password for accessing the program. When this method is started, the program checks whether the entered password is correct. If the password is incorrect, the system returns the error "Incorrect password". If the password is correct, the authorization is successful.

To calculate the required number of materials, you need to create a method for accepting the values of the entered data. Before calculating, the program must check whether the entered data is correct, that is, whether they are numbers, if not, the system returns the error "Incorrect data entered", if so, the number of required materials is calculated. The program writes this data to a variable and returns the calculated values to the form. The user can save this data.

In accordance with the previous, you need to create an algorithm for viewing saved data. To do this, you need to create a method that will accept the name of the product. Then the system will check whether there is saved data about this product. If there is no data, the message "There is no data on the selected product" is displayed. If there is, the system displays the available information about the product.

The calculation of the amount of required materials S (for example, T-shirts) is carried out in the following way:

$$S = (L_{FrSide} \cdot W_{FrSide} + L_{BSide} \cdot W_{BSide} + L_{Sleeve} \cdot W_{Sleeve} \cdot 2) \cdot n,$$

where L_{FrSide} is the value of the length of the front part, W_{FrSide} is the value of the width of the front part, L_{BSide} is the value of the length of the back part, W_{BSide} is the value of the width of the back part, L_{Sleeve} is the value of the sleeve length, W_{Sleeve} is the value of the sleeve width, n is the number of products that the user needs.

The cost calculation for the previous example will be implemented as follows:

$$M = ((L_{FrSide} \cdot W_{FrSide} + L_{BSide} \cdot W_{BSide} + L_{Sleeve} \cdot W_{Sleeve} \cdot 2) \cdot n) \cdot M_{cloth},$$

where L_{FrSide} is the value of the length of the front part, W_{FrSide} is the value of the width of the front part, L_{BSide} is the value of the length of the back part, W_{BSide} is the value of the width of the back part, L_{Sleeve} is the value of the sleeve length, W_{Sleeve} is the value of the sleeve width, n is the number of products that the user needs, M_{cloth} is the price per 1 m^2 of fabric.

The class diagram by object-oriented approach was created of software for calculating material needs based on sketches in garment production (Figure 1). The diagram shows classes that inherit from the Forms system class, such as PantsForm, TshortForm, KoftaForm, JacketForm, ShortsForm, which are screens for determining the required amount of materials, accept data entered by the user. The JacketData, KoftaaData, ShortsData, TshirtData, Form1, BookData classes display saved data on the calculation of materials for products, BookForm is a form for saving data on new materials to the materials directory, Login_form is an authorization window.

Figure 1. Object-oriented class diagram of software for calculating material needs

The class diagram contains the CalculateInterface, EnterInterface, OpenFormsInterface, SavingInterface, ShowDataInterface interfaces, which are implemented using the corresponding classes.

There are two more classes in the program, namely the Resources class and the internal Settings class. The Resources class includes local program resources, and the Settings class is an internal class responsible for the system configuration of the software.

The object-oriented approach presented in this work has demonstrated substantial value in streamlining the process of translating two-dimensional garment sketches into precise material-requirement calculations, thereby addressing one of the most persistent inefficiencies in apparel manufacturing. By encapsulating the distinct characteristics of each garment component – fabric type, pattern complexity, seam allowances, and waste factors – within discrete, reusable software objects, the system achieves an unprecedented level of modularity and adaptability. This design paradigm not only facilitates rapid prototyping and extension of the core functionality but also ensures that future enhancements – such as the integration of new fabric categories, cutting techniques, or production constraints – can be incorporated with minimal disruption to existing code. Moreover, the clear separation of concerns inherent in the object-oriented model significantly reduces the likelihood of calculation errors, as each object is responsible for a narrowly defined subset of the overall task, enabling more thorough testing and validation at granular levels.

Looking ahead, the prospects for applying this solution extend well beyond the confines of conventional garment manufacturing. In an industry increasingly driven by mass customization and just-in-time production, the ability to generate accurate fabric estimates directly from designers' sketches or 3D models could become a critical competitive differentiator. Integration with digital design platforms and cloud-based collaboration tools would allow designers, technologists, and supply-chain managers to work in concert, accessing real-time material estimates that reflect both aesthetic objectives and cost constraints. Furthermore, coupling the system with advanced machine-learning algorithms could enable predictive modeling of material yield under varying production conditions, thereby supporting dynamic pricing strategies and reducing inventory overhead.

On the technological frontier, the architecture lends itself naturally to incorporation within Internet-of-Things ecosystems on the factory floor. Smart cutting tables and automated cutters could consume the system's object definitions directly, eliminating manual data entry and further reducing cycle times. Such an end-to-end digital thread – from conceptual sketch to physical production – holds the promise of dramatically shortening time-to-market for new styles while simultaneously minimizing fabric waste, thereby supporting both economic and environmental sustainability objectives. Additionally, the modular software components can be reused in adjacent industries where material optimization is critical – such as upholstery, industrial textiles, and composite-material layup – opening avenues for cross-domain collaboration and knowledge transfer.

References

1. SinglePrint. Design, construction and tailoring of clothes. URL: https://singleprint.com.ua/ua/tailoring/ (date of access: 18.06.2025).

