Cognitive Aspects of 2D Content Integration and Management in 3D Virtual Reality Spaces

Tarek Setti*, and Ádám B. Csapó†‡

Abstract—**The advent of 2D graphical user interfaces in the 1980s shifted user interactions from line-based terminals to icon-based interfaces. As smartphones emerged in the 2010s, portable 2D graphical interfaces became a reality, liberating users from being confined to a single location when accessing digital services. These transformations have profoundly altered our understanding of digital information systems, with impacts that cannot be easily quantified. Current advancements in virtual and augmented reality (VR/AR), the Internet of Things (IoT), and artificial intelligence (AI) are on the verge of ushering in the next significant leap in cognitive expansion, introducing portable and highly contextual spatial interfaces, also sometimes referred to as Digital Realities (DRs). As a result, users now anticipate the ability to engage with an increasing array and variety of digital content in ways that are more contextualized and tailored to their needs, taking into account factors such as time, location, personalized goals and user-specific histories. In this paper, we aim to give an overview of cognitive aspects relevant to content integration and management specifically in DR environments, and to propose solutions and / or best practices to address them. Our discussion is centered around a paradigm called the Doing-When-Seeing (DWS) paradigm, which we propose for the design of Digital Reality interfaces. We demonstrate the applicability of this paradigm to the design of interfaces for creating content, organizing content, and semantically representing and retrieving content within 3D Digital Reality environments.**

Index Terms—**content management; digital reality; cognitive aspects of virtual reality**

I. INTRODUCTION

Technological development has played a significant role in shaping human civilization, from the earliest tools and machines to the modern-day innovations that are transforming the world [1], [2], [3]. Two of the most prominent technological trends in recent years is the rise of artificial intelligence – including machine learning and deep learning – on the one hand [4], [5], and advances in 3D spatial technologies on the other [6]. Machine learning and deep learning technologies are being used across industries to analyze data, automate processes, and make predictions [7] – a tendency motivated by multiple factors, including the need for more automation, the desire to improve decision-making, and the need to accomplish this in the face of a growing volume of information. In turn, virtual reality and other 3D spatial technologies – including even 3D digital twins – have been shown to be instrumental tools in presenting users with more information at a lower cognitive load, thereby increasing the interpretability of complex physical-digital scenarios [8], [9], [10], [11]. The merging of these fields is rapidly leading to new synergies, as demonstrated for example in the definition of Digital Reality and Internet of Digital Reality [12], [13].

A key challenge in the use of digital realities is how to provide users with the information they need at the right time and location. Such questions are increasingly important due to a confluence of multiple factors:

- With the growing prevalence of mobile as opposed to desktop computing, users have become less tethered to any single location; hence, context-aware information retrieval is becoming the norm. This problem can be referred to as *finding the content set appropriate to the given context*;
- With the growth of data volume in users' digital life, two further trends can be observed:
  - Users are increasingly motivated to organize and manage their own information spaces; this means that users expect to be able to curate and organize their own digital content within their (3D) digital applications instead of merely consuming content created by others
  - It is increasingly challenging to present to users all of the information relevant to a given context, which can be difficult to find (i.e., filter out based on semantic relevance) and also risks inducing a high cognitive load

  Together, these two trends create the challenge of *organizing content sets and making them amenable to intuitive exploration.*

With respect to the first factor – i.e., the need for context-aware content curation – 3D digital environments can be hugely effective, as they are already spatial in nature, in a way that closely mirrors human thinking through spatial metaphors [6]. To formulate this in everyday terms: A 3D virtual classroom environment can be easily conceptually associated with the activity of learning; while a 3D virtual home

---

[*] Doctoral School of Multidisciplinary Engineering Sciences, Széchenyi István University, Győr, Hungary (e-mail: setti.tarek@hallgato.sze.hu).

[†] University Research and Innovation Center, Óbuda University, Budapest, Hungary

[‡] Eötvös Loránd Research Network, Budapest, Hungary (e-mail: csapo.adam@uni-obuda.hu).

DOI: 10.36244/ICJ.2023.6.5

cinema environment can be easily conceptually associated with leisurely activities such as watching films or listening to music. In this way, content relevant to an appropriate context can be easier to organize and retrieve [14].

The second factor – i.e., the data deluge that increasingly surrounds users – also brings about a unique set of challenges, especially when it comes to users inhabiting 3D virtual environments. In this regard, key aspects range from the mechanics of laying out content within a 3D virtual environment, all the way to the high-level organization and retrieval of information from large content sets. For example, a 3D virtual classroom might host content sets relevant to a large variety of subjects. Depending on the number of subjects available and the number of lessons within each subject, the classroom as a physical location might lose its relative importance, and the main challenge would become finding the particular lesson a user wants to work with at any given time. Clearly, there is a tradeoff here between creating new spaces and adding content to an existing space. However, even when adding new content to an existing space, the question of what operations should be available to users in helping them to lay out their content is far from trivial.

In this paper, we introduce a paradigm which we refer to as the "***Doing-When-Seeing (DWS) paradigm***". DWS is a generic design philosophy and methodology that can help address a variety of challenges, including those of 3D content layout creation, 3D content set organization, and contextual-semantic content retrieval.

