BeingAliveLanguage: Visualizing soil information from a design perspective to enhance multidisciplinary communication

Zhao Ma *, Teresa Gali-Izard
Department of Architecture, ETH Zurich, Zurich 8092, Switzerland

ARTICLE INFO

Keywords:
Soil texture visualization
Automated diagram generation
Landscape design
Parametric drawing

ABSTRACT

Soil forms the foundation for biotic and abiotic activities that shape landscapes over time. Effective communication and understanding of soil profiles, contents, and interactions with other systems such as vegetation and climate are crucial for multidisciplinary research and projects involving soil. A robust, comprehensible, and extendable visualization system is required to enhance communication across diverse disciplines, including landscape architecture, agronomy, and ecology.

This paper introduces the BeingAliveLanguage, an innovative, extensible visualization system for soil-centric information within a multidisciplinary communication framework. The system employs a fractal-based visual language to effectively convey vital soil information to professionals in various fields engaged in architecture, landscape design, and urban planning projects. The corresponding software, developed as a plugin for the Rhino-Grasshopper CAD environment, allows users to automatically generate easily understandable soil-centered diagrams using a node-based programming language. Designed to enhance communication in landscape, geoscience, and agriculture-related fields, the system provides critical information to support the design and decision-making process. We showcase the system’s efficacy through two extensions and by utilizing the tool in multiple real-world projects.

1. Introduction

In response to the escalating significance of environmental concerns, fostering interdisciplinary integration is crucial for the development of innovative and comprehensive solutions. The imperative for educating professionals across diverse fields in enhancing dialogue beyond their respective disciplines has never been more pressing. As landscape architects and researchers, we have engaged in numerous projects that encompass a broad spectrum of disciplines, acknowledging the pivotal role of soil science within various ecological systems. Through collaboration with experts in disciplines such as soil science, ecology, water management, and urban planning, we have recognized the need for a coherent communication approach that efficiently conveys information across these fields. Accordingly, this paper proposes a visualization system utilizing computational algorithms to streamline the visualization and comparison of soil data, ultimately fostering effective communication and collaboration in multidisciplinary environments.

Within the domain of soil science, digital soil mapping (McBratney et al., 2000; McBratney et al., 2003) has emerged as a discipline that emphasizes the visualization of soil information. Leveraging computer-assisted techniques, this approach generates digital maps that illustrate soil types and properties, typically employing a diverse array of computational methods to streamline the map creation process. Current research highlights a wide range of applications for digital soil mapping, such as the incorporation of machine learning (Heung et al., 2016), the geographic representation of soil organic matter content (Wiesmeier et al., 2011; Yuan et al., 2021; Zhao et al., 2017), topsoil physical properties (Ballabio et al., 2016), the creation of metadata schema for soil-agricultural research data (Specka et al., 2019), and the development of general soil information systems at various global resolutions (Arrouays et al., 2014; Hengl et al., 2014; Hengl et al., 2017). These studies further demonstrate an extensive variety of visualization techniques employed in mapping soil data.

Conventional digital soil mapping in the environmental sciences primarily focuses on understanding resource and soil condition distribution from a geographical and spatial perspective at a relatively large scale. This focus may prove inadequate for specific projects in fields such as landscape design, urban planning, and resource management. These
specific projects often require detailed soil information from a particular site, delving below the soil surface and taking into account varying depths. Technicians typically gather this information through soil surveys and present it as raw data to professionals. However, this format can create challenges in effectively communicating and collaborating across disciplines. Various research efforts have made strides in addressing these concerns: Yang et al. (2022) developed a framework for the preliminary delineation of soil profile horizons; Alqadad et al. (2017) documented several accidents that occurred during urban soil investigation due to outdated soil profile information; Temme and Vanvalleghem (2016) introduced a new model linking landscape and soil profile evolution based on sensitivity analysis; Pourabdollah et al. (2012) proposed a data exchange schema for soil and terrain data; Beaudette et al. (2013) developed algorithms to support data-driven approaches to common soils-related tasks.

