Causal Loop Diagrams and Automated Identification of Feedbacks in Economic Systems

Vladimír BUREŠ

University of Hradec Kralove, Hradec Kralove, Czech Republic; vladimir.bures@uhk.cz

Abstract: Systems dynamics is a universal and widely used discipline that examines systems' behaviour over time. The casual loop diagram represents one of the commonly applied tools during the analysis of various economic problems. This paper presents a tool facilitating the work of managers who are forced to work with different perspectives on one shared issue. The created application enables mechanical analysis of a diagram created in the Vensim Pro application. The program is used for fast validation of models and checking the correctness of the structure. Based on the similarity of the structure, shared or different loops can be highlighted. The results can serve as a basis for developing strategies, policies or group discussions directed at analysing existing economic systems.

Keywords: economic systems; model comparison; causal loop diagram; system dynamics; automated identification

JEL Classification: C61; C63

1. Introduction

System dynamics deals with understanding how the system behaviour changes over time and is gaining popularity due to its flexibility and structural focus (Kelly et al., 2013). It has already been applied to support decision-making in many disciplines ranging from medicine and health care (Clarke et al., 2021; Loyo et al., 2013; Sahin et al., 2020) to urban coastal systems (Mavrommati et al., 2013). Although system dynamics has its intellectual origins in control engineering, management science, and digital computing, it currently shapes the frontier of better-understanding the behaviour of social and economic systems (Forrester, 2003). System dynamics models often generate counterintuitive behaviour and are at odds with traditional economic theory (Radzicki, 2009; Režný & Bureš, 2018). However, models represent essential tools for generating new knowledge as they help us uncover dominant feedback, unintended consequences, synergic effects, or emergent attributes (Bureš & Tučník, 2014). We look for patterns within the details without losing sight of the big picture. Analysis of economic systems is widely associated with the presentation of mental models in which economic systems are captured. Development of ability of collection and organisation of data, interdisciplinary model development, transparency of models, and visualisation of the results represent the main issue. System dynamics helps to cope with complexity and acquire better insights, which is the primary assumption of better understanding and meaningful decision making. This need constitutes the main rationale and motivation of this study.

This paper is based on the analysis of models of economic systems represented by causal loop diagrams (CLD). CLD represent the notation of the bounded system structure which is composed of variables and their relationships that interact with each other. This work aims to present a software application that reads a model created in the Vensim Pro software application and verifies the correctness of the CLD stored in this model. The manuscript is structured as follows. After this introductory section, the applied methodology focused on the CLD structure and associated MDL file details created in the Vensim Pro application is presented. Next section introduces acquired results. It focuses on the validation of diagrams, the process of loop identification, and how outcomes are provided. The last section concludes the paper.

2. Methodology

This study's main assumption is based on the conviction that economic systems, as an instance of soft systems, are not mostly associated with an exact description of structures and behaviours. However, the notation of CLD is rigorous and can be analysed. Binder et al. (2004) describe CLD as an oriented graph. Since we need to go through the graph, it is recommended to use an adjacency matrix and determine each link's polarity. It is also possible to successfully detect multiple relationships and record the absolute number of relationships without polarity (two relationships with opposite polarity would be subtracted from each other). We do not know if they are variables without a relationship, or with one either positive or negative polarity). The algorithm continues with depth-first and breadth-first search. Because we are looking for all the loops in the graph and do not know if there will be more loops, this algorithm is not suitable as a stand-alone method. For instance, the search can be applied to each variable where a loop would be found. However, it is possible that in a variable, which belongs to more than one loop, the algorithm chooses the wrong one, and we do not get all loops even in multiple runs in the graph.

There are various methods for dealing with feedbacks in system dynamics models. The Loops that Matter method can serve as an example (Schoenberg et al., 2020). When determining the algorithm for finding loops, it is interesting to be inspired by Oliva's work (2004), in which an adjacency matrix capturing variables and relationships is used. It further enriches it by using a distance matrix, showing how far variables along the link path are from each other. This approach ensures finding loops consecutively from small to larger ones and always tests when inserting a new loop into the list if it differs by at least one link from the previous one.

The solution presented in this study is based on the adjacency matrix analysis, where for each variable, the output link is inserted into the list of potential loops. At the same time, the algorithm goes through all links already inserted into the list. If any of these links starts with the end node or ends with the starting node of the processed binding, the previous link is extended by the processed link.