The paper is structured as follows. In Section II, we briefly introduce some key nomenclature with respect to virtual reality spaces that can help readers better understand the discussion in later sections. In Section III, we introduce some of the key cognitive challenges we have identified in the context of 2D content integration and management in 3D digital realities. This is followed by an introduction to the Doing-When-Seeing paradigm, in Section IV. Finally, in Sections V–VII, we demonstrate the viability of DWS by providing examples of all of the aforementioned applications along with a set of associated experimental validations.

II. KEY DEFINITIONS

To facilitate further discussions on the topic of 2D content integration and management, we introduce the following terms:

Virtual camera: The unique viewpoint in a 3D virtual space through which the 3D scene appears to the user's eyes at any given time. The virtual camera is characterized by a 3D location (x,y,z spatial coordinates) and a spatial orientation (determined by a 4-dimensional quaternion).

2D display panels (or 2D displays): Bounded rectangular surfaces within a 3D space that are used to display 2D content, including e.g. PDF files, webpages, images or videos. A 2D display panel is characterized by its 3D location (x,y,z spatial coordinates of its geometric center), a spatial orientation (determined by a 4-dimensional quaternion), a width and a height.

2D content layouts: Sets of 2D display panels appearing in the same 3D space at the same time. One can think of a content layout as a project containing 2D documents that are laid out on rectangular display panels in a 3D virtual space.

2D content integration and management: A set of tasks associated with the specification, creation and retrieval of 2D content layouts, together with their constituent 2D display panels.

III. COGNITIVE CHALLENGES BEHIND 2D CONTENT INTEGRATION AND MANAGEMENT

In this section, we provide a brief overview of key cognitive challenges we have identified with respect to 2D content integration and management in Digital Reality spaces.

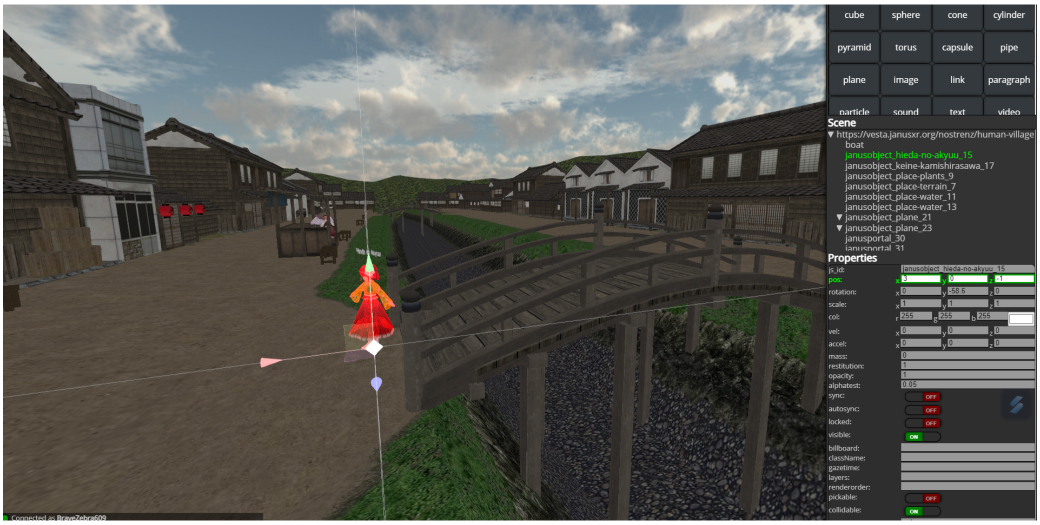
A. Challenges associated with the creation of 2D display panels

Based on an overview of the literature and currently widely adopted best practices, we have identified two main challenges that pertain to the creation of 2D display panels in particular, which we refer to as the *camera-object independence dilemma* and the *lack of higher-order structural operations*. These two terms can be defined as follows:

- The term **camera-object independence dilemma** describes the tension that arises between the desire to position 2D display panels in relation to the virtual camera (as it is positioned and oriented at the current point in time), and in relation to the surface normals of 3D objects within the environment. In particular, if the location / orientation of the virtual camera is changed, so as to enable the 2D display panels to be aligned with the surface of an object more precisely, the former goal – which relies on the camera pose remaining stationary – is immediately undermined. This challenge is particularly clear when a so-called 3D gizmo is used to edit the pose and scale of objects in a 3D virtual scene. In this case, the virtual camera either remains stationary, in which case it is impossible to see whether the object being manipulated is properly aligned with its surrounding objects; or the camera viewpoint changes, in which case the user then has to find a suitable location from which the gizmo can be accessed again (Figure 1);
- The term **lack of higher-order structural operations** describes the lack of support on many commonly used interfaces for the intuitive manipulation of higher-order structures within 2D content layouts. In particular, the operations carried out during the creation of 2D display panels and 2D content layouts can become tedious without the ability to change the scale at which manipulations can be made.