In the realm of visualization, the chosen approach often depends on the target audience and the format of the source data. Given the limited literature on soil visualization outside of soil mapping, we conducted a broader survey of this topic within the general environmental and ecological context. A significant portion of such research falls into the data visualization category, which utilizes data obtained from various survey methods to reveal targeted information. Techniques such as reinterpretation, analysis, mathematical modeling, and optimization are employed to transform data into a format that highlights the intended information. The results of these studies are often code libraries in conventional data science programming languages (Python or R), which users can utilize and build upon (Araya et al., 2018; Nguyen et al., 2022; Serrano-Notivoli et al., 2022). Another type of visualization is typically created using web-based technology for monitoring purposes and can be accessed within an internet browser (Cope et al., 2017; Jarray et al., 2022; Deval et al., 2022). These studies often involve a backend online system that continuously collects and processes data before visualizing it on the frontend. Additionally, the utilization of software architectures that amalgamate multiple web-based technologies facilitates the creation of integrated pipelines for data processing and mapping (Zhang et al., 2022). This format enables users with minimal technical background to quickly obtain the information they seek. Additionally, research catering to specific needs employs customized visualization techniques, such as game engines (Cicckci et al., 2014), 3D modeling software (Filippucci et al., 2016), and more.

In this paper, we propose a soil-centered visualization system that functions as a diagrammatic language, acting as a communication tool to translate quantitative soil information among professionals from diverse backgrounds. This system is designed to strike a balance between simplicity and complexity, ensuring that essential information is neither neglected nor requires substantial effort to learn. The system should be capable of incorporating selected critical information based on limiting factors while also being adaptable to various forms for professionals from different disciplines and for different purposes. By enabling professionals to read and understand these diagrams, we aim to encourage a closer examination of soils and the surrounding ecological systems, fostering increased cross-disciplinary discussions in this realm.

Contributions. This paper presents the design and development of a visualization system, the BeingAliveLanguage, aimed at visualizing soil-related information for multidisciplinary research and collaboration. Our solution offers automated diagram generation that

- facilitates visualization, comparison of soil profiles, and integration of additional soil data,
- offers integrated extensions for wider applications, including Bagnouts-Gaussen diagrams and dynamic root growth, and
- empowers users to generate diagrams using a node-based, flowchart programming interface for diverse scenarios.

Our proposed visualization system introduces features that are unparalleled in previous software, significantly expanding the creative and illustrative potential of computational visualization for scientific information in geoscience and ecology, reaching a wider audience. The design orientation of the BeingAliveLanguage and its utilization of node-based programming interface enable swift iterations and scenario-based parametric design – a facet not addressed by other soil visualization tools. By bridging the gap between scientific and design disciplines, the BeingAliveLanguage aims to provide a visualization system adept at effectively conveying scientific information, fostering interdisciplinary collaboration and understanding.

This paper is structured as follows: Section 2 introduces the methodology and definition of the visual system, along with two possible extensions of the system. Section 3 details the computational algorithms derived from the definition, which support the auto-generation of the diagrams. Section 4 outlines the concept of node/flowchart programming and the software development environment. Section 5 showcases results generated by the software using surveyed soil data from real projects and discusses various challenges. Finally, Section 6 offers conclusions and outlines several future research directions.

2. Methodology

This section outlines the BeingAliveLanguage framework for visualizing soil profiles and soil water content. Additionally, we present two extensions built upon this visualization system. Our objective is to establish BeingAliveLanguage as a foundational diagrammatic system for soil-centered information, thereby promoting the development of further multidisciplinary applications within the community.

2.1. Soil composition and representation

2.1.1. Soil separates

A prevalent method for classifying soil types is the utilization of the USDA soil texture classification system (Garcia-Gaines and Franken-stein, 2015). This ternary system enables the determination of one of the 12 textural classes by inputting the proportions of the three primary soil particle groups, known as “soil separates”: sand, silt, and clay (Fig. 1).

Previous studies have demonstrated a diverse array of applications for this ternary diagram, including the representation of hydraulic information (Groendyk et al., 2015) and subsurface soil properties (Hu et al., 2004), the collection of agricultural data for dynamic soil texture prediction (Aarthi and Sivakumar, 2020), and the assessment of sensor

![Fig. 1. USDA soil texture triangle (NRCS, 1993).](image)
2.1.2. Associated components representing more intricate soil compositions. Diagrams of given soil horizons can be assembled into larger diagrams. Consequently, we can construct sectional soil diagrams corresponding to specific soil profiles (Fig. 2). By aligning the vertices of the side triangles, diagrams of given soil horizons can be assembled into larger diagrams representing more intricate soil compositions.