For the practical implementation, Visual FoxPro 9.0 programming language was used as it has a set of functions for reading and writing files, as well as an extensive set of functions for working with two-dimensional arrays. The MDL file format is proprietary and well-

documented. It is a machine-readable file, where information about one element of the diagram is written on each line. At the beginning of each line, the first distinguishing character that determines the entire record structure is stated. Each line in the whole diagram is read, and based on the type of record, the line is processed. The MDL format allows writing other elements that are not used in CLD diagrams. Previously introduced parts of CLD – nodes (variables), links (relationships) and loops (feedbacks) are included in the analysis. An example of the diagram structure is apparent from Figure 1.

```
\\\---/// Sketch information - do not modify anything except names
V300  Do not put anything below this section - it will be ignored
*View 1
$192-192-192,0,Times New Roman|12||0-0-0|0-0-0|0-0-255|-1--1--1|-1--1--1|96,96,170,0
10,1,"B - Work backlog",524,57,61,25,3,131,0,0,0,0,0,0,0,0,0,0,0
10,2,"A - Available workforce",298,100,54,25,3,131,0,0,0,0,0,0,0,0,0,0,0
1,3,2,1,1,0,45,0,2,64,0,-1--1--1,|12||0-0-0,1|(401,84)|
10,4,"C - Project tasks completed",325,213,64,26,3,131,0,0,0,0,0,0,0,0,0,0,0
10,5,"E - Consumption of resources by projects",696,151,61,23,3,131,0,0,0,0,0,0,0,0,0,0,0
1,6,5,1,1,0,43,0,2,64,0,-1--1--1,|12||0-0-0,1|(631,91)|
10,7,"F - Other projects requirements",504,253,48,26,3,131,0,0,0,0,0,0,0,0,0,0,0
1,8,4,7,1,0,43,0,2,64,0,-1--1--1,|12||0-0-0,1|(382,230)|
1,9,7,5,1,0,43,0,2,64,0,-1--1--1,|12||0-0-0,1|(603,252)|
10,10,"D - Work efficiency",479,149,67,24,3,131,0,0,0,0,0,0,0,0,0,0
1,11,10,4,1,0,43,0,2,64,0,-1--1--1,|12||0-0-0,1|(425,185)|
1,12,1,10,1,0,43,0,2,64,0,-1--1--1,|12||0-0-0,1|(517,119)|
12,13,0,601,158,20,20,4,135,0,0,-1,0,0,0,0,0,0,0,0
R1
1,14,4,2,1,0,43,0,2,64,0,-1--1--1,|12||0-0-0,1|(311,175)|
///---\\\
```

Figure 1. Sketch information about the graphical representation of the diagram.

Each element appears on a separate line, and a comma separates the element's parameters. This feature is essential for further automated processing. The value distinguishes the diagram elements from the beginning of the line to the separator (comma). Nodes are determined by the number "10" at the beginning of the line, and the notation has the following structure (separated by a comma): record type (node); internal identifier; node name (text); position on X, Y axes as coordinates from the lower right edge of text rectangle variable; and the width and height of the rectangle with the text of the variable. Links Bindings are specified by the number "1", and the notation has the following structure (separated by a comma): record type (link); internal identifier; internal identifier of the node from which the link originates; internal identifier of the node to which the link is directed; line shape (circle, polygon, straight line); indication whether the link should be hidden; ASCI polarity sign (allowed characters with / S / + / for reinforcing (positive) loops and o / O / - for balancing (negative) loops. Loops are specified by the line beginning with the number "12" and the notation has the following structure (separated by a comma): record type comment (also used for loops); internal identifier; icon index associated with the comment; X, Y coordinates of the comment centre; width and height; and line shape.

The following main components and functions are used when programming the code for work with the contents of the MDL file:

```
STRCONV(FILETOSTR(lcFileName),11)
```

The program reads the entire contents of the file into the memory, where the contents can be further processed and searched.

```
OCCURS(CHR(13) + CHR(10) + "10,"
```

The text box is searched for the occurrence of a specific character set that identifies a new line in the file. From such a position, we read further to the end of the line and such a sequence of characters is for us one element in the diagram.

```
FOR lncount = 1 TO lnCountRows
    ** check if this is variable
    IF LEFT(lcTmpRows[lnCount,1],3) = "10,"
        lcTmpVars[lnCountVar,1] = lnCountVar
```

A cycle by which individual diagram objects are inserted into a two-dimensional array, for each element (row) we distinguish different attributes of the element (identifier, name, coordinates) and we write them into individual columns of the array.

```
LEFT(lcTmpRows[lnCount,1],3) = "10,"
```

