The effectiveness of lightweight automated support for learning about dynamic systems with qualitative representations

Kragten, M.; Bredeweg, B.

DOI
10.1145/3605098.3635950

Publication date
2024

Document Version
Final published version

Published in
The 39th Annual ACM Symposium on Applied Computing

License
CC BY

Citation for published version (APA):
Effectiveness of lightweight automated support for learning about dynamic systems with qualitative representations

Marco Kragten
Faculty of Education
Amsterdam University of Applied Sciences
m.kragten@hva.nl

Bert Bredeweg
Faculty of Science
University of Amsterdam
b.bredeweg@uva.nl

ABSTRACT
We developed an application which allows learners to construct qualitative representations of dynamic systems to aid them in learning subject content knowledge and system thinking skills simultaneously. Within this application, we implemented a lightweight support function which automatically generates help from a norm-representation to aid learners as they construct these qualitative representations. This support can be expected to improve learning. Using this function it is not necessary to define in advance possible errors that learners may make and the subsequent feedback. Also, no data from (previous) learners is required. Such a lightweight support function is ideal for situations where lessons are designed for a wide variety of topics for small groups of learners. Here, we report on the use and impact of this support function in two lessons: Star Formation and Neolithic Age. A total of 63 ninth-grade learners from secondary school participated. The study used a pre-test/intervention/post-test design with two conditions (no support vs. support) for both lessons. Learners with access to the support create better representations, learn more subject content knowledge, and improve their system thinking skills. Learners use the support throughout the lessons, more often than they would use support from the teacher. We also found no evidence for misuse, i.e., ’gaming the system’, of the support function.

CCS CONCEPTS
• Applied computing → Education → Interactive learning environments

KEYWORDS
Automated support, intelligent tutoring systems, qualitative representations, system thinking.

ACM Reference format:
M. Kragten and B. Bredeweg. 2024. In Proceedings of ACM SAC Conference, Avila, Spain, April 8 – April 12, 2024 (SAC’24), 10 pages. DOI: 10.1145/3605098.3635950

1 INTRODUCTION
Being able to understand and reason about dynamic systems (climate change, stock markets, epidemics, etc.) is important in modern society and such skills should be learned at school [1,2]. Modeling is commonly used approach to enable learners to understand dynamic systems. A typical approach is modeling dynamic systems on the computer using a formal language [3,4,5,6]. This modeling aids learners in the process of creating, refining, and explicitly articulating their comprehension of the given phenomenon [3,7]. However, research shows that learners find modeling difficult and that they need support [3,8,9,10,11,12].

Our goal is to address this challenge using educational software that empowers learners to construct qualitative representations with ample support. Qualitative representations are non-numerical descriptions of (physical) systems. The accompanying algorithms allow computer programs to reason about the behavior of systems, without precise quantitative information, but using qualitative descriptions of causal relations (e.g., proportional positive), values (e.g., zero, positive) and changes (e.g., decreasing, constant, increasing) of quantities [13,14]. Constructing qualitative representations aligns closely with the way humans naturally engage in causal reasoning when dealing with dynamic systems [14].

1.1 Related research
The focus of the present study is to investigate the impact of integrating a support system that can automatically assist learners in rectifying errors made during the process of constructing a qualitative representation. Notable examples of studies that have explored dynamic systems modeling applications that offers support throughout the construction process include Betty’s Brain [4], Co-Lab [6], and Dragoon [5].

Betty’s Brain uses the Teachable Agent approach [4]. Learners make a concept map ‘in the head’ of Betty, which helps Betty to understand the dynamics of a river ecosystem. This approach offers learners support on the quality of the concept map because Betty can be questioned about her understanding of the system and, if desired, can explain how she arrives at an answer. At certain times, based on the concept map, Betty must pass a predetermiend quiz. If Betty fails, then the concept map is not correct. In the experimental conditions, where learners had the opportunity to ask questions to Betty and/or access the quiz, they generated more accurate concept maps compared to those
in the control condition who did not have such access. In the initial version of the application, if the student couldn't fix their concept map before Betty's third quiz attempt, they received explicit hints. These hints guided learners on where to add missing concepts and links and, in certain cases, explained how to rectify a misdirected causal link in their concept map. The researchers found evidence supporting the idea that learners tended to ‘game the system’ [see 15 for a discussion on this phenomenon] by focusing on fixing these errors rather than truly learning the underlying concepts. They deduce that automated support for fixing errors can actually be detrimental to learning, as it can prevent learners from engaging in the critical thinking and problem-solving required to truly master the material. However, the learning gains were not experimentally tested with this version of the application. In the succeeding version of the application, when Betty doesn’t pass the test, learners are directed to study and reflect on relevant sections in the resources, instead of being provided with specific suggestions for modifying their concept map.