2.2. Extension

A key advantage of the BeingAliveLanguage visualization system is its extensibility, which has been purposefully designed. Our objective is to establish a shared foundation for multidisciplinary communication in soil-related research and projects, catering to a range of requirements. In this section, we showcase two extensions: the integration of the climate graph, such as water balance graphs and Bagnouls-Gaussen diagram (or the Gaussen diagram) (Bagnouls and Gaussen, 1957), to visualize climate data (the variation in soil water content throughout different months of the year); and the integration of root growth simulation into our tessellated soil diagram, providing a conceptual representation of the interaction between plant roots and soil. We anticipate that future users will further expand these extensions to suit their specific applications.

2.2.1. Climate graph integration

In multidisciplinary collaboration, it is crucial to connect soil information with other data, such as climate data, to provide a more comprehensive understanding of various biological and ecological processes. BeingAliveLanguage has the ability to integrate climate data extracted from water balance graphs or Bagnouls-Gaussen diagram, which describes the relationship between temperature and precipitation, to illustrate the limiting factors (mainly duration and intensity of drought) for plant growth and reproduction (Filippi, 2007).

For example, given a specific geographic location, it is possible to acquire the monthly temperature and precipitation data from the associated Bagnouls-Gaussen diagram. Subsequently, potential evapotranspiration can be estimated using temperature-based methods (Wanniarachchi and Sarukkalige, 2022) and corrected according to the latitude of the location (hours of light). This information enables the calculation of the soil’s hydraulic balance, including surplus, deficit, or reserved water amounts, for a given soil depth (Samani, 2000). Moreover, the soil water content ratio can be determined by taking into account the soil texture and maximum reserved water amount, and subsequently represented in our visualization system.

The integration process described above is illustrated in Fig. 5, where the water balance graph is overlaid on the Bagnouls-Gaussen diagram, combined with the calculated information for a typical geographic location. The comparison of soil water content at two different soil depths during two specific months, namely May and October, demonstrates the effectiveness of the BeingAliveLanguage in visualizing the combined information of soil, water, and climate using a unified visual language.

The integrated approach offered by the BeingAliveLanguage not only enhances our understanding of the interplay between soil, water, and climate factors but also uncovers insights from various disciplines. By visualizing these interconnected elements, the system supports more informed decision-making in the context of landscape design and management.

2.2.2. Root growth integration

Soil serves as the fundamental living environment for most vegetation. Plant root systems interact with the soil (Jin et al., 2017) to obtain essential water and mineral resources necessary for their metabolic activities while releasing byproducts into the soil. Kuzyakov and Blagodatskaya (2015) propose a detailed view of the microbial processes happen occurring in the rhizosphere and their ecological relevance. In one of our previously published articles (Gali-Izard et al., 2022), we described the relationship between a root and its surrounding environment from a narrative perspective. In this paper, we aim simulate the growth of a root from a technical and diagrammatic standpoint.

To graphically represent root growth within a triangular grid as described in this paper, a map providing topological information of the base grid is typically needed. Various approaches have been employed in existing research to build maps depending on the specific objectives. For instance, mixed integer programming is widely used for automated metro map generation (Oke and Siddiqui, 2015; Xu et al., 2022; Nollenburg and Wolff, 2011), with derived or extended optimization approaches also appearing in other sources (Stott et al., 2011). These

---

1 The Bagnouls-Gaussen diagram, also known as the ombrothermic diagram or Bagnouls-Gaussen graph, is a graphical representation of climate, specifically designed to study the aridity of a region. It was developed by French scientists Frederic Bagnouls and Henri Gaussen in the 1950s. The diagram plots two critical variables: temperature and precipitation, on a monthly basis.

2 There are two distinct sets of evapotranspiration estimation methods available. In our case, we utilize a temperature-based method, specifically the Thornthwaite equation.

3 Computation related to these factors has also been developed as an extra component into the BeingAliveLanguage system.
graph-based approaches focus on the topological relationships between elements. In contrast, Isikdogan et al. (2017) developed an automated river mapping technique from remote sensing data based on the Cartesian system, where actual scale and distance are of greater importance.

In this paper, to establish the connection between the root and the soil diagram, we extract the outer triangles of the particle units and treat them as a graph $G=(V,E)$. We then restrict the root to grow only along the vertices $V$ and edges $E$, using the Euclidean relationship between vertices to control the root’s scale. Consequently, we can simulate root growth in a stepwise, probability-based manner within our soil diagram and associate the root with various plants. We present several examples in Figs. 12 and 13, and describe the detailed algorithms in Section 3.3.