B. Challenges associated with content retrieval in digital realities

Based on a literature review in [15], we have shown that an overwhelming majority of current solutions towards semantic document retrieval depend on predetermined semantic connections, either explicitly defined or extracted from human-created datasets.



js_id: janusobject_hieda-no-akyuu_15
pos: x 3 y 0 z -1
rotation: x 0 y -58.6 z 0
scale: x 1 y 1 z 1
col: r 255 g 255 b 255
vel: x 0 y 0 z 0
accel: x 0 y 0 z 0
mass: 0
restitution: 1
opacity: 1
alphatest: 0.05
sync: OFF
autosync: OFF
locked: OFF
visible: ON
billboard:
className:
gazetime:
layers:
renderorder:
pickable: OFF
collidable: ON

Fig. 1. Example of a 3D gizmo used to change the position of an object in the scene. The camera-object independence dilemma is generated by the fact that conceptually for the user, the camera and the object should be independent (allowing users to change their viewpoint in order to ascertain whether the object being manipulated is properly aligned with its surroundings), whereas the use of the gizmo tethers the camera to the object and the 3D gizmo, making it cumbersome to change the virtual camera orientation and to access the gizmo again.

However, it can be expected that users' thinking and memory recall will often be guided by associative and episodic factors. This means that, for instance, users will often recall content sets based on memories of the time, location or topic in the context of which they last accessed them. This suggests that instead of relying exclusively on similarity of meaning, a retrieval system might also rely on past episodes of interaction as a surrogate for semantic similarity – including, for example, by linking unsuccessful search entries to subsequent search entries resulting in successful document retrieval. Such a system would be capable of generating personalized, evolving semantic labels through the course of its normal use.

When it comes to creating document store and retrieval systems with adaptive semantic labels, we have identified the following 3 challenges:

- **syntactic saturation:** the number of associations in a semantic retrieval system reaches a point where several search entries have a similar syntax, thereby becoming easy for users to mistype;
- **semantic saturation:** the number of documents in a semantic retrieval system reaches a point where several documents have semantically similar search entries associated with them;
- **pragmatic saturation:** provided that a semantic retrieval system is configured so as to learn from user inputs (as documents are being retrieved), an increasing number of queries supplied to the system can result in a web of associations so complex that it reduces the availability of new semantic labels. We refer to this kind of saturation as pragmatic because it relates to the way in which search queries are entered into the system.

The three-fold problem of syntactic, semantic and pragmatic saturation can be characterized – from a simplified perspective – as problems having to do with "*too many search entries*" (i.e., syntactic saturation), "*too many documents*" (i.e., semantic saturation), and "*overfitting from too many user interactions*" (i.e., pragmatic saturation), respectively.

IV. THE DOING-WHEN-SEEING PARADIGM

We propose the ***Doing-When-Seeing (DWS) paradigm*** as a general set of principles for the design of content management related interactions in Digital Reality – including 3D virtual – spaces.

The paradigm consists of the following types of operations, which can be carried out in a loop, always executed in the order $1 \rightarrow 2 \rightarrow 3$ or $1 \rightarrow 3$:

1) A set of *selection operations*, in which the user performs an interaction that can serve as a uniquely identifiable '*target*' during subsequent operations. Examples could include the selection of an element or group of elements on an interface (depending on the application domain), or the submission of a keyword or search term in a text box; in all of these cases, the element(s) or the text entered can serve as a uniquely identifiable 'target';
2) A set of *retrieval operations*, in which the user 'retrieves' the previously selected 'target', thereby effecting an operation in connection to it. Examples could include duplicating the previously selected element, or conceptually linking a previously entered text with a new element.
3) A set of *update operations*, in which the user can 'modify' a previously selected 'target'.

The name 'Doing-When-Seeing' arises from the logic of these operations: one can only select as a target an entity that one sees; and one can only perform retrieval or update operations whenever one has seen and selected the target entity.

At face value, DWS can simplify many operations from a cognitive perspective because it removes the need to effect

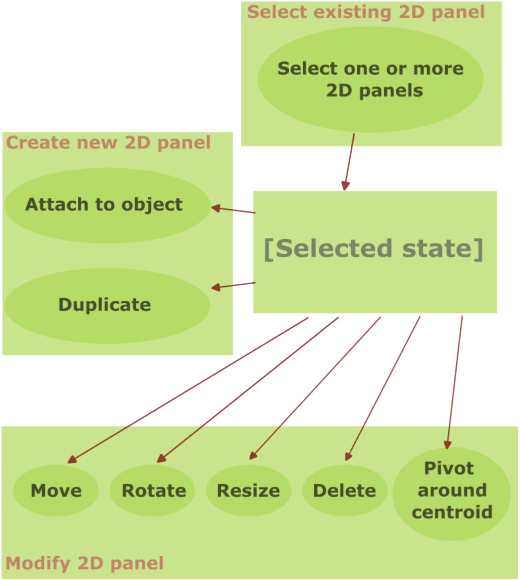


Fig. 2. Doing-When-Seeing operations proposed in [16] for creating and editing 2D display panels.

changes that are not based on something that is already visible or available to the user (here, visibility can be interpreted more broadly as being in the purview of one's cognition).

In the remainder of the paper, we demonstrate the applicability and viability of DWS in the use cases outlined earlier in the introduction.

V. EXAMPLE APPLICATIONS OF DWS: CREATING 2D DISPLAY PANELS

A. *Use of DWS to create 2D display panels*

Using the Doing-When-Seeing paradigm, in [16] we proposed a novel method that includes a set of operators categorized as selection operators, addition operators, placement operators, manipulation operators, and persistence operators for the editing of 2D display panels and 2D content layouts in 3D virtual spaces. The specific states and commands within the workflow are shown in Figure 2.

Prior to accessing addition operators, the user needs to choose a location and orientation. Since specifying orientations can be challenging, the user can simply point the 3D cursor at a surface within the space. Next, the user can issue the 'Attach to object' command to create a new 2D display panel.