With the Dragoon application [5,16], learners can construct numerical models, which are mathematical representations of dynamic systems using differential equations and algebraic formulas. It provides a graphical interface where variables in a dynamic system are represented using stock-and-flow notation to support equation building. Dragoon has a support system to provide immediate feedback on errors by comparing the learner's model with a norm model. If multiple errors occur, the learner's input is then replaced with the corresponding entry from the norm model. To discourage ‘gaming the system’, the support system flags ingredients in the model that have been changed multiple times by a yellow perimeter. Additionally, the application provides students with guidance on their next steps. If they begin working on an ingredient prematurely, the support system issues a warning. The effect of the support system was examined in a post-test-only study, featuring two conditions: one with the support system and one without it [5]. Overall, there was no significant difference in performance between the two conditions. However, when analyzing the subset of learners who spent a high amount of time with the application in both conditions, those who received the automated support outperformed those who did not. Additionally, the researchers assessed the efficiency of both conditions within the subset of learners who devoted substantial time to the application. When measuring efficiency as the ratio of the post-test score to the time spent on Dragoon, the condition with the support system outperformed the condition without it. Hence, with regard to the impact of support on learning outcomes, the results in this study are somewhat inconclusive. No learning gains could be established due to the limitations of the post-test only design, and significant effects were only identifiable among a subset of learners. Furthermore, the study focused on evaluating students’ improvement in modeling skills related to dynamic systems, rather than assessing their acquisition of specific content knowledge of the systems they modeled.

Co-Lab shares a number of characteristics with Dragoon: (i) it is an application designed for learners to create numerical models using stock-and-flow notation, (ii) the support is generated by comparing the learner's model with a norm model, and (iii) it provides guidance on potential improvements and model expansion [6,17]. Learners have the option to request advice on the quality of their models. Co-Lab’s advice system sorts and selects advice based on various criteria. For example, teachers can customize the level of detail in the advice according to whether the learner’s model is in the initial, intermediate, or final phase. To prevent ‘gaming the system’, the availability of support can be configured. For example, learners may need to construct a minimum number of ingredients and/or there might be a minimum time interval between checks for advice. The researchers evaluated how learners utilized the support system and gathered feedback through a questionnaire regarding their experiences. The evaluation did not include measuring learning gains through an experimental setup.

1.2 Research questions

We conclude that research on the effectiveness of automated support for learning about dynamic systems by modelling is limited and that findings are inconclusive. Furthermore, most application, including those discussed in the preceding section, require the designer of a lesson activity to identify common learner errors and misconceptions, create test-items, and stipulate the timing and appropriateness of feedback. Other application use large amounts of previous user data to automatically generate support [18]. Neither solution suits our case, because one of our aims is to integrate the ‘learning by constructing qualitative representation’ approach for many lesson activities covering a wide range of topics within a variety of subjects. Teachers can either use previously developed lesson activities and adapt them for use in their classrooms or develop their own to suit their specific needs. As a result, there are many different lesson activities for different (often small) target groups of learners.

As a solution, we have devised an approach that autonomously generates support by exclusively leveraging a norm representation established by the teacher during the lesson design phase. An advantage of representation-based support is that it requires no further work once the lesson is designed. The support is automatically generated from the representation created during the design of the lesson, which now becomes the norm representation. The teacher does not need to define in advance the possible errors and how to react, nor define general rules that a correct solution must follow. Furthermore, no data from previous learners is needed. This makes it a promising approach for situations where many different lessons in various subjects using various topics are created and used for relatively small groups of learners. It also allows teachers to modify pre-existing lesson activities to fit the educational goals they want to achieve and still use the automated support.

The pedagogical approach for delivering automated support in our application also differs from those discussed in section 1.1. For instance, our support system refrains from providing the correct answer but instead uses cues and help in the form of hints (see section 2.2 for details). Additionally, learners are free
Lightweight support for learning with qualitative representations

in their access to the support function, both in terms of frequency and timing. We also do not provide any cues to discourage learners from using it.

To gain insight into the use and effectiveness of such an automated support, we seek to answer two questions. Research Question 1 (RQ1) is: ‘How do learners interact with the automated support?’ For instance, we provide insight into how often learners use specific parts of the support, during what parts of the lesson and for which ingredients of the representation. An inherent concern associated with automated support systems is the potential for learners to exploit the system, completing lessons with minimal effort [15]. In our study, we also examine whether learners exhibit such behavior.

