Integrated Qualitative and Quantitative Modelling in GarpN

Marco Kragten[1], **Jan Wielemaker**[2], **and Bert Bredeweg**[1,3]
[1]Faculty of Education, Amsterdam University of Applied Sciences, Amsterdam, The Netherlands
[2]Faculty of Sciences, Vrije Universiteit Amsterdam, The Netherlands
[3]Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands
{m.kragten, b.bredeweg}@hva.nl , jan@swi-prolog.com

Abstract

We present GarpN, a software tool designed to integrate qualitative and quantitative modelling. GarpN has three functionalities: translating a qualitative representation into a quantitative model, running numerical simulations and detecting quantitative states, and matching these states to their qualitative counterparts. We show the use of GarpN through a worked example of modelling a spring–mass system. As GarpN is a work in progress, we also discuss current challenges in integrating qualitative and quantitative modelling.

1 Introduction

Modelling on the computer is regarded as a promising approach to help learners gain a deeper understanding of system behaviour [1, 2]. In education, both qualitative and quantitative modelling are used as instructional approaches [3, 4, 5] with each its own distinct advantages and downfalls for understanding systems [3, 5, 6, 7]. These two forms of modelling differ in how they represent and reason about system behaviour. Quantitative modelling using mathematical equations or stock-and-flow diagrams, allowing for precise predictions often shown in graphs or data tables [6]. It typically relies on a discrete time step simulation procedure, in which values are updated iteratively. Qualitative modelling, by contrast, represents system behaviour without relying on numerical data and emphasizes causal relationships and the possible states of a system [8]. It uses qualitative reasoning algorithms that apply logical inference rules to determine possible states and state transitions over time [9].

This paper presents **GarpN**, a software tool designed to integrate qualitative and quantitative modelling. GarpN aims to support students in understanding system behaviour by combining the strengths of both modelling approaches, while also leveraging how learning in one form can enhance understanding in the other. For example, learners may gain clearer insight into how a differential equation describes system behaviour after first exploring similar dynamics conceptually in a qualitative model.

The paper is structured as follows. Section 2 introduces qualitative modelling in DynaLearn [10]. Section 3 presents the GarpN workflow and its key functionalities. Section 4 discusses a worked example of using GarpN to model a spring–mass system. As GarpN is a work in progress, we conclude in Section 5 with a discussion of current challenges and directions for future development.

2 Qualitative modelling

The DynaLearn software enables learners to construct qualitative models across five levels of increasing complexity [10]. This paper focuses on how integrated qualitative and quantitative modelling is supported in GarpN at level 4. Therefore, this section describes the modelling ingredients available in DynaLearn at that level.

Entities are either physical objects or abstract concepts, characterized by one or more *quantities*—changeable features of entities. Each quantity has a *derivative*, denoted as δ, indicating its direction of change: decreasing, constant, or increasing. *Quantity spaces* define the possible states of the system by determining the range of possible *values* for each quantity, represented as alternating *point* and *interval* values. *Correspondences* (C) can be added to co-occurring values to further determine the possible states of the system. The relationships between quantities are described by two types of causal relationships: *influence* and *proportionality*. A causal relationship is of type influence (I) when an active process, indicated by a quantity, is the primary cause of a change in another quantity. This relationship can be either positive (I+) or negative (I-), depending on the directionality of the effect initiated by the process. When the relationship is of type positive, a positive **value** of the process results in an increase of the influenced quantity, while a negative value results in a decrease. In cases of a negative influence, a positive value of the process causes a decrease in the influenced quantity, and a negative value causes an increase. Causal relationships of type proportionality (P) describe how **changes** in one quantity lead to similar changes in another quantity, either in the same direction (P+) or in opposite directions (P-). Exogenous influences are external factors that have a continuous effect on the change of a quantity.

Simulation within Dynalearn starts with a scenario: the initial settings that define the starting conditions of the model. From these settings, a state graph is generated, visually representing the possible states and transitions of the system.

Learners can use this graph to explore the behaviour of the system by navigating through different states.

3 GarpN workflow and functionalities

The workflow of GarpN (Fig. 1) consists of three functionalities: (*i*) translation of the qualitative model into a simulation script with corresponding mathematical equations, (*ii*) parsing the script, running a numerical simulation and quantitative state detection, and (*iii*) matching between qualitative and quantitative states. GarpN initiates the qualitative reasoning algorithms to generate the qualitative states required for matching.

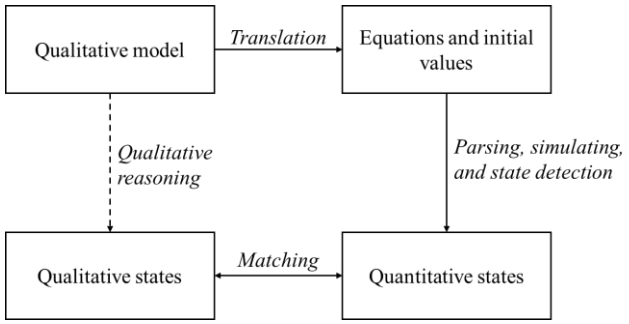

Fig. 1. GarpN workflow and functionalities.

The qualitative model and its simulation results shown in Fig. 2 are used in this section to demonstrate the functionalities of GarpN. This model is not based on a real-world system; instead, it employs placeholder names—*Q1*, *Q2*, and *Q3* for quantities and *E1*, *E2* for entities—to keep the notation concise and focus attention on specific features of GarpN.