An alternative method to 2D display panel creation is to select one or more existing panels, and to duplicate them ('Duplicate' command). In this case, the position, orientation and size of the existing panel will help determine the respective parameters of the new panel (while orientation and size can be kept, generally it is a good idea to place the new panel directly beside the existing one).

Additionally, a 2D display panel that has just been selected, or just been created can be further manipulated via the placement and manipulation operators. These include the 'Move', 'Rotate', 'Resize', 'Delete' and 'Pivot around centroid' commands – the latter enabling a group of panels to be rotated around a single point such that their relative positions and orientations do not change.

We note that this interface design respects the principles of the Doing-When-Seeing paradigm in that new 2D display panels can only ever be created based on an existing panel ('Duplicate') or an existing object and its surface normal ('Attach to object'). Since a 2D panel can never be created in mid-air, the camera-object independence dilemma is circumvented, as the need for panel positions to be radically altered is obviated.

B. *Experimental validation*

Through usability tests, we validated the proposed method within 3 different VR spaces inside the MaxWhere VR Platform. MaxWhere VR is an extensible desktop VR platform, built over the OGRE graphical engine, that provides a dynamic document model (known as the Where Object Model, or WOM), together with a Javascript based API for creating interactive 3D spaces. MaxWhere also provides a concept of 2D display panels, referred to as smartboards, which can hold PDF files, audio-visual files, or any other web-based content that could normally be rendered inside a Chrome browser.

Based on our experiments, we showed that the proposed method and the operators it includes were all found to be useful by users, and at the same time were sufficient for the re-creation of existing 2D content layouts. We reported on this experiment in detail in [16].

In the experiment, 10 test subjects were given the task of re-creating existing 2D content layouts in 3 different virtual reality spaces based on screenshot images of the original layouts. Following the experiments, we validated the speed and precision with which the layouts could be re-created.

Results showed that test subjects were able to re-create the layouts at a rate of less than 45 seconds per 2D display panel, and excepting outliers, at a root mean squared accuracy of less than 20 cm in terms of 2D display panel position, as well as less than 0.01 radians in terms of 2D display panel orientation.

The question of whether 45 seconds spent on the creation of each individual 2D display panel is a long or short time can be argued from many different perspectives. Nevertheless, in post-experiment interviews, subjects found the editing method to be intuitive and easy to use. Additionally, considering other digital operations relevant to 2D content integration and management, such as selection of appropriate documents, selection of a location in which to place the documents or navigation in the 3D space, in empirical terms 45 seconds is not an inordinately long period of time.

VI. EXAMPLE APPLICATIONS OF DWS: CREATING 2D CONTENT LAYOUTS

A. *The concept of the 3D File System*

Whereas in the previous section, we outlined a set of operations in which individual 2D display panels could be created and duplicated, we further extended the application of the DWS paradigm to the duplication of existing 2D content layouts as a higher-level entity [17]. This approach, if shown to be practical, would be an example of higher-order structural operations.

The operations outlined in the previous section do in fact include the selection of multiple 2D display panels and their duplication as a unit. However, in [17], a further step was taken in that existing groups of 2D layouts were associated with a unique project name, along with other semantic attributes such as topic, date and location at which the project was created. Such metadata attributes can potentially help trigger episodic memories, helping users to recall the project that should be loaded for duplication. We labeled the proposed concept as a "3-Dimensional File System", and called our proposed implementation the "MaxWhere File System v0.1", based on our reference implementation on the MaxWhere Platform [17].

B. Experimental validation

Validating the viability of the proposed approach to duplicating 2D content layouts again raises some questions of epistemology: after all, the question of whether it is faster to duplicate (and then modify) an existing 2D content layout than to create a new one based on individual 2D display panels and their duplicate versions will depend on: a.) how many 2D display panels are included in the layout, b.) how easy is it to recall and retrieve the content layout – among many other candidates – for duplication.

Clearly, no hard and fast rules can be formulated and there is a tradeoff between the two extremes of duplicating complete layouts and building new ones from scratch. However, it is also possible to consider whether there are reasonable cases where layout duplication can be faster. If the answer is yes, this will provide an existence proof for the potential superiority of higher-order structural operations.

To this end, we proposed a benchmark protocol for the standardized evaluation of layout creation methods. We showed that while the benchmark protocol encapsulates a trade-off between the time required to position individual display panels (favoring layout duplication) and the time required to find the appropriate layout to duplicate (favoring manual creation of individual display panels), it can be used to derive an existence proof for increased effectiveness of the proposed 3D file system method in that it provides a set of viable use-case scenarios for 2D content layout creation.