The goal of implementing automated support is to assist learners in constructing qualitative representations and thereby improve their learning outcomes. In our lesson activities, we aim for two specific learning outcomes: (i) to foster an understanding of subject-specific content knowledge and (ii) to concurrently cultivate general systems thinking skills through the acquisition of a formal qualitative vocabulary. As such, Research Question 2 (RQ2) is: ‘What is the effect of the availability of automated support on learning outcomes?’ We expect that implementation of the support will facilitate learners in constructing their qualitative representations leading to a positive impact on their learning outcomes. This effect is facilitated by the ‘just in time’ nature of automated feedback. Learners no longer need to wait for the teacher to answer questions; instead, they receive help immediately, preventing them from getting stuck. This eliminates needlessly waisted time, and could prevent frustration, which might otherwise lead to demotivation or even giving up completely. Given the support in the form of cues and hints, learners are encouraged to reflect upon their mistakes, which in turn promotes the learning process. The anonymity of computer support can also be beneficial for some learners, who might not ask the teacher out of fear of feeling dumb [19].

2 DynaLearn Environment

The DynaLearn application consists of a canvas facilitating the construction of qualitative representations for learning about dynamic systems [13,20]. The application uses the Garp3 workbench which integrates various knowledge-based techniques from Artificial Intelligence [21]. The canvas is organized into a set of distinct levels with increasing complexity. For the work presented in this paper learners worked at level 3. At this level, cause-effect relationships can be represented to support reasoning about changes propagating through a system (Table 1 and Fig. 1).

Learners represent the entities (physical objects, e.g., Inhabitants) that make up the system and the configurations relating them (e.g., Live in). Connected to entities are quantities which characterize the entities (e.g., Temperature), and causal dependencies (– & +) between those quantities (e.g., Temperature positively influencing Crop production). Quantities have an associated direction of change (δ, which can be negative, zero, or positive) and possibly a quantity space, which specifies the possible values a quantity can take on. The latter allows learners to represent the idea that a system moves through different states (e.g., entity River valley with quantity Food supply and quantity space {Scarcity, Transition, Surplus}). A quantity space always has an alternation of intervals and point values.

Table 1: Ingredient types on level 3 of DynaLearn.

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>🎨</td>
<td>Physical object or an abstract idea.</td>
</tr>
<tr>
<td>External agent</td>
<td>🔄</td>
<td>Entity outside of the modelled system.</td>
</tr>
<tr>
<td>Quantity</td>
<td>🛡</td>
<td>Changeable feature of an entity.</td>
</tr>
<tr>
<td>Derivative</td>
<td>Δ</td>
<td>Direction of change of a quantity.</td>
</tr>
<tr>
<td>Quantity space</td>
<td>⭲</td>
<td>Characteristic states of a quantity</td>
</tr>
<tr>
<td>Correspondence</td>
<td>C</td>
<td>Relation between co-occurring values</td>
</tr>
<tr>
<td>Proportionality</td>
<td>+</td>
<td>Causal relationship in which a change is propagated.</td>
</tr>
<tr>
<td>Assign</td>
<td>🔄</td>
<td>Denotes initial values. *</td>
</tr>
<tr>
<td>Exogeneous influence</td>
<td>←→</td>
<td>Exerts an exogenous continues effect on a quantity.</td>
</tr>
</tbody>
</table>

*Ingredient types that represents starting value are not part of the support system discussed in this study. Other types of behavior (e.g., sinus) of the exogeneous influence are also available.

Additionally, it is possible to represent the idea of an external agent and exogenous quantity behaviour to distinguish the ‘system’ from the ‘external factors’ affecting it (e.g., Climate with a continuously increasing Temperature). The notion of correspondence (C) is used to specify co-occurring values (for instance, IF Food supply is Scarce, THEN Society type is Semi-sedentary). Learners can run a simulation at any point. This results in a state-graph (a sequence of states and transitions) which can be used to inspect the simulation results.

2.1 Qualitative Representation of Neolithic age

Fig. 1 shows a qualitative representation of the dynamic system during the Neolithic Age, one of the two topics of the lessons activities evaluated in this paper. The representation illustrates the transition from a hunter-gatherer society to an
agricultural society. Temperature rise due to climate change is the exogenous influence that drives the changes of the quantities within the representation. The climates' increasing temperature has a negative effect on the amount of water available in the mountains and forests.

Decreasing availability of water has a negative effect on the amount of wild plants and animals thereby decreasing the carrying capacity and the number of hunters and gatherers that can be supported. Decreasing carrying capacity of the mountain and forest areas has a positive effect on emigration of the hunters and gatherers. Increasing emigration of hunters and gatherers has a negative effect on their number. The climates’ increasing temperature has a positive effect on the crop production in the river valleys. Increasing crop production has a positive effect on food supply. At first, food supply is still scarce and the inhabitants of the river valley are bound to a semi-sedentary way of living. As temperature keeps increasing, food supply becomes surplus and the society type changes to agricultural. The availability of food and the change of society type are linked, hence, there is a correspondence between the quantity spaces. This restricts the qualitative representation to have three states (simulation is not shown here): (1) food is scarce and inhabitants live in a semi-sedentary way, (2) transition from food being scarce to surplus and society from being semi-sedentary to agricultural, and (3) food is surplus and society is agricultural. See [22] for a detailed description of how learners construct the qualitative representation during the lesson activity.