2. Accounting programs for garment production. A4 Group. URL: https://a4.com.ua/1s-dlya-shvejnogo-virobnitstva/ (date of access: 18.06.2025).
3. Kharysh I. Designing CNN Neural Network Model for Detecting Fractures of Lower Extremities by X-ray Images / I. Kharysh, O. Sobko, O. Mazurets // The Impact of Scientific Research on the Development of the Modern World. Proceedings of the XLIV International scientific and practical conference. – Dubrovnik, Croatia, 2024. – P. 91–96.
4. Novak Y. Practical Application of Method of Automated Personal Identification by Fingerprints Using Convolution Neural Networks / Y. Novak, O. Mazurets // Proceedings of the V International Scientific and Practical Conference «Modern Strategies of Global Scientific Solutions». – Stockholm, Sweden: International Scientific Unity, 2023. – P. 136–140.
5. Hladun O. Real Time Detection the Person Emotion State Using Neural Network / O. Hladun, O. Mazurets, M. Molchanova, O. Sobko // Scientific Research: Modern Innovations and Future Perspectives.– Montreal, Canada, 2024. – P. 119-123.
6. Zalutska O. O. Method of preventing failures of rotating machines by vibration analysis using machine learning techniques / O. O. Zalutska, O. V. Hladun, O. V. Mazurets // Radio Electronics, Computer Science, Control. – 2025. – Vol. 1. – P. 142–152.
7. Mazurets O. Object-Oriented Intelligent System for Neural Network Detection of Sugar Crystallization Zones / O. V. Mazurets, V. I. Klimenko, M. O. Molchanova, A. V. Sultanov // Global Science: Prospects and Innovations. Proceedings of the 10th International scientific and practical conference. – Liverpool, United Kingdom: Cognum Publishing House, 2024. – P. 198–207.
8. Zharnovskyi O. Approach to Identification of Artificial Intelligence-Generated People Images by Means of Machine Learning / O. Zharnovskyi, O. Mazurets, O. Sobko // Key Aspects of the Development of Scientific Research in Modern Conditions. Proceedings of the XLV International scientific and practical conference. – 2024. – P. 69–73.
9. Molchanova M. Object-oriented model for neural network damage detection of mail packages / M. Molchanova, O. Mazurets, V. Klimenko, Ev. Kuflevsky // Solving Scientific Problems Using Innovative Concepts. – Copenhagen, Denmark, 2024. – P. 58-62.
10. Pokhytun A. Method for Neural Network Detecting Changed Images of People's Faces Using CNN / A. Pokhytun, O. Mazurets, M. Molchanova, O. Tyschenko // New Horizons in Scientific Research: Challenges and Solutions. Proceedings of the 1st International Scientific and Practical Conference. – 2024. – P. 35–40.
11. Mazurets O. An Approach to Using MobileNet CNN-model for Gesture Recognition / O. Mazurets, O. Zalutska, O. Tyschenko, A. Bohdanova // Proceedings of the XXIII International Scientific and Practical Conference «Problems of Science and Technology: the Search for Innovative Solutions». – 2024. – P. 59–64.
12. Molchanova M. O. Approach to Data Dimensionality Reduction and Defect Classification Based on Vibration Analysis for Maintenance of Rotating Machinery /

M. O. Molchanova, V. O. Didur, O. V. Mazurets // Radio Electronics, Computer Science, Control. – 2025. – Vol. 1. – P. 84–95.
13. Mazurets O. Datalogic Model for Image Recognition by Convolutional Neural Network Using Cloud Services / O. Mazurets, M. Molchanova, V. Klimenko, D. Klopotivskyi // Proceedings of the XXII International Scientific and Practical Conference «Modern Scientific Research: Theoretical and Practical Aspects». – Oslo, Norway, 2024. – P. 64–68.
14. Bohdanova A. Gesture Recognition Using a Neural Network in Real Time / A. Bohdanova, O. Mazurets, O. Sobko // Black Sea Science 2023: Proceedings of the International Competition of Student Scientific Works. – Odesa: Odesa National University of Technology, 2023. – P. 556–566.

A PROPOSAL FOR MULTITHREADING VISUALIZATION TECHNIQUES USING VISUAL STUDIO CODE

Pavliuk Alina
Student
Yaremenko Vadym
Assistant, PhD student
Educational and Research Institute for Applied System Analysis
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine

Multithreading is a widespread practice in modern software systems, especially in programs that need to optimize hardware utilization or respond to large numbers of concurrent events. When threads and interactions grow in number, the structure of concurrency logic becomes harder to follow. Synchronization primitives like mutexes or semaphores have a tendency to be scattered across files and functions within a program so that it's hard to see how threads are created and how they sync. The developers not only struggle in debugging but in also understanding the executing parallel structure of the program they are constructing. Dynamic analysis tools attempt to assist this by collecting thread activity during the execution of a program. Intel Inspector or Visual Studio's Parallel Stacks viewer [1][2] are examples. The tools provide information regarding thread activity during execution and provide live call stacks and thread states. Nonetheless, they rely on high runtime coverage, and they cannot detect bugs that do not happen under some run. In practice, this means that concurrency bugs in rare branches or under some timing conditions are likely to go undetected. Moreover, these tools must create an environment by which the program can potentially be executed securely and observed, and that is not always easy in early stages of development.

The approach researched here is based on static code analysis. The goal is to build a visualization of the multithreaded logic from source code directly, without executing the program. To achieve this, the plugin utilizes Tree-sitter [3], a parsing library that is