Using the proposed benchmark protocol, we gave an existence proof for the potential superiority of the proposed project duplication method, by showing that users were able to create new 2D content layouts based on existing layouts significantly faster than when having to manually create each individual 2D display panel. In the experiment, 3 conditions were compared:

- Scenario A ("no retrieval control group"): test subjects were asked to create all content layouts anew
- Scenario B ("retrieval group"): in the case of some content layouts, duplication of existing content sets was allowed, but users were not told which content sets to duplicate, only that they had to duplicate an already existing layout
- Scenario C ("deep retrieval group"): similar to scenario B, with the difference that test subjects could study the content and layout of each project for 2 minutes after having created it

Table I shows the comparison between Scenarios A and B; while Table II shows the comparison between Scenarios A and C. Prior to the analysis, the normality of the data was confirmed using the Shapiro-Wilk test. Based on the comparisons, then, an independent samples T-test revealed that the subjects from the retrieval group achieved significantly lower creation times per 2D display panel (N = 80, M = 72.725, SD = 15.290) than subjects from the control group (N = 80, M=96.058, SD= 10.518), $(t(140) = 11.245, p < 0.001)$; similarly, the subjects from the deep retrieval group achieved significantly lower creation times per 2D display panel (N = 64, M = 74.708, SD = 10.392) than subjects from the control group (N = 80, M = 96.058, SD = 10.518), $(t(142) = 12.168, p < 0.001)$; In both cases, the effect size was large (Cohen's $d = 1.778$ and 2.041, respectively).

A third interesting comparison would have been the difference between the retrieval group and the deep retrieval group. However, contrary to our expectations, in this case, the average time taken by subjects within the deep retrieval group was actually somewhat greater than the time taken by subjects within the retrieval group (though not to a significant extent). It is possible that further investigations in an ecologically more valid scenario (e.g., by having subjects work with the content sets over a longer period of time) could uncover an advantage to subjects being acquainted with the layouts they need to retrieve for duplication.

VII. EXAMPLE APPLICATIONS OF DWS: RETRIEVAL OF 2D CONTENT IN 3D VIRTUAL ENVIRONEMNTS

With the increasing volume of data surrounding users, as described in the introduction, effective content retrieval is becoming an ever-more pressing issue on many computing platforms. In a research article by Ames and colleagues [18], which predates the recent upsurge in VR and AI technologies, it was observed that merely the swift expansion of storage capacity and the growing accessibility of cross-platform multimedia content have contributed to a significant increase in the quantity and diversity of digital materials gathered by users. The authors emphasize that while Internet search capacities have greatly outperformed local search on desktop systems, the conventional file-folder hierarchical arrangement of files is ill-equipped to tackle the problem in a broader sense.

A. The Graph-Indexed Tensor Structure (GITS) model for semantic retrieval

In [15], we proposed a graph-tensor representation based document retrieval model, referred to as the Graph-Indexed Tensor Structure (GITS), that is pragmatic in the sense that it does not make any assumptions with respect to semantic relationships defined a priori; instead, associations are created as the model is used.

The general architecture of GITS can be seen in Figure 3. The core of the model is a tensor, the dimensions of which can represent any kind of semantic dimension, such as time, location or topic. The documents, in turn, are stored within

TABLE I

Comparison of average time taken to create a 2D display panel for the control group and retrieval group. The independent samples t-test – using Welch correction due to inequality of variances – shows that the difference between the two groups is significant.

	Test	Statistic	df	p	Mean Difference	SE Difference	Cohen's d	SE Cohen's d
Seconds/SB	Student	11.245	158.000	$< .001$	23.333	2.075	1.778	0.212
	Welch	11.245	140.091	$< .001$	23.333	2.075	1.778	0.212

Assumptions check: Test of equality of variances (Levene's)

	F	df_1	df_2	p
Seconds/panel	29.994	1	158	$< .001$

TABLE II

Comparison of average time taken to create a 2D display panel for the retrieval group and deep retrieval group. The independent samples t-test shows that the difference between the two groups is significant (in this case, no Welch correction is required).

	t	df	p	Mean Difference	SE Difference	Cohen's d	SE Cohen's d
Seconds/panel	12.168	142	$< .001$	21.350	1.755	2.041	0.233

Assumptions check: Test of equality of variances (Levene's)

	F	df_1	df_2	p
Seconds/panel	0.214	1	142	0.644

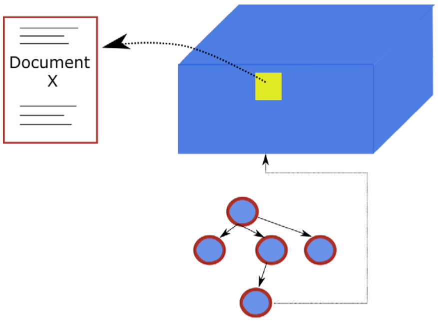


Fig. 3. Schematic view of the graph-indexed-tensor concept. While the directed graphs which are used to index the tensor in each dimension could have a variety of structures, this diagram shows directed acyclic graph, similar to a radix trie.

the tensor (i.e., can be indexed via a specific coordinate in each of the dimensions of the tensor). However, to ensure that multiple values in a dimension could potentially be used to index the same document, and to be able to model richer forms of associativity than merely sequential relationships afforded by the proximity of coordinates within a tensor, the GITS model also includes an index graph corresponding to each of the dimensions of the tensor.

Depending on the type of graph that is used, several variations of GITS can be conceived of, such as:

- the keyphrase-based index graph, which uses text-based keywords or keyphrases and employs a radix trie to store them, such that leaves of the trie point to different coordinates within the corresponding dimension;
- the hierarchical index graph, which can be used to model hierarchical concepts such as location or time (e.g., dates within the same week, or at around the same time of the year, or on the same day of the week would then be 'closer' to each other than other, more general pairs of dates)
- the associative keyphrase-based index graph, which is similar to the keyphrase-based index graph, but also can include associations between leaves of the radix trie; such a model can be used to link together the key phrases, i.e. a keyphrase leading to too many search results can subsequently be linked to the next keyphrase entered which returns useful results.