2.2 Implementation of support

Automated support was developed for the DynaLearn application. It compares a learner-constructed representation with the norm representation (created by the teacher as part of the design of the lesson). After each manipulation executed by the learner in the canvas a new mapping between these two representations is made using a Monte-Carlo-based heuristic approach. For each discrepancy found the support provides: cueing and help. Cueing means a small red circle is placed around each deviating representation ingredient and a red question mark appears on the right-hand side in the canvas (Fig. 2). When the learner clicks on the question mark, a message-box

appears providing help for each deviation in the form of a short hint, e.g., ‘Causal dependency: between wrong quantities?’ or ‘Quantity: assigned to wrong entity?’. Note that because the learner’s representation is compared to the best fitting subset of the norm representation, learners do not receive support on parts of the model they have not yet created.

Note that the idea of using a norm representation shares similarities with how support is generated in numerical modeling applications, as discussed in section 1.2. However, the technical implementation varies considerably due to the difference between logic-based qualitative representations and mathematical-based numerical models. For example, in numerical modeling, a support function should be able to check if the learner’s formula matches the norm model [16]. This can be challenging because multiple expressions might produce similar results, for example, $2(x+y)$ is equivalent to $2x+2y$. A comprehensive exploration of these differences falls beyond the purview of the current paper.

3 METHODOLOGY

3.1 Participants

A total of 63 ninth-grade learners from a Dutch secondary school participated in this study as a part of their regular courses. We divided the learners into two groups based on their regular class: the support condition ($n = 27$) and the no support condition ($n = 36$). The groups were randomly assigned to the conditions. All participants had previous experience with the application, having completed two lessons on different subject content at level 2 in the previous school year.

3.2 Design of the lesson and representation

Together with high school teachers two lessons were designed. The topics were Star Formation and Neolithic Age. Two qualitative representations were designed to describe the system dynamics of these topics. The representation of Neolithic Age is shown in Fig. 1 and described in section 2.1. The representation of Star Formation has two entities (Star and Core) connected by a configuration (has). There are five quantities (Mass, Gravity, Pressure, Temperature, and Nuclear fusion) connected by four positive causal dependencies describing a single causal chain (from Mass to Nuclear fusion). Mass and Nuclear fusion both have
Lightweight support for learning with qualitative representations a quantity space with three values (Low, Turning point, High). There is a correspondence between the quantity spaces representing the idea that stars with low (high) mass have low (high) fusion. Mass increases by an exogenous influence. The representations were used as the norm representation for support during the lessons (in the support condition). Next, a workbook was created that allowed learners to work independently during the lesson. The workbook contained information about the subjects Star Formation and Neolithic Age, questions to get learners thinking about the systems and scaffolds for constructing the representation in the application.

3.3 Research design

The study used a pre-test/intervention/post-test design with two conditions (no support or support) for both lessons. As part of their regular school schedule, a two-hour lesson was arranged for each condition. Learners were in the same condition (no support vs. support) for both lessons. Teachers were present to provide guidance. There was a three-month interval between the Star Formation and the Neolithic Age lessons.

3.4 Data collection

Data for this study consisted of tests and action logs of learner interaction with the support of the Dynalearn application.

3.4.1 Pre-test and post-test. The tests of both lessons consist of two parts: (i) the content knowledge test and (ii) the system thinking test.

For the Star Formation lesson we measured learners’ content knowledge of the causal relationships between quantities that are involved in the formation of a star by one open question: "Explain, step-by-step, the effect of a high stellar mass on the formation of a star" (Table 2). The open question was analyzed for the number of correct causal relationships and corresponding changes and values that learners correctly describe. The research group discussed ambiguous responses until a consensus was reached. The maximum score on this question was four.

For the Neolithic Age lesson a content knowledge test of seven items was developed. All items measure the extent to which learners were able to reason causally about the development of the agricultural society in the Neolithic age. The items were developed by one of the researchers (an expert in item design within the field). These were then presented to the teachers for feedback and adjustments were made where necessary. All items were closed questions, including multiple-choice and multi-ordering questions. A multi-ordering question tasks learners with arranging causal chain events in the correct sequence. For each correct answer on an item a student scored 1 point. Table 2 presents a short description of the items and the item-total correlations which were between .50 and .66.

In the second part of the test, learners’ system thinking skills were assessed by measuring their understanding of the qualitative vocabulary at level 3. This part of the test had six multiple-choice and -response items (see Table 2, item-total correlations between .26 and .61). The items were formulated in such a way that they could be answered without knowledge of particular subject content. For example, item 1 required learners to choose the representation that correctly describes the effect of ocean pollution on coral reef mortality, i.e., this system had two entities (ocean and coral reef), two quantities (pollution and mortality) and one causal relationship (pollution causes mortality). Item 1, 3, 5 and 6 were given a maximum score of 1 point. Item 2 had two subitems and the maximum score was 2 points. Item 4 had three subitems and the maximum score was 3 points. This led to a maximum system thinking score of 9 points.