If a line starts with the sequence "10" it is a "node" in the diagram. For a line starting with "1" it is a link:

```
LEFT(lcTmpRows[lnCount,1],3) = "1,"
```

We go through all the relationships and for each we write in the adjacency matrix of the polarity of the individual relationships.

```
DO CASE
    CASE INLIST(lcTmpCausL[lnCount,8],"+","s","S")
        lnMatrixVal = 1
    CASE INLIST(lcTmpCausL[lnCount,8],"-","o","O")
        lnMatrixVal = -1
```

3. Results

When constructing the algorithm, we proceed from nodes in the diagram by developing their list. We find the internal number in the original diagram (unique) for each node, assign a unique internal number for our processing, and save the name and coordinates. We also examine relationships. As with nodes, we find out the number of links and their output and input nodes. In the list, we store the link's original identifier and the indication from where and to where the link leads. We only store our internal node identifiers here. We will also find out whether the links are reinforcing (the relationship designation is "S / s / +") or balancing "O / o / -" and if the designation is missing or out of the permitted values. If so, we mark the link as incorrect. From the list of links, we process the adjacency matrix.

The link in Figure 2 leads from the node with internal number 2 to the node with internal number 1. In the adjacency matrix, there is a value of -1 in column 1 (where the link leads to) which indicates a negative link.

```
List of causal links (InternalID, OriginalID, VariableIDFrom, VariableNameFrom,
VariableIDTo, VariableNameTo, coordinates)

 1  3  2 A - Available workfo  1 B - Work backlog        CIRCLE    -    1|(401,84)|
 2  6  4 E - Consumption of r  1 B - Work backlog        CIRCLE    +    1|(631,91)|
 3  8  3 C - Project tasks co  5 F - Other projects r  CIRCLE    +    1|(382,230)|
 4  9  5 F - Other projects r  4 E - Consumption of r  CIRCLE    +    1|(603,252)|
 5 11  6 D - Work efficiency   3 C - Project tasks co  CIRCLE    +    1|(425,185)|
 6 12  1 B - Work backlog      6 D - Work efficiency   CIRCLE    +    1|(517,119)|
 7 14  3 C - Project tasks co  2 A - Available workfo  CIRCLE    +    1|(311,175)|

Matrix of causal links
  0  0  0  0  0  1
 -1  0  0  0  0  0
  0  1  0  0  1  0
  1  0  0  0  0  0
  0  0  0  1  0  0
  0  0  1  0  0  0
```

Figure 2. Link and its position in the matrix.

Furthermore, it can also be seen that in the matrix in row 1 and column 2 there is 0. If the relationship also leads back from point 1 to point 2, which is allowed, then there will also be a non-zero number at this point [1,2], expressing the link in the opposite direction.

If there is more than one link leading from one node to another, Vensim signals this state of the diagram with a warning, but such writing is not allowed in causal loop diagrams. The algorithm evaluates the number of links greater than 1 between two nodes as an error and records the absolute values of each constraint in the matrix so that the values of 1 and -1 cannot be used for information whether each relationship has a different strength. This situation would lead to a final sum of 0 and the target state could not be distinguished from the state where there is no link between the nodes. For better visual differentiation of erroneous or missing polarity of the coupling, *the algorithm uses* the value 100 instead of +/- 1.

3.1. Identification of Loops

The loops are drawn in the diagram by a circular arrow symbol, the designation R or B and a serial number. The loops are not tied to the links in the diagram in the same way as for nodes and links. The "comment" element is used to draw the loop and it has no connection to other objects in the diagram. The drawing of the loop only corresponds graphically that it is surrounded by the relationships forming the loop. We want to check that the drawn loops correspond to the relationships surrounding them and find out if a loop is missing. Therefore, we find all the loops that can exist in the diagram, regardless of whether they are drawn in the diagram or not. Then, we examine according to the position of the links and the drawn loops, whether found loops correspond to the drawn ones.

We can expect two results of this procedure, either "no difference" or an error in the diagram (a loop is completely missing or inaccurately drawn with respect to its position or polarity indication).