The associative keyphrase-based variant of GITS conforms especially well to the key ideas behind the DWS paradigm, since all search actions by users – besides serving their primary purpose – also specify a '*target keyphrase*'. In case the target keyphrase is not immediately useful, it can still be subsequently linked to keyphrases that are useful, but did not immediately come to the user's mind.

B. Experimental validation of the associative keyword-based GITS model

To verify the extent to which associativity can be useful within the GITS model, we conducted an experiment comparing the keyphrase-based variant with two different ways of using the associative keyphrase-based variant.

1) Preliminaries: Prior to the experiment, multiple screenshots were prepared, capturing different views of the content inside a 3D virtual reality space, such that in each screenshot, a single 2D display panel was highlighted as showing the target content. The content was related to 3 distinct, but related topics: Sports, Football and Hungarian Football.

A '*baseline GITS model*' was also prepared containing 3 keyphrases for each 2D display panel which described the content displayed in the panels using everyday terms. In the case of this experiment, the GITS model had only 1 index dimension – therefore, the tensor component could be conceived of as a vector, such that it is indexed by some kind of an index trie (associative or not).

Each of the keyphrases were entered into the model in different variations, such that each variation started with a different word within the keyphrase. For example, for the keyphrase "Pele holds a ball", the keyphrases "Pele holds a ball", "holds a ball", "a ball" and "ball" were all added, so that users could potentially start with any word within the keyphrase when searching, and would not be limited by the grammatical structure of the keyphrase.

2) Experimental design: Subjects were then tasked with typing in a keyword corresponding to the highlighted panel. Careful selection of screenshots ensured that the content of the targets was clearly visible, and some repetition was incorporated to assess the subjects' ability to recall previously entered phrases. All subjects were shown the same content in the same order, however, each of the subjects belonged to 1 of 3 groups:

- Scenario A ("*Simple search method*"): subjects used the basic keyphrase-based index trie model without associativity to search for the appropriate 2D display panels within the baseline GITS model.
- Scenario B ("*Personal search method*"): subjects performed the same task on their own personal copy of the baseline GITS model, such that the model was also adaptive to their search queries.
- Scenario C ("*Collective search method*"): subjects perfomred the same task on a collective copy an adaptive GITS model, which was initially a cloned version of the baseline GITS model, but evolved to eventually include all of the unsuccessful keyphrases entered by the test subjects. In this scenario, we hypothesized that the task of subjects using the model at a later stage in the experiment would become easier.

In the case of each individual search task, we recorded the number of times the user entered a keyphrase before the search returned 3 or less search results. In case the search returned more than 3 search results, nothing was displayed to the subjects and they were asked to try again. In each case, subjects could try again at most 4 times (after the 5th keyphrase, subjects were asked to proceed to the next display panel).

3) Results: A total of six test subjects participated in the experiments (2 subjects per scenario).

Due to the deviation of the results from the normal distribution, instead of conducting a T-test, we opted to perform the Wilcoxon Signed-Rank Test, which is a non-parametric analogue of the T-test. Results of the test are shown in Table III. The results indicate that in a statistically significant sense, users in Scenario A required more trials compared to users in Scenario B, and users in Scenario B required more trials compared to users in Scenario C to search for the same content.

The effect sizes, based on the rank-biserial correlations, were not very large. However, this could be due to the relatively small number of subjects and small number of trials. Nevertheless, the results go to show that associativity within the GITS model can help reduce the time taken to search for the same documents.

C. Addressing the challenges of syntactic, semantic and pragmatic saturation

1) Quantifying syntactic saturation: To automatically determine the degree to which *syntactic saturation* can be a problem in a specific GITS model, we have proposed a Shannon entropy-based detection method in [15]. According to this method, the Shannon-entropy of a node at any given layer (i.e., corresponding to any given prefix) in the index graph can be computed based on the information contained in the next character that is typed with respect to the resulting content set:

$$E(n_i) = -\sum_{o=l}^{O} \frac{\sum_{c} I_{c\to o} f_{c\to o}}{C} \log_2 \frac{\sum_{c} I_{c\to o} f_{c\to o}}{C}. \qquad (1)$$

where O is the number of possible outcomes (i.e. number of potentially different search results), C is the number of possible characters that can be entered (e.g. all alphanumeric characters), $I_{c\to o}$ is an indicator function whose value is 1 if typing character c as the next character may potentially yield the search result o, and $f_{c\to o}$ is the fraction of subsequent paths in the keyphrase-based index graph that begin with character c and eventually terminate in the search result o.

By calculating the average Shannon-entropy per node in a given layer, as described above, an entropy value can be associated with each layer in the index graph. This value, as well as if and by how much it is reduced from one layer to the next, can provide a comparative indication as to how syntactically saturated an index graph is.

2) Quantifying semantic saturation: Semantic saturation arises when the documents in a GITS model are too numerous per category, i.e. many documents with a similar semantics exist. In such cases, using keyphrases that describe the documents at a different semantic level may be a viable approach (for example, if a document store has 50 videos on birds, the keyword 'bird' may be less useful than a more specific keyword, like 'blue jay' or 'cardinal').