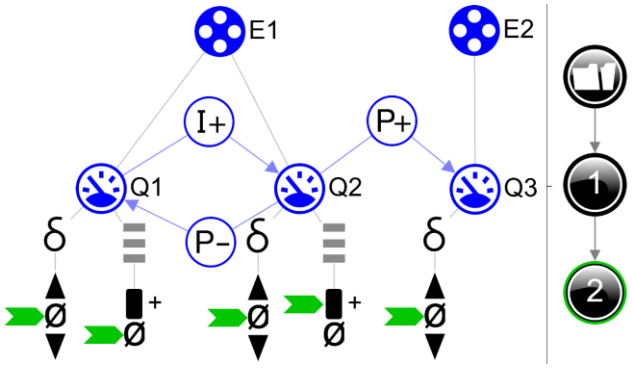

Fig. 2. Qualitive model and simulation results used for demonstrating GarpN functionalities. The right side shows the state graph starting with the scenario followed by two consecutive states (state 2 is selected). The left side shows the model and the simulation result of state 2 (green arrows).

Entity *E1* has quantities *Q1* and *Q2*, both of which have a quantity space defined as $\{0, +\}$. Entity *E2* has quantity *Q3*, which does not have a defined quantity space. *Q1* exerts a positive influence (I+) on *Q2*, while *Q2* has a negative proportional relationship (P^-) with *Q1*, forming a negative feedback loop. In addition, *Q2* has a positively proportional relationship (P+) with *Q3*.

The initial values of *Q1* and *Q2* are set to positive (+) (not shown in Fig. 2). The state graph shows that a simulation with these values results in two consecutive states. State 2 (S_2) is selected in Fig. 2. Green arrows indicate the values and derivatives of the quantities at that state.

The full simulation result per state for each quantity is summarized in Table 1. For each state the value and direction of change of the quantities is given as a tuple $\langle \boldsymbol{v}, \delta \rangle$.

Table 1. Simulation results per state with value and direction of change for each quantity of the model shown in Fig. 2.

	Q1	*Q2*	*Q3*
S_1	⟨+, -⟩	⟨+, +⟩	⟨u, +⟩
S_2	⟨+, 0⟩	⟨+, 0⟩	⟨u, 0⟩

u = undefined.

3.1 Translation

How qualitative models relate to numerical models is not the primary focus of this paper. For a more detailed discussion, we refer the reader to our previous work [11]. For this paper, it suffices to note that the translation depends on the types of relationships between quantities. In Fig. 2, *Q1* exerts a positive influence (I+) on *Q2*, meaning that the **value** of *Q1* determines the rate of change of *Q2*. This relationship translates into a differential equation for *Q2*. In contrast, *Q2* and *Q3* have a positive proportional relationship (P+), which can be expressed using a mathematical equation that preserves the direction of change—i.e., when *Q2* increases, so does *Q3*. By default, GarpN translates proportional relationships to linear equations.

The translation functionality of GarpN takes a qualitative representation as input and applies a set of predefined rules to generate a simulation script. This script includes four sections: (*i*) a list of equations, (*ii*) constants, (*iii*) initial values, and (*iv*) time-related settings.

Script notation. When translating the qualitative representation into the simulation script, quantities are labelled using their own name along with the name of their associated entity as a superscript—for example, $Q1^{E1}$. While qualitative modelling does not strictly require quantity names to be unique (though parsimony is often preferred), GarpN enforces uniqueness to enable numerical calculations. Additionally, each quantity in the script must have a unique identifier so that GarpN can correctly associate it with its counterpart in the qualitative model when matching qualitative and quantitative states. For this reason, the automatically generated names of quantities cannot be modified. Note that when there is only one entity and/or when all quantity names are already unique, labelling quantities with the entity name as a superscript is unnecessary and therefore omitted during translation.

Additionally, the script does not use explicit time notation such as *Q(t)* or *Q(t−1)*. New values are calculated based on the previous value using the mathematical operations defined in the script. To keep the equations concise and readable, GarpN includes a built-in function that detects whether symbols are placed between parentheses in the names of

quantities within the qualitative representation, e.g., *Temperature (T)*. If such a symbol is present, it is used in the generated equations, allowing for simplified expressions that maintain a clear connection to the qualitative model. If the symbol is not unique, it is automatically extended with the entity name as a superscript to ensure uniqueness. These mechanisms are designed to prevent unnecessary complexity in the script while preserving traceability between the qualitative and quantitative models.

Translation modes. When translating a qualitative representation into mathematical equations, one of four translation modes must be selected: *full*, *derivatives*, *values*, or *mixed*. These modes offer flexibility in how causal relationships between quantities are expressed in the simulation script.

In **full mode**, equations are generated for both the derivatives and values of all quantities, regardless of whether a quantity has a defined quantity space. Note that in DynaLearn, adding a quantity space is optional—it is a modeling choice rather than a requirement. A quantity space is only necessary when distinguishing between different qualitative values is important for describing system behaviour. By default, a quantity in DynaLearn only has a derivative. Translating the qualitative model in Fig. 2 using full mode generates the following equations:

$$\Delta Q2 = c_1 \cdot Q1 \cdot \Delta t \quad (1)$$
$$Q2 = Q2 + \Delta Q2 \quad (2)$$
$$\Delta Q1 = c_2 \cdot \Delta Q2 \quad (3)$$
$$Q1 = Q1 + \Delta Q1 \quad (4)$$
$$\Delta Q3 = c_3 \cdot \Delta Q2 \quad (5)$$
$$Q3 = Q3 + \Delta Q3 \quad (6)$$

In this mode, each quantity is assigned two equations: one to compute its derivative (Δ) and another to update its value. Even though *Q3* has no quantity space, an additive update equation (6) is still generated. Combined, (1) and (2) constitute a basic differential equation. For proportional relationships, linear functions are proposed by default. Full mode requires the most input: initial values must be set for constants (c_1, c_2, c_3) and quantities (*Q1*, *Q2*, *Q3*).