When looking for an algorithm, we must first choose a suitable method of searching. Let's assume that if we examine all the relationships in the diagram, we cannot omit any of the existing loops. Thus, we can proceed by considering each relationship as a starting point of the loop, and as we go through the relationships, we will gradually form groups of

relationships that follow each other in the nodes. If we go through all the relationships in this way, we get some loops. Figure 3 presents the CLD associated with a project management system. A few steps are outlined.
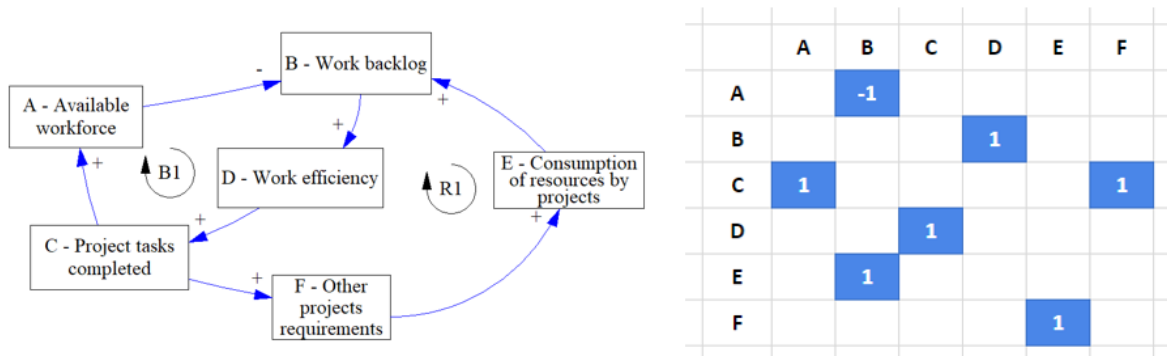

**Figure 3.** Initial state of the procedure.

Figure 4. Loops after the third step.

We start the path of the matrix line by line. They show us the direction of the relationships. We place each link we process on a new line in our list of "future loops". The first relationship leads from A to B, so we write A-B in the resulting field as the first loop's starting point. We continue through the second line, where we have another starting point of the B-D loop. We can't go to the third step at the moment, because we have found so far that we have two potential beginnings of loops A-B and B-D. We see that we can extend the relationship A-B by B-D as there is a match in the nodes, and after the connection from A-B and B-D it becomes A-B-D. At the end of the second step, we have three potential loops: A-B, B-D and A-B-D. We can proceed to the next steps which can be summarised as follows:

1. Take a new link and make it a separate line.
2. Find out whether the link can be connected via the same node name to the end or beginning of another line.
3. Check it out in this way for all rows already identified.
4. Repeat this sequence for all non-zero elements in the matrix.

In the end, we obtain a list of existing loops. From the perspective of feedback verification, there is a complication we need to deal with. Relationships are not represented by lines, but by parts of a circle or a curved line. Since the loops are drawn to the centre of

the polygon defined by the individual nodes in the CLD, it is not very likely that the drawn loops would be at the edge of one of the circles that encircle the loop. Thus, we can approximate when calculating a circular segment forming a loop by lines between two nodes. The program evaluates the found loop as correct if it occurs inside the polygon (see Figure 5). There are several algorithms to determine whether a point is inside a polygon. For implementation, Jordan Curve Theorem method (Kline, 1942) is applied. It states that if a ray projected from the point of investigation intersects the investigated polygon horizontally in an odd number of places, the investigated point lies inside the polygon. An even number of intersections means that the point lies outside the polygon (Haines, 2001). Therefore, we proceed by obtaining the equations of all edges of the polygon. We use the drawn point coordinates with the loop symbol and test whether such a line formed by two adjacent vertices intersects the Y coordinate of the point under investigation. If so, we are interested in whether such a point of intersection lies to the left or to the right of the point under investigation. We apply this procedure on all lines, and if we have an even number of intersections on one side (right or left), then the examined point is outside the polygon, in other cases, it is inside the polygon. In Figure 5, we can see that the ray from loop B1 intersects the polygon twice and is thus outside the polygon and the ray R1 intersects the polygon once (in one direction) and thus lies inside the polygon. Thus, for each loop found in the diagram, it is determined, in which detected loops, represented by the corresponding polygons, the indicators of loops are located.
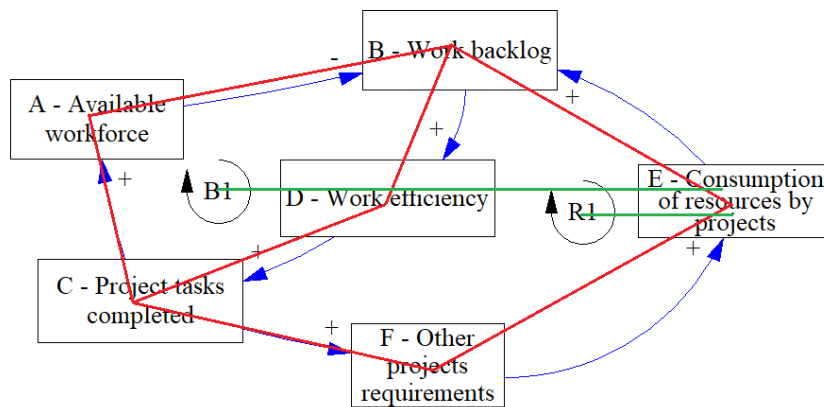
Figure 5. Matching feedback identifications with existing polygons.