Although we acknowledge that there are more root types in reality, based on the Raunkiaer system (Raunkiær, 1934), a detailed exploration of these types is beyond the scope of this paper. We recommend that...
interested readers refer to the corresponding literature for more information on the subject.

3. Algorithm and implementation

This section elucidates the fundamental algorithms we devised to underpin the definition and methods detailed in Section 2. These algorithms have been implemented in the BeingAliveLanguage software as a plugin for a widely-used CAD platform. The particular software environment is discussed in Section 4.2.

3.1. Soil type classification

In Section 2.1.1, we discussed employing the USDA soil texture diagram to classify soil types according to the proportion of soil separates. This 2D diagram establishes a graphical relationship among various soil types, which can be abstracted into an algorithm to automate the soil classification process. The algorithm is derived by examining the classification criteria for each soil type and translating them into mathematical terms. Table 1 presents the complete results of this deduction, which have been encoded into the BeingAliveLanguage. It is worth noting that, due to the intricate shapes of loamy sand, silty loam, and sandy loam, the corresponding determining conditions for these three textures are also more complex than others and necessitate special consideration.

Upon accurately identifying the soil texture, we can extract the soil water holding capacity based on existing geographical data (Datta et al., 2017). The definition of the soil unit (Section 2.1.1) enables the proper integration of such data, allowing for automatic generation of detailed diagrams containing all relevant information for the current soil texture. The outstanding question pertains to the method for combining these units into a diagram that accurately represents the corresponding soil horizon.

Table 1

<table>
<thead>
<tr>
<th>Soil Type</th>
<th>$r_{sandy}$</th>
<th>$r_{silt}$</th>
<th>$r_{clay}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sand</td>
<td>(0.5$r_{day} + 0.85$)</td>
<td>NA</td>
<td>(0.00 0.10)</td>
</tr>
<tr>
<td>Silt</td>
<td>NA</td>
<td>(0.80 1.00)</td>
<td>(0.00 0.14)</td>
</tr>
<tr>
<td>Clay</td>
<td>(0.00 0.45)</td>
<td>(0.00 0.40)</td>
<td>(0.40 1.00)</td>
</tr>
<tr>
<td>Sandy Clay</td>
<td>(0.45 1.00)</td>
<td>NA</td>
<td>(0.35 1.00)</td>
</tr>
<tr>
<td>Silty Clay</td>
<td>NA</td>
<td>(0.40 1.00)</td>
<td>(0.40 1.00)</td>
</tr>
<tr>
<td>Clay Loam</td>
<td>(0.20 0.45)</td>
<td>NA</td>
<td>(0.27 0.40)</td>
</tr>
<tr>
<td>Silty Clay Loam</td>
<td>(0.00 0.20)</td>
<td>NA</td>
<td>(0.27 0.40)</td>
</tr>
<tr>
<td>Sandy Clay Loam</td>
<td>(0.45 1.00)</td>
<td>(0.00 0.27)</td>
<td>(0.27 0.35)</td>
</tr>
<tr>
<td>Loam</td>
<td>(0.00 0.53)</td>
<td>(0.28 0.50)</td>
<td>(0.07 0.27)</td>
</tr>
<tr>
<td>Loamy Sand</td>
<td>$g_{sand} \land r_{sandy} &lt; (0.70 1.00) \land r_{clay} &lt; (0.00 0.15)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Silty Loam</td>
<td>$(r_{clay} \in (0.00 0.27) \land r_{silt} \in (0.5 0.8))$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandy Loam</td>
<td>$r_{clay} \in (0.00 0.07) \land r_{sandy} &lt; (0.53 1.00) \land r_{silt} &lt; (0.00 0.50)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$g_{(loamy,sand)} \land r_{clay} &lt; (0.00 0.02) \land r_{sandy} \in (0.53 1.00)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following equations outline the process of generating a soil horizon diagram that accurately represents the distribution of three soil separates akin to the example in Fig. 2. To accomplish this, we require the correct ratio data of the soil separates $(r_{sandy}, r_{silt}, r_{clay})$ along with two
additional input parameters: a specified rectangular boundary \( (S_{\text{bbox}}) \) and a subdivision density \( p_{\text{subdiv}} \).