In **derivative mode**, GarpN generates equations that calculate only the derivatives of quantities wherever possible, even if those quantities have a quantity space. For example, although *Q2* has a quantity space, only its derivative is calculated. However, this approach is not applicable to quantities that represent processes within the system, as their values determine the rate of change of other quantities. For these, both a derivative and a value equation are required. *Q1* is such a process quantity. As a result, GarpN will generate Equations (1), (3), (4), and (5), omitting value equations for *Q2* and *Q3*. In this mode, initial values for *Q2* and *Q3* do not need to be set.

In **value mode**, GarpN generates equations based solely on **values**, even for quantities that do not have a defined quantity space. Unlike **full mode**, this approach omits the explicit calculation of derivatives—no separate equations for derivatives are generated in the script. Instead, value updates are calculated directly. In this mode, GarpN generates Equations (4), (6), and (7):

$$Q2 = Q2 + c_1 \cdot Q1 \cdot \Delta t \quad (7)$$

This mode reduces the number of equations and the amount of input required, but it may obscure the distinction between value and rate-of-change, which is explicit in other modes.

In **mixed mode**, GarpN determines the type of equation to generate based on whether the target quantity—the quantity being influenced—has a defined quantity space. If a quantity has a quantity space, GarpN generates equations based on its value, and the derivative is not explicitly calculated. If a quantity lacks a quantity space, only its derivative is calculated. This mode aligns most closely with the information in the qualitative representation itself. For the model in Fig. 2, GarpN generates (4), (5), and (7).

It is important to note that all four modes describe the same underlying system behaviour. However, the required constants and initial values may differ depending on the mode.

Constants. By default, any constants generated during the translation process are included in the constants section of the simulation script. The names of these constants are editable. Additional constants may also be introduced if required for accurate numerical simulation, even if these are not part of the original qualitative representation. Such constants may not be essential for describing the qualitative behaviour of the system but needed for valid quantitative calculations. GarpN provides flexibility to define and adjust these as necessary. Note that the constant c_1—used in (1) and (7)—functions as a scalar that adjusts the size of the value of *Q1* (a process) on *Q2*. In most cases, unit consistency ensures that no scaling is required: if *Q1* is in units per time, and *Q2* is in the corresponding unit, then c_1 can be set to 1. GarpN assumes this by default, but the scalar can be modified if rescaling is needed. A planned enhancement will allow omission of such scalars altogether when unit consistency is assumed, helping to simplify the generated equations and improve readability.

Initial values. GarpN automatically detects which quantities require an initial value for numerical simulation. In qualitative representations, zero is the only explicit numerical value; all other values are indicated only by their sign (positive or negative). If a quantity is explicitly set to zero in the qualitative model, GarpN carries this over and initializes it as zero in the script. For all other quantities, GarpN can infer whether the value is above or below zero, but not the exact magnitude. These values must be manually specified before the numerical simulation can be run.

Time settings. Time-related initial values and equations are organized in a separate section to maintain clarity. By default, GarpN generates the equation $t = t + \Delta t$, sets initial time to $t = 0$, and proposes a default time step of $\Delta t = 0.1$.

3.2 Parsing, simulation and state detection

Once the script is complete and all constants and initial values are defined, the numerical simulation can be run. Rather than specifying an end time—as is common in some numerical

modelling environments—GarpN requires setting the number of iterations. This choice reinforces the idea of numerical simulation as a stepwise, iterative process. Before running a simulation, the script is parsed to prepare it for execution. For this, an integration method must be chosen; GarpN currently supports both the Euler method and the Runge–Kutta method (RK4). Simulation results are visualized in a graph that plots the values and derivatives of quantities over time. These can be selectively displayed or hidden. Additionally, a downloadable CSV file is available, providing the full time-series data for further analysis.

3.3 Matching

GarpN includes an automatic matching functionality that maps qualitative states with the corresponding quantitative states. By default, this matching is based on the quantities and derivatives that are explicitly calculated in the script. However, the matching criteria can be adjusted using a **matching settings** panel (Fig. 3). Additionally, matching based on second derivatives is available as an optional setting. GarpN calculates and stores these values internally during simulation, allowing them to be used for analysis and state matching.

	Matching on:		
Quantity	$\mathcal{V}$	Δ	Δ²
Q1	☒	☒	☐
Q2	☒	☒	☐
Q3	☐	☒	☐

Fig. 3. Matching settings panel. Settings correspond to the equations of the model in Fig. 2 generated in mixed mode.

The graph can be interacted with to inspect specific time intervals. When a time interval is selected, GarpN shows a table how the quantitative behaviour aligns (or fails to align) with the qualitative states.

Table 2. Matching of quantitative behaviour and qualitative states of the model shown in Fig. 2 for a specific time interval.

	Q1		*Q2*		*Q3*	
Time	$\mathcal{V}$	Δ	$\mathcal{V}$	Δ	Δ	State
3.10	+	▼	+	▲	▲	S1
3.20	+	▼	+	▲	▲	S1

Figure 4 shows the result of the numerical simulation based on the equations generated in mixed mode, translated from the qualitative representation shown in Fig. 2. The parameter and initial values used are: $c_1 = -0.3$, $c_3 = 1.5$, *Q1* = 2, and *Q2* = 4. Matching between qualitative and quantitative states is visualized in the graph. Vertical dotted lines distinguish quantitative states. Matching qualitative state numbers are displayed above the graph: when a state spans an interval, its number appears between two dotted lines; when it aligns with a single point, the number is placed above a single line.