3.2. Outcome Presentation

Suppose the program evaluates, based on the found loops and the polarity of the relationships that make them up, that a loop is incorrectly determined (R instead of B or vice versa). In that case, the incorrectly identified loop is marked in red, and the correct loop type is indicated (see Figure 6). If the loop is drawn with its centre outside the polygon formed by the identified loop's individual nodes, it is evaluated as missing and shown in the correct place. The method of calculating the centroid of the polygon is used in the calculation. Therefore, the original loop is marked as unpaired and is also marked in red as incorrect (see Figure 7). To fix it, move the loop to the right place.
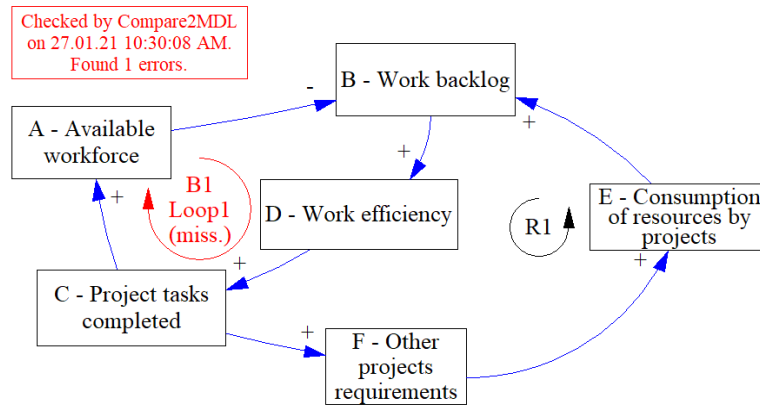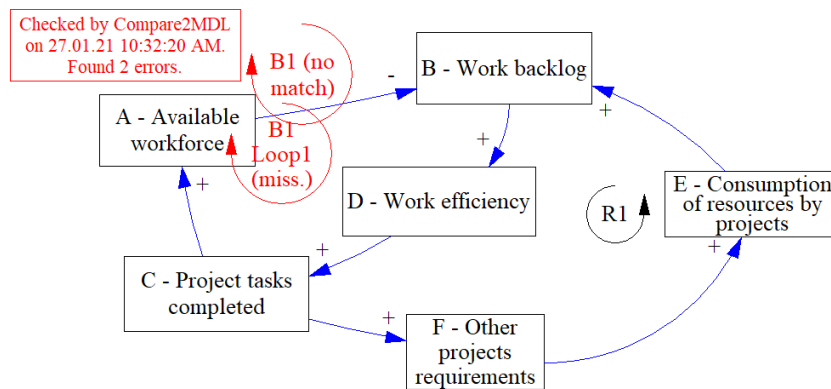
Figure 6. Missing feedback indicator.

Figure 7. Incorrect position of the feedback indicator.

4. Discussion and Conclusion

Approaches to CLD analysis can be divided into two categories. The first category can be labelled as "mechanical". We are not familiar with the CLD content or topic of interest. We can consider variables as objects associated with textual strings and located somewhere in the diagram. Links connect these variables and we consider loops as comments with round arrows. The second category can be titled as "contextual" as we understand the diagram's content and the analysis is based on working with this content. The presented method is based on the mechanical analysis as we deal with verifying whether the diagram meets all the rules necessary for its construction. These rules can be transformed into an algorithm to create a tool to check the correctness of the diagram. Vensim Software, in which the method is developed, contains interactive tools for checking the diagram. The user can mark a specific variable in the model and run the "Loops" tool to check which loops pass through the variable. This tool is suitable for continuous checking or finding the cause of a specific error in a diagram during the modelling process. The marked loops R1 and B1 are marked with the help of the element "comment" and are not connected with the loops in the diagram. In Vensim, it is impossible to check whether the marked loops correspond to those that acurrently exist in the diagram. The presented method provides the added value for this technical perspective as it extends the software functionality.