The algorithm initially subdivides the boundary \( S_{\text{bbox}} \) into a series of regular triangles based on the density parameter \( p_{\text{subdiv}} \) (additional post-processing techniques are applied to align triangles near the borders). Subsequently, it performs two iterations of the “selection + subdivision” processes to determine sand and silt: it selects a subset of triangles that corresponds to the area ratio, and then subdivides the remaining triangles into four smaller fractal ones. Following these two iterations, there should be three levels of triangles, each representing the soil separates of sand, silt, and clay, respectively.

Upon completing this step, we can incorporate other soil compositions, as computed by the algorithm in Section 3.1, into the diagram, thereby generating a comprehensive visual representation of the given horizon. The pseudocode of the entire algorithm is provided in Algorithm 1.

**Algorithm 1 Soil Diagram Generation**

<table>
<thead>
<tr>
<th>Require</th>
<th>( r_{\text{root}}, r_{\text{dil}}, r_{\text{bbox}}, p_{\text{subdiv}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensure</td>
<td>( r_{\text{bbox}}, p_{\text{subdiv}} ) is rectangular</td>
</tr>
<tr>
<td>function</td>
<td>SUBDIVIDE(( r_{\text{bbox}}, p_{\text{subdiv}} ))</td>
</tr>
<tr>
<td></td>
<td>Subdivide ( S_{\text{bbox}} ) into ( T_{\text{div}} )</td>
</tr>
<tr>
<td></td>
<td>for ( \text{for} ) ( i = 1 ) to ( r_{\text{bbox}} ) ( \text{do} )</td>
</tr>
<tr>
<td></td>
<td>Select ( T_{\text{div}} ) from ( T_{\text{div}} )</td>
</tr>
<tr>
<td></td>
<td>( T_{\text{div}} ) ( \Rightarrow ) FRactalDivide(( T_{\text{div}} )</td>
</tr>
<tr>
<td></td>
<td>for ( \text{for} ) ( i = 1 ) to ( r_{\text{bbox}} ) ( \text{do} )</td>
</tr>
<tr>
<td></td>
<td>return Diagram</td>
</tr>
<tr>
<td>end function</td>
<td>CREATE Soil Diagram</td>
</tr>
<tr>
<td></td>
<td>CREATE Water Content</td>
</tr>
<tr>
<td></td>
<td>CREATE Organic Matter</td>
</tr>
<tr>
<td></td>
<td>return Diagram</td>
</tr>
</tbody>
</table>

It is essential to recognize that the BeingAliveLanguage system comprises numerous individual components, and combining all the information can result in a highly compacted diagram. In real-world applications, users can choose a subset of the information (e.g., soil separates and wilting point) tailored to their specific requirements, thus producing diagrams with improved readability.

3.3. Root grow simulation

The primary factor influencing a root’s growth direction is gravity. Given that we determine the root should grow between the soil particles without breaking them, the triangle vertices naturally become the locations where a root grows from or towards. Consequently, the “actual” topology graph \( G( V, E ) \) employed by a root is similar to the one depicted in Fig. 2. Bottom.

Since such a diagram consists of three differently sized triangles, each vertex \( v_i \) is connected to 4 or 6 neighboring vertices \( v_j \), with equal or unequal Euclidean distances \( |d_{ij}| \). For each vertex, we store the direction and the distance of these neighbors and construct a hash table for the data to enable rapid querying.

For each growing step, we apply a probability function, typically a normal distribution, in all directions. This results in a probability-based growth effect towards a preferred direction while allowing for a certain degree of randomness. At each step, we only permit unary or binary branching to neighboring vertices. To control the overall size of the root, we employ a global radius \( R_{\text{grow}} \) to limit the accumulated Euclidean distances from the starting root vertex to the farthest leaf vertex (Fig. 6).

We demonstrate this algorithm with more examples in Section 5.1.

4. Software design and usage

While developing the BeingAliveLanguage as a versatile visual system for promoting multidisciplinary communication, we expected that users can employ the system independently or alongside other digital drawing tools. Consequently, we decided to integrate our software within a CAD environment, allowing for direct modification of results or additional development on top of our system. The availability of node-based programming helps us achieve this goal, offering a pliable and accessible platform for users to engage with and build upon our visual system.