Table 2: Test items and item-total correlations of the Star Formation and Neolithic Age lessons.

<table>
<thead>
<tr>
<th>Short item description</th>
<th>R_{it}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Star Formation</strong></td>
<td></td>
</tr>
<tr>
<td>1. Effect of high stellar mass on formation of a star (O)</td>
<td></td>
</tr>
<tr>
<td>Neolithic Age</td>
<td></td>
</tr>
<tr>
<td>1. Effect of temperature on hunter-gatherer population (MC)</td>
<td>.62</td>
</tr>
<tr>
<td>2. Effect of temperature on river valley population (MC)</td>
<td>.50</td>
</tr>
<tr>
<td>3. Effect of temperature on water availability in mountains and forests (MC)</td>
<td>.54</td>
</tr>
<tr>
<td>4. Effect of water availability on wild plants and animals (MC)</td>
<td>.66</td>
</tr>
<tr>
<td>5. Effect of plants and animals on carrying capacity (MC)</td>
<td>.56</td>
</tr>
<tr>
<td>6. Effect of carrying capacity on emigration of hunter-gatherer population (MO)</td>
<td>.50</td>
</tr>
<tr>
<td>System thinking</td>
<td></td>
</tr>
<tr>
<td>1. Qualitative representation of a system (MC)</td>
<td>.40</td>
</tr>
<tr>
<td>2. Propagation of effect in linear causal chain (MR) 2pt</td>
<td>.36</td>
</tr>
<tr>
<td>3. Propagation of effect in causal chain with multiple causes (MC)</td>
<td>.45</td>
</tr>
<tr>
<td>4. Correspondence of values of quantity spaces (MR) 3pt</td>
<td>.61</td>
</tr>
<tr>
<td>5. Effect of exogenous influence on states (MC)</td>
<td>.46</td>
</tr>
<tr>
<td>6. Values of a quantity space (MC)</td>
<td>.26</td>
</tr>
</tbody>
</table>

Note. MR = multiple-response, MC = multiple-choice, MO = multiple-ordering, O = open, R_{it} = item-total correlation.

3.4.2 Learner interaction. The final version of the learner-created representations were automatically evaluated by counting the number of correct ingredients. The maximum representation score of the Star Formation lesson was 23 and the maximum score of the Neolithic Age lesson was 40. Use of help was automatically registered together with a timestamp. Use of cueing was harder to detect, because it appeared automatically when an error was made, but may not always have been used (or even noticed) by the learner. In this study cueing was operationalized as follows: whenever a learner receives a cue for a given ingredient and changes it as their next action, this was interpreted as ‘making use of cueing’. These interactions were also automatically registered together with a timestamp. The one exception to this rule was when learners created a new ingredient. New ingredients must always be given a name. In that case, a cue automatically appears because the representation is evaluated before a name is assigned. Since a learner usually has a name in mind when creating an ingredient, they probably did not use cueing but were already planning to assign the name.
Therefore, when a name was given to an ingredient immediately after it was created, this was not interpreted as ‘making use of cueing’.

### 3.5 Data analysis

In the result section, we first present the descriptive statistics of the measures collected in this study. We then answer both research questions.

#### 3.5.1 RQ1: Learner interaction with the automated support

This interaction was analysed in several ways. Firstly, the way learners used the support is visualised using plots that show cueing and help throughout the lesson, separated by type of ingredient the support was used for. We also present and discuss descriptive statistics for the use of support per type of ingredient. Secondly, learners might exploit the support to complete the lesson with minimal effort by making more or less random changes to an ingredient until it is correct. Such behaviour was searched for by looking for sequences where learners repeatedly change one ingredient until it was correct without working on anything else.

#### 3.5.2 RQ2: Effect of automated support on learning outcomes

Two-way repeated measures ANOVAs were conducted with content knowledge scores and system thinking score as the dependent variable, condition (no support vs. support) as between-subjects factor and test (pre-test vs. post-test) as within-subject factor. The system thinking test was administered four times in a row, with two tests conducted during the Star Formation lesson (pre-test and post-test) and two during the Neolithic Age lesson. As a result, the within-subject factor test of this two-way repeated measures ANOVA had four levels. Tests for normality (Shapiro-Wilk), equality of variances (Levene), and sphericity (Mauchly’s W) affirm that the data satisfies the necessary assumptions for these analyses. The Mann-Whitney U test was employed to assess differences in the representation scores between the two conditions in the two lessons. This test was chosen due to the frequent occurrence of maximum representation scores in the support condition, and the non-normal distribution of these scores. The significance level was set at $p < .05$.