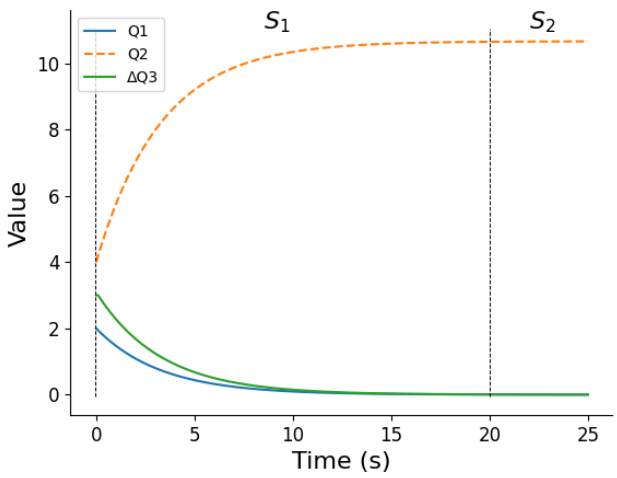

Fig. 4. Numerical simulation results and matching of states from the qualitative model shown in Fig. 2.

4 Modelling a spring–mass system with GarpN

In this section, we demonstrate how integrated qualitative and quantitative modelling using GarpN can be incorporated into a lesson, using the example of a mass–spring system. The main goal of this lesson is to help learners understand the relationships between force, acceleration, velocity, and position in a mass–spring system. Assuming no friction, such a system exhibits simple harmonic motion. When a spring is compressed and then released, it oscillates around an equilibrium position. During this oscillation, the spring alternately stretches and compresses. The position of the mass relative to the equilibrium point determines the restoring force directed toward that equilibrium. This force causes an acceleration in the same direction. The acceleration changes the velocity of the mass, which in turn changes its position. This change in position creates a new restoring force, continuing the cycle.

Learners are guided by a workbook that supports the step-by-step construction of the qualitative representation. At each step, the qualitative representation is translated into its corresponding numerical form using GarpN. Qualitative and numerical simulations are run at each step, and learners investigate how quantitative states match to their qualitative counterparts. Students are prompted with reflection questions to deepen their understanding of the behaviour of the system. Throughout the lesson, the mixed translation mode and default time settings in GarpN are used.

In the first step, learners construct the entity *Spring-mass system* and add two quantities: *Force (F)* and *Acceleration (a)*. They are introduced to the principle that acceleration is proportional to net force, following Newton's second law ($a = F/m$), and add a positive proportional relationship between *Force (F)* and *Acceleration (a)*. Fig. 5 shows the representation and the simulation result. An exogenous increasing influence is acting on *Force (F)*. The state graph shows a single state: as *Force (F)* increases, *Acceleration (a)* also increases.

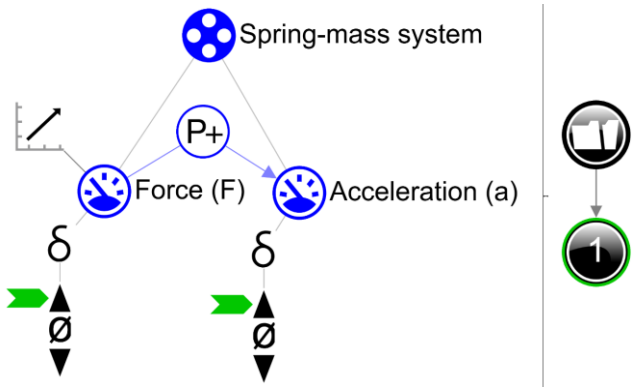

Fig. 5. A positive proportional relationship between force and acceleration. An increasing exogenous influence is acting on the force of the spring. The simulation result of state 1 is shown.

Learners are than asked to translate the qualitative representation into a numerical model. At this stage, neither *Force (F)* nor *Acceleration (a)* has a quantity space, so in the mixed translation mode, GarpN generates equations based on derivatives. The resulting equations are:

$$\Delta F = c_1 \quad (8)$$
$$\Delta a = c_2 \cdot \Delta F \quad (9)$$

Note that in the qualitative representation, an increasing exogenous influence is acting on force, meaning $\Delta F > 0$. In (8), the constant c_1 represents the rate at which force increases per unit of time. When translating an increasing exogenous influence into a numerical equation, GarpN assumes a constant rate of increase, but any equation for which $\Delta F > 0$ would be consistent with the qualitative representation [11]. Students are asked to set c_1 to 0.2 Newton. The constant c_2 is a scalar conversion factor and is set to 1 by default.

Fig. 6 shows the result of the numerical simulation, which was run for 250 iterations (equivalent to 25 seconds with a time step $\Delta t = 0.1$). The graph shows that the ΔF is constant rate of 0.2 Newton per time step. GarpN determines that qualitative state 1 (S_1) corresponds to the entire duration of the numerical simulation. The Δa is not shown, as it is identical to ΔF.