Fig. 6. Vertex relationship and parameters in probability-based root growing. Left: The two typical cases where a vertex has either 4 or 6 neighbors (to keep data structure aligned, we use a null value if the neighboring vertex does not exist); TopRight: Normal distribution for vertex probability; BottomRight: Using \( R_{\text{grow}} \) for global root radius control.
4.1. Node-based programming

Node-based programming, also referred to as flowchart programming, constitutes a category of visual programming language (VPL) that enable users to construct programs by graphically connecting a sequence of "nodes" for the execution of logical operations, as opposed to delin- eating them through textual code. This methodology provides a more intuitive and accessible means for designing and developing software, particularly for individuals possessing limited programming expertise. Presently, it is among the most prevalent visual programming models employed across diverse fields, permitting users, including children, with minimal programming knowledge to participate in programming activities effortlessly. Fig. 7 depicts an array of prominent software applications for visual programming (encompassing both node-based and non-node-based) spanning various sectors and industries.

The core principle of node-based programming pertains to the employment of an abstracted node model for the representation of data or a computing operator. Each “node” accepts zero or multiple inputs and yields outputs subsequent to executing a computing operation. The scope of these computational operations encompasses elementary arithmetic tasks as well as intricate customized functions necessitating considerable computational resources. (Fig. 8).

The intricacy of project development for users is predominantly contingent upon the assortment of nodes supplied by both the software development team (intrinsic default functions) and community developers (plugin functions). A robust platform typically reaps the advantages of an active community-driven ecosystem, encompassing hundreds of plugins contributed by an array of sources. This, in turn, enables an extensive scope of functionality and customization for the end-users.

4.2. Development environment

For the BeingAliveLanguage, we chose the Rhino6-Grasshopper7 platform as the base environment. This platform is widely accepted for 3D modeling and parametric design in the architecture, landscape, and computer-aided design community. We implemented our algorithms as a plugin for the Grasshopper environment (Fig. 9), leveraging Rhino’s modeling and drawing API alongside Grasshopper’s default computing units to produce illustrative diagrams.

We adhere to several software development best practices, such as version control, issue tracking, and pull requests via GitHub, to guarantee a reliable development process. Additionally, we employ the MkDocs5 system to generate HTML documentation, which is accessible through a permanent website hosted by ETH Zurich.

The majority part of the BeingAliveLanguage has been developed using the C# language within Microsoft’s.NET Framework and is integrated into Rhino’s plugin ecosystem. The plugin can be installed on Rhino v7+ utilizing its build-in package management system.

In terms of computational complexity, the two most resource-intensive processes are building the soil map and simulating root growth. By making appropriate choices for data structures (Hash map and K-d tree) and leveraging parallel computing, we have successfully reduced the computational complexity to an O(nlogn) level. Our experiments demonstrate that constructing a soil map containing approximately 1 million vertices takes less than 10s on a modern i7-8550U laptop with 16 GB RAM. Generating diagrams at a moderate scale allows for almost instantaneous feedback and enables an interactive design process.

5. Results

5.1. Demonstrations

We have evaluated the effectiveness and interactivity of BeingAliveLanguage during diagram creation across various projects, generating soil diagrams that incorporate other elements for diverse scenarios. The subsequent examples showcase these applications. As we mentioned in Section 1, the BeingAliveLanguage visualization system is designed for flexibility, enabling users select a wide range of available computing units. Consequently, the final appearance of the diagram largely depends on the user’s choice of information to display.

Soil Profile Comparison. On numerous occasions, it is necessary to visualize quantitative soil data obtained from field surveys or laboratory analyses and convey this information with various stakeholders. Indeed, one of the primary purposes of developing BeingAliveLanguage is to facilitate communication between soil scientists and professionals from other disciplines, such as architecture, landscape design, urban planning, using a unified visual tool. Typical applications for visualizing soil information involve comparing soil conditions across different soil horizons, contrasting multi-horizon soil profiles, and illustrating various aspects within the soil, such as soil water content and organic matter. In the following, we present two examples that demonstrate the effectiveness of the BeingAliveLanguage in addressing these needs.

In the first example, we demonstrate that the BeingAliveLanguage visualization system is capable of aligning well with existing soil survey results by creating diagrams based on existing research. Fig. 10 presents the redrawn diagrams and the soil sectional photos side by side, where the diagrams facilitate comparisons between different soil profiles. It accurately represents the actual conditions and conveys essential information from soil science, establishing a solid foundation for communicating information across various disciplines.