### 4 RESULTS

#### 4.1 Descriptive statistics

Table 3 shows the descriptive statistics of pre- and post-test scores and learner interaction (representation score, representations completed, cueing and help) with the representation.

<table>
<thead>
<tr>
<th></th>
<th>Star Formation</th>
<th>Neolithic Age</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No support</td>
<td>Support</td>
</tr>
<tr>
<td>Pre-test content knowledge</td>
<td>M (SD)</td>
<td>M (SD)</td>
</tr>
<tr>
<td>Post-test content knowledge</td>
<td>.58 (SD .94)</td>
<td>.63 (SD 1.04)</td>
</tr>
<tr>
<td>Pre-test system thinking</td>
<td>1.5 (SD 1.23)</td>
<td>2.52 (SD 1.42)</td>
</tr>
<tr>
<td>Post-test system thinking</td>
<td>4.25 (SD 1.84)</td>
<td>3.3 (SD 1.79)</td>
</tr>
<tr>
<td>Representation score</td>
<td>18.71 (SD 1.87)</td>
<td>22.33 (SD 1.59)</td>
</tr>
<tr>
<td>Representations completed</td>
<td>5 (SD 4.43)</td>
<td>22 (SD 2.02)</td>
</tr>
<tr>
<td>Cueing</td>
<td>24.41 (SD 16.96)</td>
<td>18.59 (SD 16.62)</td>
</tr>
<tr>
<td>Help</td>
<td>6.67 (SD 16.96)</td>
<td>12.44 (SD 16.62)</td>
</tr>
</tbody>
</table>

Note: * Total number of students that completed the representation.

#### 4.2 Learner interaction with automated support

Learners used the support for all types of ingredients in the representation (Table 4). In the Star Formation lesson, all learners used cueing. Six learners used it less than 10 times. These learners mostly used cueing for guidance with constructing quantity spaces and correspondence. Additionally, one student did not use help, while four learners used it more than 10 times. The maximum number of times help was used is 35.

Table 4: Descriptive statistics of the Star Formation and Neolithic Age lessons.

<table>
<thead>
<tr>
<th>Ingredient type</th>
<th>Star Formation</th>
<th>Neolithic Age</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cueing M (SD)</td>
<td>Help M (SD)</td>
</tr>
<tr>
<td>Entity (2/5)</td>
<td>2.3 (4.25)</td>
<td>.67 (1.47)</td>
</tr>
<tr>
<td>Configuration (1/4)</td>
<td>1.67 (1.49)</td>
<td>.33 (1.83)</td>
</tr>
<tr>
<td>Quantity (5/10)</td>
<td>2.41 (5.51)</td>
<td>.78 (2.28)</td>
</tr>
<tr>
<td>Quantity space (10/8)</td>
<td>15.00 (11.66)</td>
<td>4.19 (5.72)</td>
</tr>
<tr>
<td>Correspondence (1/4)</td>
<td>.52 (1.93)</td>
<td>.30 (1.10)</td>
</tr>
<tr>
<td>Proportionality (4/11)</td>
<td>2.52 (3.40)</td>
<td>.41 (.75)</td>
</tr>
</tbody>
</table>

* The total number of ingredients in the representation of Star Formation/Neolithic Age. ** Includes the values of the quantity spaces.
Learner interaction with the support was analyzed for ‘gaming the system’ by looking for sequences of repeatedly modifying an ingredient until it is correct. During the Star Formation lesson, students made a single modification to an ingredient on average 18.96 times ($SD = 11.95$). Making only one modification suggests that the students noticed an error, either via cueing or help, and then corrected the mistake. The number of times students made two sequential modifications on the same ingredient was relatively low, averaging .63 ($SD = .93$) times. One student did this four times during the lesson. Incidentally, students made three or more modifications on the same ingredients sequentially. The low numbers do not suggest any significant misuse of cueing or help. The Neolithic Age lesson shows similar results. Students made a single modification to an ingredient on average 13.11 times ($SD = 9.38$) and the frequency of multiple modifications was very low.

Finally, there are some differences in the ratio between the number of times ingredient types occur in the qualitative representations and the number of times learners use support for this ingredient. For instance, the number of responses to a cue for quantity spaces in the Star Formation lesson are high compared to how often they occur in the norm representation (average of 15.00 times on 10 ingredients of this type). This suggest that learners found it difficult to construct this ingredient. The use of cueing for proportionalities is relatively low (average of 3.89 times on 11 ingredients of this type) in the Neolithic Age lesson, suggesting that learners need less support with constructing the correct causal relationships.

### 4.3 Effect of availability of support on learning outcomes

Fig. 4 shows the learning outcomes of the Star Formation and Neolithic Age lessons.