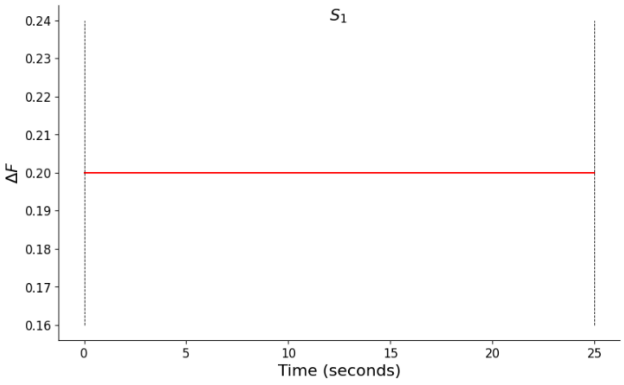

Fig. 6. Numerical simulation results and matching of states of the qualitative model shown in Fig. 5.

Next, learner adds a quantity space defined as {-, 0, +} to the quantities *Force (F)* and *Acceleration (a)*. A correspondence is added between these quantity spaces, stating that when *Force (F)* is negative, *Acceleration (a)* is also negative; when *Force (F)* is zero, *Acceleration (a)* is zero; and when *Force (F)* is positive, *Acceleration (a)* is positive.

Fig. 7 shows the qualitative representation. The initial value for *Force (F)* is set to negative (-) (not shown). This leads to three consecutive states. Fig. 7 shows the simulation result for state 3, while Table 3 shows the values and changes of direction of both quantities across all states.

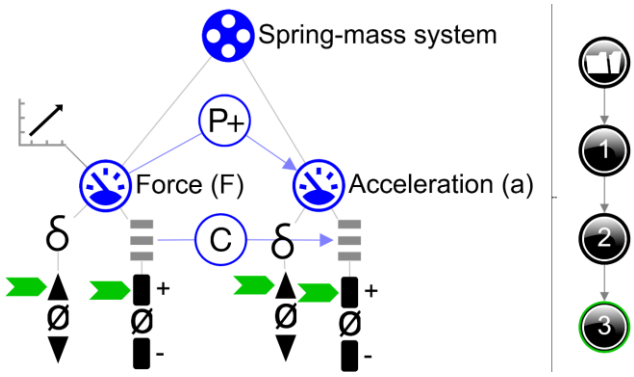

Fig. 7. A correspondence between the quantity spaces of force and acceleration. The simulation result of state 3 is shown.

In S_1, *Force (F)* is negative (-) and increasing; in S_2, *Force (F)* is zero (0) and increasing; and in S_3, *Force (F)* is positive (+) and still increasing. *Acceleration (a)* behaves identically across all three states, reflecting the positive proportional relationship between the two quantities and correspondence between their quantity spaces.

Table 3. Simulation results per state with value and direction of change for each quantity of the model shown in Fig. 7.

	Force (F)	*Acceleration (a)*
S_1	⟨-, +⟩	⟨-, +⟩
S_2	⟨0, +⟩	⟨0, +⟩
S_3	⟨+, +⟩	⟨+, +⟩

Translation by GarpN now generates the equations:

$$F = F + c_1 \cdot \Delta t \quad (9)$$
$$a = c_2 \cdot F \quad (10)$$

Equation (9) is a differential equation: the value of force at each time step is computed by adding its change ($c_1 \cdot \Delta t$) to its previous value. Equation (10) is a multiplicative expression that directly relates acceleration to force. This contrasts with the additive form used in (4) and (6) in the previous section, where values were calculated incrementally. Here, GarpN infers the multiplicative form from the positive proportional relationship and the correspondence specified in the qualitative representation.

At this point, mass is introduced. Although mass is not part of the qualitative representation—it is assumed to be constant

and thus does not affect the qualitative behaviour—it is required for quantitative calculation. Learners therefore add mass as a constant (i.e., m = 1 kg), and update the equation for acceleration accordingly. The equation $a = c_2 \cdot F$ is replaced by $a = F / m$, and the scalar c_2 is removed. For running the numerical simulation, the initial value of F also needs to be set explicitly.

Fig. 8 shows the simulation result, with an initial value of $F = -2$ and a rate of change $c_1 = 0.2$. From $t = 0$ to $t < 10$, force remains negative, corresponding to S_1. At $t = 10$, F equals zero, corresponding to state S_2. From $t > 10$ to $t = 25$, F is positive, corresponding to S_3. Note that if the simulation is run for too short—for example, only 50 iteration (5 seconds)—only S_1 will be reached, and subsequent states will not occur.

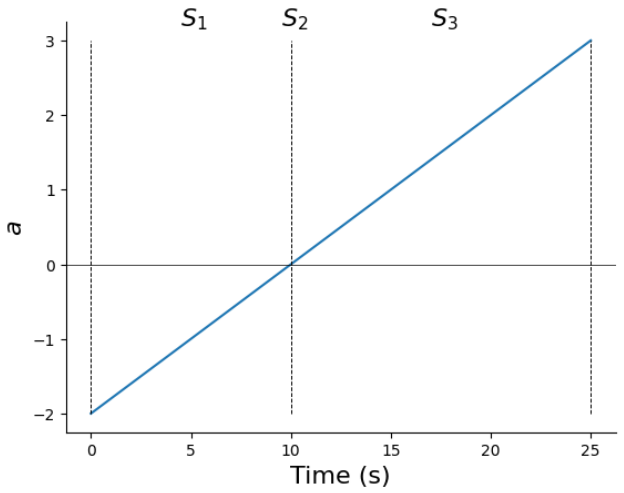

Fig. 8. Numerical simulation results and matching of states of the qualitative model shown in Fig. 7.

In the next step, learners add the quantity *Velocity (v)*, with a quantity space defined as {-, 0, +} and a causal relationship between *Acceleration (a)* and *Velocity (v)* of type positive influence (I+). The initial values for *Force (F)* and *Acceleration (a)* remain as in the previous simulation, and *Velocity* is set to a negative (-) initial value. Fig. 9 shows the simulation results with five consecutive qualitative states (S_3 is shown).