Since the soil profiles in Fig. 10 exhibit minimal differences and do not display significant contrasts across soil horizons within the same soil profile, we present a second example featuring soil profile diagrams from two geographically distant locations, Fusagasugá, Colombia, and Suterranya, Spain. The contrasts across soil horizons, as well as between different soil profiles, are clearly demonstrated in Fig. 11.

Integration with Extra Environmental Factors. The BeingAliveLanguage has been purposefully designed for flexibility and extensibility, making it highly valuable for integrating work from various disciplines and fostering impactful multidisciplinary collaborations. In this section, we present two examples showcasing how this system has been utilized in such contexts. It is important to note that in different examples, we have deliberately colored the selected information based on the specific information we aim to deliver, further emphasizing the adaptability and customization potential of the BeingAliveLanguage system for diverse applications.

Fig. 12 displays the results from two different locations in the Senan region of Spain. In collaboration with local researchers, we examined environmental factors such as soil conditions and vegetation to plan long-term land management strategies. These diagrams facilitated communication with local experts from related fields, providing a shared visual language that transcends disciplinary boundaries.

The second example illustrates the change in soil conditions over time due to interactions with the environment and their impact on animal husbandry. Fig. 13 showcases the relationship between horse hoof treatment and soil conditions, demonstrating that better conditioned soil can support horses’ foot health by allowing them to walk with bare feet. These diagrams facilitate communication with scientists in the field of animal husbandry, promoting interdisciplinary understanding and

---


---

8. The two projects are part of a more extensive, ongoing initiative that has spanned several years. For more information: https://thegarden-senan.arch.ethz.ch.
5.2. Discussion and limitations

The presented results have many implications to consider for many future research directions. As the software is developed along several projects, some of the functionalities are not implemented into a “perfect” state. Moreover, as we have just started to use this visualization system during our collaboration with professionals from other disciplines, the BeingAliveLanguage is still in its early phase and actively collecting feedback.

5.2.1. A unified visual language for effective soil utilization in landscape projects

In addition to the digital soil mapping review in Section 1, a substantial amount of research on soil/landscape management also utilizes “mapping” or GIS-supported approaches, as it is one of the most effective ways to associate collected data with geographic information (Qu et al., 2013; Pechanec et al., 2015). These map-based studies effectively reveal the interrelationships of the data associated with geographic distribution, providing a valuable overview for targeted users or professionals. However, little research focuses on projects at the scale of building, landscape, or urban block. Resource distribution at such scales does not vary significantly geographically but varies considerably at different depths within the soil. The collected data is limited and does not differ substantially on geographic maps. Furthermore, multidisciplinary collaboration across various disciplines is more intense between different parties within these projects.

The BeingAliveLanguage is our attempt to fill this gap by creating a unified visual language that facilitates effective communication about proper soil use and related resources across various professional disciplines. To the best of our knowledge, it is the first of its kind specifically designed to target communication of soil information among multidisciplinary professionals. This challenge also pushes us to develop visualizations that deliver information with the appropriate level of detail for the intended audience: determining the right balance of qualitative and quantitative data to extract, abstract, and present, and how to tailor the visualizations for different audiences (soil scientists, architects, landscape architects, agronomists, etc.) within a unified framework.

Unlike the tools or programming language packages we reviewed in

![Diagrams showing various visual programming platforms and node-based computing processes.](image-url)
Section 1, our approach draws knowledge from design and computer graphics fields to create software specifically designed for a CAD platform, allowing for extended development. Users can select specific components separately and compose independent diagrams/drawings or integrate them into drawings of existing projects. The reader should have already noticed the varied emphasis on visualized information in the figures presented throughout this paper. This flexibility enables the BeingAliveLanguage to cater to a wide range of user needs and preferences, fostering effective communication and collaboration among professionals from diverse disciplines.

5.2.2. Challenges and potential for extensibility

While our software is developed on a specific CAD platform, it necessitates that users have a basic familiarity with both CAD environments and node-based programming. Once these prerequisites are met, users typically experience a relatively smooth learning curve when creating visualization diagrams with the BeingAliveLanguage, as evidenced by our students. Interpreting these diagrams also demands a fundamental understanding of soils, such as soil separates, water content, and organic matter. However, this information can be easily communicated to professionals from various backgrounds. One of the advantages of our approach is that once created, the drawings can be documented alongside the project as vector-based digital assets.