To quantify the degree of semantic saturation in a document store, a viable solution is to log the number of times the user searches unsuccessfully for a document (we call this the *search length*), and the number of documents each search operation returns (we call this the *multiplicity*). A candidate metric, then, which we refer to as the *semantic saturation index*, can be computed based on the difficulty of selecting a document from a multiplicity of M for any given search iteration i:

TABLE III

WILCOXON SIGNED-RANK TEST. FOR ALL TESTS, THE ALTERNATIVE HYPOTHESIS SPECIFIES THAT MEASURE 1 IS GREATER THAN MEASURE 2. FOR EXAMPLE, SCENARIO A IS GREATER THAN SCENARIO B.

Measure 1		Measure 2	W	z	p	Hodges-Lehmann Estimate	Rank-Biserial Correlation	SE Rank-Biserial Correlation
Scenario A	–	Scenario B	1577.500	3.015	0.001	1.000	0.427	0.141
Scenario B	–	Scenario C	953.000	4.507	$< .001$	1.500	0.763	0.168

Assumptions check: Test of normality (Shapiro-Wilk): significant results suggest a deviation from normality

Measure 1		Measure 2	W	p
Scenario A	–	Scenario B	0.772	$< .001$
Scenario B	–	Scenario C	0.661	$< .001$

$$D_i = I_{M_i>0}(1 - \frac{1}{M_i}) + I_{M_i=0} \tag{2}$$

where i is the index of the search iteration, M_i is the multiplicity of documents returned for that iteration, and $I_{M_i>0}$ and $I_{M_i=0}$ are indicator functions the return 1 when the multiplicity is greater than 0, or equal to zero, respectively (and 0 otherwise). In this case, D_i is the difficulty of selecting a document in the given iteration, which becomes increasingly closer to 1 as the multiplicity increases (and is 1 if the multiplicity is 0).

For an entire search length, then, comprising I iterations, the average difficulty can be taken as a basis for semantic saturation:

$$SemSat = \frac{1}{I}\sum_{i=1}^{I}\left(I_{M_i>0}(1 - \frac{1}{M_i}) + I_{M_i=0}\right) \tag{3}$$

For example, for a search length of 2, with 4 documents and 2 documents returned, respectively, the semantic saturation value would be 0.625. For a search length of 3, with 8 documents, 6 documents and 3 documents returned, the semantic saturation value would be 0.78. On the other end of the spectrum, for a search length of 2, with 2 documents and 1 document returned, respectively, the semantic saturation value would be 0.25.

3) Quantifying pragmatic saturation: Pragmatics has to do with the relationship between a language and its speakers, or users. The GITS model is to a large degree pragmatic, in that it does not (necessarily) make use of any existing notions of semantic similarity, but instead builds associations based on usage patterns from the past. Pragmatic saturation arises when an associative keyphrase based index tree begins to hold associations that cause conflicts in some sense; when the root of the association (i.e. the search phrase that is typed instead of another) closely resembles one or more other search phrases, either in a syntactic or semantic sense. In either case, it may be difficult for users to separate the search terms at a cognitive, conceptual level.

Based on this description, it is clear that the assessment of pragmatic saturation is closely related to, and can be reformulated in terms of the assessment of semantic and syntactic saturation.

VIII. CONCLUSIONS

Rapid advances within the fields of 3D spatial technologies (VR, AR, MR), artificial intelligence and others have led to the emergence of new synergies, often characterized as digital realities. Key aspects of digital realities include, among others, users being less and less tethered to any specific physical location (via mobile platforms); users wanting to work with an increasingly complex web of contextual information; and users increasingly wanting to work within spatial computing environments based on spatial interaction metaphors. Trends such as these have led to the challenge of organizing heterogeneous digital content sets and making them amenable to intuitive exploration.

In this paper, we introduced a paradigm which we refer to as the ***"Doing-When-Seeing (DWS) paradigm"***. DWS is a design philosophy and methodology that can help address the challenges of 3D content layout creation, 3D content set organization, as well as content retrieval in 3-dimensional digital realities. DWS relies on the basic steps of selection operations, retrieval operations and update operations. A key feature of the approach is that retrieval and update operations can only be applied entities that have first been selected; therefore, it obviates the need to create new entities outside of an existing context.

To demonstrate the viability of the DWS paradigm, we gave several examples in the paper, in which DWS was used to design a 2D display editor interface, a 2D content layout creation interface, as well as an adaptive semantic digital document store and retrieval model. In each case, we verified the usefulness of the proposed solution through usability experiments and statistical evaluations. In the latter case of the digital document store and retrieval model, we also proposed a set of analytic methods to evaluate different kinds of saturation issues that can arise when the store contains too many documents, too many documents that are similar, or too many semantic search entries that are in some sense similar.

Based on the examples, we showed that the Doing-When-Seeing paradigm in general, and the proposed solutions in particular can be useful in designing new digital reality interfaces.

REFERENCES

[1] W. Arber, “The impact of science and technology on the civilization,” *Biotechnology advances*, vol. 27, no. 6, pp. 940–944, 2009. DOI: 10.1016/j.biotechadv.2009.05.014.

[2] L. Mumford, *Technics and civilization*. University of Chicago Press, 2010. ISBN: 978-0-226-55027-5.

[3] M. Mauss, *Techniques, technology and civilization*. Berghahn Books, 2006. ISBN: 978-1-57181-662-7.