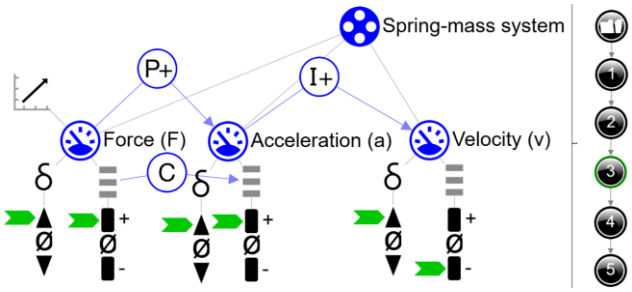

Fig. 9. Causal relationship between acceleration and velocity of type positive influence. The simulation result of state 3 is shown.

Table 4 presents the value and direction of change for each quantity across all five states. In S_1, *Velocity (v)* is negative and becoming more negative, indicating that velocity is increasing in the negative direction. This is due to a negative acceleration acting in the same direction. Since force is increasing (it is moving towards zero and thus becoming less negative), acceleration becomes less negative. In S_2, acceleration reaches zero, meaning there is no longer a change in velocity; velocity remains negative and becomes constant . In S_3, acceleration becomes positive—opposing the negative velocity—so the object slows down (i.e., velocity becomes less negative). In S_4, velocity reaches zero and acceleration continues to act in the positive direction. In S_5, velocity is positive and increasing.

Table 4. Simulation results per state with value and direction of change for each quantity of the model shown in Fig. 9.

	Force (F)	*Acceleration (a)*	*Velocity (v)*
S_1	⟨-, +⟩	⟨-, +⟩	⟨-, -⟩
S_2	⟨0, +⟩	⟨0, +⟩	⟨-, 0⟩
S_3	⟨+, +⟩	⟨+, +⟩	⟨-, +⟩
S_4	⟨+, +⟩	⟨+, +⟩	⟨0, +⟩
S_5	⟨+, +⟩	⟨+, +⟩	⟨+, +⟩

The addition of *Velocity (v)* to the qualitative representation results in GarpN generating a differential equation for the simulation script:

$$v = v + a \cdot \Delta t \qquad (11)$$

An initial value for v must now be specified. Any negative value is suitable to reflect the direction indicated in the qualitative model; in this case, we set the initial velocity: v = -5, while retaining all other initial settings from the previous simulation. Fig. 10 shows the simulation result, including the matching of the five qualitative states to the corresponding quantitative states.

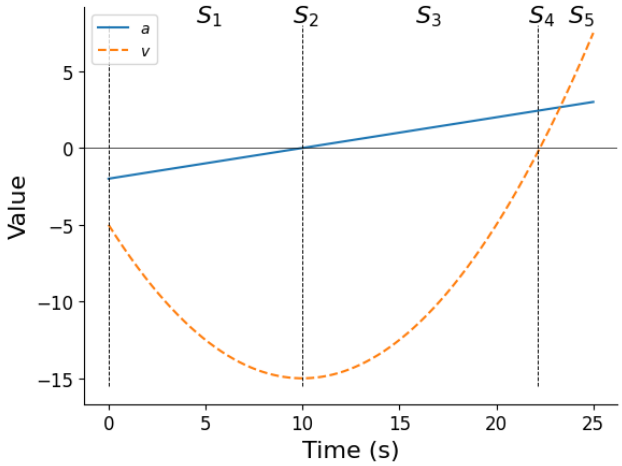

Fig. 10. Numerical simulation results and matching of states of the qualitative model shown in Fig. 9.

In the final step, learners add the quantity *Position (x)*, with a quantity space defined as {-, 0, +} and a causal relationship between *Velocity (v)* and *Position (x)* of type positive influence (I+). Additionally, learners add a negative proportional relationship between *Position (x)* and *Force (F)*, along with an inverted correspondence: when *Force (F)* is positive (+), *Position (x)* is negative (-); when *Force (F)* is zero (0), *Position (x)* is zero (0); and when *Force (F)* is negative (-), *Position (x)* is positive (+). At this point, the exogenous influence acting on *Force (F)* can be removed, as the force exerted by the spring is now internally determined by the displacement from equilibrium (i.e., its position). Fig. 11 shows the qualitative representation and the simulation results, with initial settings: *Velocity (v)* is zero (0) and *Position (x)* is positive (+). The state graph shows a recurring pattern of eight states, corresponding to the simple harmonic motion of the system.

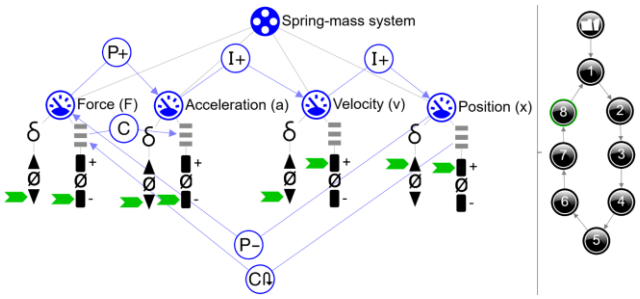

Fig. 11. Full model of the spring-mass system. The position of the mass is negative proportional related to the force exerted by the spring. The simulation results in a recurring pattern of eight states. State 8 is selected.

Table 5 shows the simulation results per state with value and direction of change for each quantity of the model shown in Fig. 11.

Table 5. Simulation results per state with value and direction of change for each quantity of the model shown in Fig. 11.