A significant hurdle for the BeingAliveLanguage lies in achieving its widespread adoption across various disciplines, which we are actively promoting. By engaging in an increasing number of collaborations, our goal is to establish this visualization system as a standard for visualizing soil-centered information in multidisciplinary projects.

Contrarily, our system, which is designed to provide visualizations for specific geolocations with detailed soil information and manually

Fig. 9. The BeingAliveLanguage plugin. Top: The “nodes” provided by the plugin; Bottom: Using the plugin in a diagram development process.
retrieved local climate data, envisions potential avenues to exploit existing remote sensing data and web-based APIs. For instance, the utilization of resources such as Crop-CASMA (Zhang et al., 2022) for US soil moisture data can be beneficial. This could lead to the development of integrated data streaming pipelines for visualizing soil information at varying degrees of granularity. Not only would this augment the interoperability of our system, but it would also foster enhanced multidisciplinary communication at the developmental stage, consequently bolstering collaborations across fields. Nevertheless, the prospect of identifying a unified data framework with global applicability remains uncertain to us.

Although we have demonstrated only two extensions of the BeingAliveLanguage’s usage, we envision numerous opportunities to expand the system, incorporating vegetation systems, better-integrated climate factors, a more extensive collection of root systems, and more. All of these enhancements will require additional time for validation and integration. As we receive feedback and learn about the needs of various professions, the BeingAliveLanguage will continue to evolve and improve to better serve the diverse needs of interdisciplinary collaboration.

6. Conclusions

This paper presents the BeingAliveLanguage, a first-of-its-kind, extensible visualization system for soil-centered information in a multidisciplinary communication context. The system develops a fractal-based visual language to effectively convey essential information about soil and its related resources to professionals from various backgrounds. Unlike most resource mapping tools that associate targeted data with geographic maps, the BeingAliveLanguage focus on projects at the scale of architecture, landscape design and urban planning, where soil data varies less significantly across geographic locations but in depths, generating soil sectional diagrams. The extensibility of the BeingAliveLanguage is demonstrated by two examples: the integration of Bagnouls-Gaussen diagrams and geographically-relevant climate data, and the abstracted simulation of root systems within the proposed soil system.

Fig. 10. Comparison across two soil profiles extracted from Przewoźna (2014). The photos has been processed into grayscale to eliminate visual disturbance.

Fig. 11. Soil diagrams comparing two soil profiles from the Fusagasuga, Columbia and Suterranya, Spain. Difference across soil horizons (within the same soil profile) or geographical locations (between the 2 soil profiles) can be clearly identified.
Fig. 12. Integrated and colored soil diagrams illustrating soil composition, water content, organic matter, and plant profiles from two sites in Senan, Spain. The diagrams are based on data gathered from local fieldwork surveys and laboratory analyses, offering a comprehensive visualization of the site-specific dynamics between soil properties and environmental factors.

Fig. 13. Soil diagrams illustrating the effects of different soil conditions on horse feet, showcasing the impact of varying environmental factors on animal husbandry. Credit: Uxia Varela.
The paper also presents the critical algorithms for automating the diagram generation processes in detail and implements the entire system as a software plugin for the Rhino-Grasshopper CAD environment. This enables users to design and generate such visualizations parametrically. With the demonstrated usage of the BeingAliveLanguage in various real-world projects and our ongoing commitment and dedication, we will continue to develop the BeingAliveLanguage visualization system and explore additional extensions.

Code availability section

BeingAliveLanguage

Contact: ma@arch.ethz.ch

Hardware requirements: x64 PC (Windows OS).

Program language: C#

Software required: Rhino 3D v7.0, Grasshopper v1.0 (shipped with Rhino 3D).

Program size: 1.6 M (this software only).

The source codes are available for downloading at the link: https://github.com/ChairBeingAlive/BeingAliveLanguageGH

The documentation of the software available at the link: beingalive_language.etzd.arch.ethz.ch

CRediT authorship contribution statement

Zhao Ma: Methodology, Software, Writing-original-draft, Writing-review-editing.

Teresa Gali-Izard: Conceptualization, Writing-review-editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to express their gratitude to Luke Harris, Cara Turett, Bonnie Kathleen Walker, and Uxia Varela for engaging in insightful discussions on the topics, as well as for their generous support and sharing of their previous work with the authors.

References


Z. Ma and T. Gali-Izard
Z. Ma and T. Gali-Izard