#### 4.3.1 Content knowledge score of Star Formation

There is a significant interaction effect between condition (no support vs. support) and test ($F(1, 61) = 7.33, p < .01, \eta^2 = .03$) for content knowledge score of the lesson Star Formation (Fig. 4 upper panel). A significant simple main effect was found for the no support condition ($p < .001$) and the support condition ($p < .001$). This latter indicates that learners in both conditions significantly improved content knowledge score from pre-test to post-test. Pairwise comparisons with Holm-Bonferroni correction shows that content knowledge score is significantly higher at the post-test score for the support condition than for the no support condition ($p < .01$). This indicates that learners in the support condition learned more than learners in the no support condition.

#### 4.3.2 Content knowledge score of Neolithic Age

There is a significant interaction effect between condition and test ($F(1, 61) = 12.87, p < .001, \eta^2 = .10$) for content knowledge score of the lesson Neolithic Age (Fig. 4 second panel). A significant simple main effect was found the support condition ($p < .001$). However, no significant simple main effect was found for the no support condition.
Pairwise comparisons (Holm-Bonferroni) shows that content knowledge score is significantly higher at the post-test score for the support condition than for no support condition ($p < .001$). This indicates that learners in the support condition increased their content knowledge score and learners in the no support condition did not.

4.3.3 **System thinking score.** There is a significant interaction effect between condition and test ($F(3, 61) = 7.33, p < .001, \eta^2 = .04$) for system thinking score (Fig. 4 third panel). A significant simple main effect was found for no support ($p < .001$) and support ($p < .001$). Post-hoc analyses (Holm-Bonferroni) shows that there is a significant difference between the pre-test and post-test score of the Star Formation lesson for the no support condition ($p < .001$) and the support condition ($p < .001$). Thus, learners in both conditions increase their system thinking score. Under the no support condition, there is a significant decrease ($p < .05$) in the system thinking score of learners between the pre-test score of Star Formation and the pre-test score of the Neolithic Age. Learners show an improvement in their system thinking score ($p < .05$) under the support condition of the Neolithic Age lesson. The difference between pre-test and post-test system thinking score in the no support condition of this lesson is not significant ($p > .05$). The learners in the support condition exhibited a significant increase in their score (from Star Formation pre-test to Neolithic Age post-test) after completing the two lessons ($p < .001$). However, the post-test scores of the Neolithic Age lesson for both conditions do not show a significant difference ($p > .05$).

4.3.4 **Representation score.** Mann-Whitney U test was conducted to determine whether there is a difference in representation score between the no support and the support condition. The results (Fig. 4 lower panel) indicate a significant difference between conditions in the Star Formation lesson ($U = 133.50, p < .001, \text{rank } r = .71$) and the Neolithic Age lesson ($U = 53.00, p < .001, \text{rank } r = .86$). It can be inferred that the availability of support has a large impact on the extent to which learners construct the representation.

5 **CONCLUSION AND DISCUSSION**

We investigated the use and effect of lightweight automated support designed to aid learners in constructing qualitative representations. By constructing qualitative representations learners learn subject content knowledge about specific dynamic systems and develop general system thinking skills. The support consists of two parts: cueing (alerting the student to errors), and help (learners receive a hint on the type of error they made). The support is automatically generated from the representation, requiring no additional work from either the teacher or the designer of the lesson, and no data from previous users. The paper presents results from a study on the use and effect of this support on student learning outcomes in two lessons: Star Formation and Neolithic Age.
Lightweight support for learning with qualitative representations

In answer to RQ1, we found that learners make use of both support features throughout the lesson. On average, a learner has over 21 cueing and nine help interactions during a lesson, which is significantly more support than they would receive from a teacher during a similar lesson. This indicates that this lightweight support fulfills a role in the learning process that supplements the support provided by the teacher in a regular lesson. Learners that make use of the support generally keep doing so throughout the lesson, indicating that they view it as a useful feature. There is no evidence for ‘gaming the system’ which is often a risk with automated support [15]. An explanation for why the latter did not occur could be that the support function merely identifies the errors made and their types without providing the correct answers. Thus, gaming the system is discouraged, because trying all the options still constitutes a relatively large amount of effort.

Regarding RQ2, it turns out that having access to the support significantly increased learning outcomes. Learners with access to the support created higher quality representations in both lessons. The lesson on Star Formation showed a significant improvement in content knowledge for both the no support and support condition. However, the support condition resulted in a considerably higher post-test score, implying that the support condition learners learned more than their no-support counterparts. In the lesson on Neolithic Age, the support condition showed a significant improvement in content knowledge, while no significant improvement was found for the no support condition. Both conditions showed a significant improvement in their system thinking skills, operationalized as their understanding of the qualitative vocabulary (at level 3), after the first lesson on Star Formation. However, learners in the no support condition had a significant decrease in their understanding of the qualitative vocabulary between the Star Formation lesson and the Neolithic Age lesson, whereas those in the support condition seemed to have a more persistent learning gain. After the Neolithic Age lesson, learners in the support condition showed an improvement in their understanding of the qualitative vocabulary, while those in the no support condition did not. However, the difference in post-test scores between the two conditions after the Neolithic Age lesson was not significant.