	Force (F)	*Acceleration (a)*	*Velocity (v)*	*Position (x)*
S_1	⟨-, 0⟩	⟨-, 0⟩	⟨0, -⟩	⟨+, 0⟩
S_2	⟨-, +⟩	⟨-, +⟩	⟨-, -⟩	⟨+, -⟩
S_3	⟨0, +⟩	⟨0, +⟩	⟨-, 0⟩	⟨0, -⟩
S_4	⟨+, +⟩	⟨+, +⟩	⟨-, +⟩	⟨-, -⟩
S_5	⟨+, 0⟩	⟨+, 0⟩	⟨0, +⟩	⟨-, 0⟩
S_6	⟨+, -⟩	⟨+, -⟩	⟨+, +⟩	⟨-, +⟩
S_7	⟨0, -⟩	⟨0, -⟩	⟨+, 0⟩	⟨0, +⟩
S_8	⟨-, -⟩	⟨-, -⟩	⟨+, -⟩	⟨+, +⟩

The additions to the qualitative representation in the final step lead GarpN to generate the following equations for the simulation script:

$$x = x + v \cdot \Delta t \quad (12)$$

$$F = -c_3 \cdot x \quad (13)$$

Equation (9), which previously represented the exogenous influence on Force, is now removed and replaced by (13), reflecting the spring's internal dynamics. The negative sign in (13) captures the negative proportional relationship between *Position (x)* and *Force (F)*, as well as the inverted correspondence between their quantity spaces. Students are instructed to rename the constant c_3 as the spring constant k (in N/m) and to set its value to $k = 0.1$.

Fig. 12 shows the simulation result, including the matching of the eight qualitative states to the corresponding quantitative states.

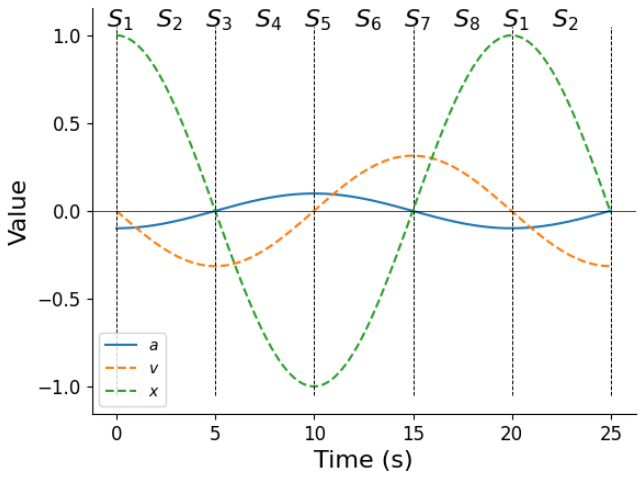

Fig. 12. Numerical simulation results and matching of states of the qualitative model shown in Fig. 11.

5 Conclusion and discussion

This paper presents GarpN, a tool that demonstrates the technical feasibility of integrating qualitative and quantitative modelling. While still under development, GarpN already shows potential as a pedagogical tool for helping learners understand system behaviour. Through a worked example, we illustrated how integrated modelling can be implemented in an educational context. That said, several challenges remain to be addressed in further development. These relate to three areas: the translation of qualitative models into numerical scripts, the mapping of qualitative and quantitative simulation states, and the refinement of the pedagogical approach—including user interface design—to most effectively support understanding of system behaviour.

Translation. Several challenges remain regarding the translation functionality in GarpN. One key challenge is proposing a numerical equation that best fits a given proportional relationship in the qualitative model. The most appropriate equation depends on multiple factors: the sign of the proportional relationship, the quantity spaces of the related quantities, defined correspondences, and initial values [11]. For example, a proportional relationship between *Q2* and *Q1* of the model shown in Fig. 2 is best expressed through an additive equation. A multiplication equation such as $Q1 = -c_1 \cdot Q2$ would not fit, as it would result in negative values for *Q1* while its quantity space is defined as {0, +}.

When multiple quantities are proportionally related to another quantity, the complexity of suggesting an appropriate

equation increases further. At present, GarpN supports only relatively simple cases, as demonstrated in this paper. For instance, in the model shown in Fig. 2, an additive equation, i.e., (7), is inferred as the best fit, whereas in Fig. 11, proportional relationships between force and acceleration, and between position and force, are better described using multiplication equations, i.e., (10) and (13). Procedures for translating other variants of proportional relationships are planned for future development.

Another issue is how simulation preferences in Dynalearn affect the qualitative simulation results. For example, preferences can be set to determine the depth of reasoning—where possible states and transitions are calculated by tracing causal effects through changes in the first, second, or third derivatives. If the qualitative reasoning algorithm is configured to reason only with first-order derivatives, but the translated numerical equations exhibit second-order behaviour, a mismatch between qualitative and quantitative states may occur.

A further challenge in translation is that it is not always evident which values of constants or initial quantity settings will yield simulation results that match the qualitative behaviour. This also applies to the number of iterations required to reach specific states. The issue is compounded when learners build a model step by step: each step may introduce new equations and change simulation dynamics, requiring adjusted settings for proper matching.

While many of these challenges can be mitigated through structured guidance in a tightly designed lesson plan, supporting learners who use GarpN in a more exploratory way will require improved diagnostics, feedback, and scaffolding in future versions.