[4] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, “Ai and ml accelerator survey and trends,” in *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–10, IEEE, 2022. DOI: 10.1109/HPEC55821.2022.9926331.

[5] S. Paek and N. Kim, “Analysis of worldwide research trends on the impact of artificial intelligence in education,” *Sustainability*, vol. 13, no. 14, p. 7941, 2021. DOI: 10.3390/su13147941.

[6] I. Horvath, Á. B. Csapó, B. Berki, A. Sudar, and P. Baranyi, “Definition, background and research perspectives behind ‘cognitive aspects of virtual reality’(cvr),” *Infocommunications Journal: A Publication of the Scientific Association for Infocommunications (HTE)*, no. SP, pp. 9–14, 2023. DOI: 10.36244/ICJ.2023.SI-IODCR.2.

[7] M. Chui, J. Manyika, M. Miremadi, N. Henke, R. Chung, P. Nel, and S. Malhotra, “Notes from the AI frontier: Insights from hundreds of use cases,” *McKinsey Global Institute*, vol. 2, 2018.

[8] G. Pfeil, R. Holcomb, C. T. Muir, and S. Taj, “Visteon’s sterling plant uses simulation-based decision support in training, operations, and planning,” *Interfaces*, vol. 30, no. 1, pp. 115–133, 2000. DOI: 10.1287/inte.30.1.115.11620.

[9] E. L. Legge, C. R. Madan, E. T. Ng, and J. B. Caplan, “Building a memory palace in minutes: Equivalent memory performance using virtual versus conventional environments with the method of loci,” *Acta psychologica*, vol. 141, no. 3, pp. 380–390, 2012. DOI: 10.1016/j.actpsy.2012.09.002.

[10] B. Berki, “Desktop VR and the Use of Supplementary Visual Information,” in *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 000333–000336, IEEE, 2018. DOI: 10.1109/CogInfoCom.2018.8639925.

[11] P. R. Daugherty and H. J. Wilson, *Human + Machine: Reimagining Work in the Age of AI*. Harvard Business Press, 2018. ISBN: 978-1-633-69386-9.

[12] P. Baranyi, A. Csapo, T. Budai, and G. Wersényi, “Introducing the Concept of Internet of Digital Reality – Part I,” *Acta Polytechnica Hungarica*, vol. 18, no. 7, pp. 225–240, 2021. DOI: 10.12700/APH.18.7.2021.7.12.

[13] G. Wersényi, A. Csapó, T. Budai, and P. Baranyi, “Internet of Digital Reality: Infrastructural Background – Part II,” *Acta Polytechnica Hungarica*, vol. 18, no. 8, pp. 91–104, 2021. DOI: 10.12700/APH.18.8.2021.8.5.

[14] Á. B. Csapó, I. Horvath, P. Galambos, and P. Baranyi, “VR as a Medium of Communication: From Memory Palaces to Comprehensive Memory Management,” in *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 000389–000394, IEEE, 2018. DOI: 10.1109/CogInfoCom.2018.8639896.

[15] T. Setti and Á. B. Csapó, “GITS: A Graph-Indexed-Tensor Structure for the Adaptive Associative Semantic Tagging of Digital Documents,” in *2023 IEEE Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 1–6, IEEE, 2023 (accepted, in press). DOI: 10.1109/SACI58269.2023.10158658.

[16] T. Setti and A. B. Csapo, “A canonical set of operations for editing dashboard layouts in virtual reality,” *Frontiers in Computer Science*, vol. 3, p. 659600, 2021. DOI: 10.3389/fcomp.2021.659600.

[17] T. Setti and Á. B. Csapó,“Quantifying the effectiveness of project-based editing operations in virtual reality,” in *2022 1st IEEE International Conference on Cognitive Aspects of Virtual Reality (CVR)*, pp. 000049–000054, IEEE, 2022. DOI: 10.1109/CVR55417.2022.9967642.

[18] S. Ames, N. Bobb, K. M. Greenan, O. S. Hofmann, M. W. Storer, C. Maltzahn, E. L. Miller, and S. A. Brandt, “LiFS: An Attribute-Rich File System for Storage Class Memories,” in *Proceedings of the 23rd IEEE/14th NASA Goddard Conference on Mass Storage Systems and Technologies*, IEEE, 2006. DOI: not available.

Tarek Setti This author received his MSc degree in 2019 from the University of Pécs. In 2019 he started his PhD studiesat Széchenyi István University in Győr, Hungary where he is currently finishing his PhD research.

His main research focus is on the design of virtual reality spaces well adapted to human cognitive capabilities, enabling more seamless information management and retrieval in 3D digital environments.

Mr. Setti has published 7 papers in international conferences and journals.

Ádám B. Csapó This author obtained his PhD degree at the Budapest University of Technology and Economics in 2014.

Since 2016, he has worked as an Associate Professor at the Széchenyi István University, and since 2022, at Óbuda University in Budapest, Hungary.

Dr. Csapo’s research focuses on soft computing tools for developing cognitive infocommunication channels in virtual collaboration environments, with the goal of enabling users to communicate with each other and with their spatial surroundings in novel and effective ways. He has also participated in the development of a commercial desktop VR platform serving both educational and industrial use cases. Dr. Csapo has over 50 publications, including 1 co-authored book and 16 journal papers. His work has received over 2,000 independent citations in the scientific literature.