What conclusions can be drawn about the effectiveness of the lessons on learning system thinking skills based on these results? Recall that the application used in our study has multiple levels of complexity. It appears that learning level 3 qualitative vocabulary is challenging and there seems to be limited improvement in comprehension after two lessons. This result is consistent with prior literature demonstrating that formal descriptions of systems pose a challenge for learners and take time to learn [3]. However, it can be noted that the lessons prioritize examining the causal relationships among quantities and their respective quantity spaces while other ingredients get less attention. For instance, learners make an exogenous influence once in both lessons. It turns out that by constructing the representations, learners can enhance their comprehension of subject content, even without a complete mastery of the vocabulary. They just need to follow the instructions in the workbook and correct any errors by taking advantage of the support function. That learning can be facilitated despite having an incomplete understanding of the vocabulary is advantageous. However, it is also important to teach learners a generic vocabulary for describing systems, as it is expected that this can aid in the transfer of understanding between different subject content. To ensure a successful lesson, it is necessary to strike a balance between focusing on the subject content and the learning of the formal vocabulary.

The effect of support on learning outcomes is most pronounced in the Neolithic Age lesson. While none of the learners in the no support condition were able to complete the representation of this lesson, 20 out of 27 learners managed to do so in the support condition. The Neolithic Age lesson’s representation is more complex than that of the Star Formation lesson, as it involves a larger number of ingredients and a greater number of causal relationships between quantities. Consequently, the Neolithic Age lesson may require more support. It is noteworthy that in the present study all learners in the support condition, except for one in the Neolithic Age lesson, were actively involved in constructing the representation throughout the lesson. Although the conditions are not entirely comparable, this is in contrast to the behaviour of learners in the support conditions in the VanLehn study [5] where a significant proportion was only actively interacting with the system for a short period of time. Having access to an automated support can help prevent learners from getting stuck (and maybe getting frustrated and giving up), thus increasing activity. However, caution should be exercised when directly comparing the use of support and its influence on learning behaviour and outcomes in our specific context with that in other dynamic systems modeling applications [4,5,6]. For instance, the distinction between describing a dynamic system through a mathematical approach versus using a qualitative vocabulary represents fundamentally different approaches, and consequently, the requirements for support may also diverge significantly [3].

In conclusion, the light-weight support appears to work well for helping learners learn subject content knowledge and develop general system thinking skills. It provides learners the support they (individually) need. Consequently, it should be considered for automated tutoring systems aimed at constructing qualitative representations, especially in situations where the application is used for many different lesson activities in different subject areas or where teachers use the system to create their own lesson activities.
5.1 Limitations

We focus on the implementation of qualitative representations in high school settings. As such, we developed lessons in collaboration with high school teachers and incorporated these into the actual curriculum. Such a setting is always less clean than a laboratory setting; it is often not possible to use randomized groups or to tightly control the conditions to isolate the effect of the independent variables. In such situations, research requirements need to be balanced with educational needs. However, this way of research also offers advantages because the approach we developed is meant to be implemented in such settings. Still, it does mean that there are always alternate explanations for the results.

5.2 Further development and research

Currently, more than 40 distinct lesson activities have been developed, each lasting approximately 2 hours. Typically, these lesson activities are carried out in just a few classes without pre-existing data. Therefore, implementing a lightweight automated support system is a valuable solution. We would like to add further lightweight support features. For instance, the current version only provides support for errors and not about what to do next. A way to improve the support is to add hints about next steps or missing ingredients. However, care has to be taken when implementing such hints, as learners might start exploiting the hints to finish the representation with minimal effort and minimal learning [4,15]. Research is needed to find the balance between preventing learners from getting stuck and allowing them to complete the lesson without learning anything. Another possibility is to add an automated lightweight meta-cognitive tutor that supports learning strategies and metacognitive activities such as goal setting, monitoring, and evaluation. Such a meta-cognitive tutor has been found effective in prior research [4,5]. It should be noted that the present pedagogical approach requires students to construct a representation in adherence to a norm, with only one correct answer. Alternatively, a more learner-centric approach would allow students to develop their own representations. In such cases, the existing support system is not applicable, and new forms of assistance need to be created [23].

To build upon the present study’s results, our future research will examine how frequently and under what circumstances users use cueing and help, also in relation to levels 2 and 4 of the application. We aim to examine whether different types of learners (e.g., novices, experts) use support functions differently, and if so, what factors contribute to these differences. By this, we seek to explore how the current support functions can be optimized to provide the most effective and efficient assistance to learners.

REFERENCES