Matching. One of the main challenges lies in matching qualitative and quantitative simulation results, as certain qualitative states may not appear in the numerical output. A particular case of this challenge is at the start of the simulation. For example, in the model presented in Fig. 2, if quantity *Q2* has an initial value of zero, the simulation result includes an additional state prior to qualitative state 1 (see Table 1), where *Q1* is ⟨+, -⟩, *Q2* is ⟨0, +⟩, and *Q3* is ⟨u, +⟩. In a corresponding numerical simulation, if the initial values are $Q1 = 5$ and $Q2 = 0$, then according to (7), *Q2* is updated immediately: $Q2 = 0 + 5 \cdot 0.1 = 0.5$. Thus, already after the first iteration, *Q2* is both positive and increasing, skipping the qualitative state where *Q2* is still zero but increasing. This type of mismatch becomes more pronounced when a causal chain includes multiple successive influences.

A related challenge in matching arises when multiple quantities are expected to reach a value or direction of change simultaneously within a qualitative state. For example, in state 1 of the model in Fig. 11, the mass is at its maximum position and steady ⟨+, 0⟩, while its velocity is zero and decreasing ⟨0, -⟩. However, due to the discrete time-step approach in numerical simulation, these conditions may unfold over several iterations rather than occurring concurrently.

Another matching challenge arises when a quantity in a qualitative simulation passes through zero—transitioning, for instance, from a negative to a positive value. This results in three distinct qualitative states: one in the negative interval, one at zero, and one in the positive interval. In a numerical simulation, however, it is unlikely that a value will be exactly zero at a specific iteration, meaning this quantitative state effectively occurs between two time steps. GarpN includes a procedure to detect such transitions around zero. However, similar challenges occur for non-zero point values—such as reaching a boiling point that marks the transition between liquid and gas states. To handle such transitions, the numerical equivalent of these point values must be explicitly defined, allowing the GarpN to detect when they are crossed. This functionality has not yet been implemented and remains a task for future development.

Matching is also challenging when quantities approach a steady state—i.e., when their derivatives become zero. The corresponding mathematical equations often exhibit asymptotic behaviour: values continue to change by ever smaller amounts and may never exactly reach a constant value. To address this, GarpN uses threshold-based matching. When the change of direction of a quantity falls below a predefined threshold, it is treated as effectively zero. Importantly, in many cases, multiple quantities must simultaneously meet this threshold to match a qualitative state. For example, in state 2 of the model shown in Fig. 2, *Q1*, *Q2*, and *Q3* all become steady at the same time.

Thus, simulation results from qualitative and quantitative models do not align one-to-one in terms of temporal progression. In qualitative reasoning, the principle that "cause precedes effect" is represented by a sequence of discrete states, whereas numerical models update all values simultaneously at each time step.

A critical design challenge for GarpN is how to detect such mismatches and how to present them to learners in a way that supports, rather than confuses, their conceptual understanding of causality and system dynamics.

Pedagogical approach. At present, GarpN runs as a separate browser tab. Future development will need to focus on how to embed it within the Dynalearn canvas. This includes exploring which user interface designs best support learning processes and how the tool can be seamlessly integrated into modelling workflows. In addition, the pedagogical approach for integrated modelling should be further developed. This approach should go beyond simply presenting qualitative and quantitative modelling side by side; rather, it should promote synergy between the two. For example, learners might better interpret the meaning of qualitative states and transitions when they also view their numerical counterparts as line graphs. Developing and testing such an approach in classroom settings will be a next step.

References

[1] Jacobson, M., & Wilensky, U.: Complex systems in education: Scientific and educational importance and implications for the learning sciences. *The Journal of the Learning Sciences*, 15(1), 11-34, 2006.

[2] National Research Council: *Next generation science standards: For states, by states*, 2013.

[3] Bredeweg, B., Kragten, M., Holt, J., Kruit, P., van Eijck, T., Pijls, M., Bouwer, A., Sprinkhuizen, M., Jaspar, E., & de Boer, M: Learning with interactive knowledge representations. *Applied Sciences*, 13(9), 5256, 2023.

[4] Fisher, D.: Reflections on teaching system dynamics modelling to secondary school students for over 20 Years. *Systems*, 6(2), 12, 2018.

[5] VanLehn, K.: Model construction as a learning activity: A design space and review. *Interactive Learning Environments*, *21*(4), 371-413, 2013.

[6] Sins, P., Savelsbergh, E., van Joolingen, W. R.: The difficult process of scientific modelling: An analysis of novices' reasoning during computer-based modelling. *International Journal of Science Education*, 27(14), 1695-1721, 2005.

[7] Kragten, M., Jaspar, E., & Bredeweg, B.: Learning domain knowledge and systems thinking using qualitative representations in secondary education (grades 10-12). *In Proceedings of the 35th International Workshop on Qualitative Reasoning*. Vienna, Austria, 2022.

[8] Forbus, K. D.: *Qualitative representations: How people reason and learn about the continuous world*. MIT Press, 2019.

[9] Bredeweg, B., Linnebank, F., Bouwer, A., & Liem, J.: Garp3—Workbench for qualitative modelling and simulation. Ecological informatics, 4(5-6), 263-281, 2009.

[10] Bredeweg, B., Liem, J., Beek, W., Linnebank, F., Gracia, J., Lozano, E., Wißner, M., Bühling, R., Salles, P., Noble, R. A., Zitek, A., Borisova, P., & Mioduser, D.: DynaLearn - An intelligent learning environment for learning conceptual knowledge. *AI Magazine*, 34(4), 46-65, 2013.

[11] Kragten, M., & Bredeweg, B.: Qualitative and quantitative modelling of dynamic systems: how do they relate? In *Proceedings of the 37th International Workshop on Qualitative Reasoning*. Santiago de Compostela, Spain, 2024.