Implications of applying this method during the analysis of economic systems are apparent. Vague and ambiguous relationships between variables characterise economic systems. Bureš (2006) provides an example in the case of knowledge management and associated endeavour if its implementation in organisations. The issue with economic systems is also supported by the fact that, unlike variables and relationships that can be easily read from CLD, loops are more complex. Although CLD is associated with several weak points (Richardson, 1986), they can be used as a tool for a better understanding of structures and behaviour in areas such as industrial-technological innovation (Guo et al., 2021), value chain analysis (Muflikh et al., 2021), investment analysis (Nasirzadeh et al., 2021), productivity in manufacturing industries (Kamble & Wankhade, 2021), sustainability factors (Bureš & Racz, 2017), ambient intelligence (Mikulecký et al., 2011), or the industry growth (Eshaghpour et al., 2021). In the current form, the method provides an outcome that can be used as a valuable mean of introduction and presentation of the modelled system to specific target groups such as modelling workshop newcomers, new team members or decision- or policy-makers that need some kind of initial insight. System modellers can use it to clarify the story itself. In this way, the complexity of an analysed economic system is presented to the same target group, which enables to tame complexity as well. Hence, more appropriate and precise system comprehension can be anticipated. Hence, approved version can serve as a material for development of corresponding scenarios (Mohelská & Sokolová, 2010) or publication at social networks (Černá & Svobodová, 2017).

Not only can the method serve as a control mechanism for CLD construction for individuals, i.e. reassurance that decision making is based on realistic assumptions, but also at the corporate and inter-corporate level. It can serve as a tool for the unification of mental models in case of group modelling and decision making. Mostly, the final product of the group diagram development is a complex model with many variables and even more links included (see for instance work of Allender et al. (2014)). Not only at the corporate level, but also in multi-corporate institutions such as economic clusters where different perspectives are common, and a consensus has to be found. Evaluation of cluster effectiveness, productivity or efficiency can serve as an example when CLD can be successfully deployed (Bureš et al., 2012). Moreover, subjectivity represents a significant issue here (Jung, 2017).

There are limitations associated with the developed method. First, the method was tested by analysis of relatively simple CLDs. The question is whether the method would be able to provide outcomes of more complex CLDs in a reasonable time. Current testing reveals that the speed of the algorithm for large CLD (20+ variables) is quite unsatisfactory when the finding of all loops requires tens of minutes. However, there are methods which can be used for CLD simplification (Bureš, 2017). Reduction of complexity enables the application of the method without this type of limitation. Moreover, the development of CLD with tens of variables can be counterproductive and support confusion. Second, the application is developed in an environment which is not mainstream in the current computer science technologies. Thus, the conversion to more appropriate and robust environments is recommended. However, the method and software application can reach the target in any CLD and can identify the loops. The program can intervene in the native MDL format and

highlight errors in it. The modified file can be reopened and further processed by Vensim. The developed algorithm is finite and always reaches the goal, which represents one of the strong points.

References

Allender, S., Owen, B., Kuhlberg, J., Lowe, J., Nagorcka-Smith, P., Whelan, J., & Bell, C. (2015). A Community Based Systems Diagram of Obesity Causes. *Plos One, 10,* e0129683. https://doi.org/10.1371/journal.pone.0129683.

Binder, T., Vox, A., Belyazid, S, Haraldsson, H., & Svensson, M. (2004). Developing system dynamics models from causal loop diagrams. *Proceedints of the 22nd International Conference of the System Dynamics Society 2004* (pp. 1-21). Oxford, England.

Bureš, V. & Tučník, P. (2014). Complex agent-based models: Application of a constructivism in the economic research. *E+M Ekonomie a management, 17*(3), 152-168. https://doi.org/10.15240/tul/001/2014-3-012.

Bureš, V. (2006). Knowledge management and its implementation. In *Proceedings of the 2nd International Conference on Web Information Systems and Technologies Location (WEBIST) 2006* (pp. 115-118). INSTICC.

Bureš, V. (2017). A Method for Simplification of Complex Group Causal Loop Diagrams Based on Endogenisation, Encapsulation and Order-Oriented Reduction. *Systems, 5*(3), 46. https://doi.org/10.3390/systems5030046.

Bureš, V., & Racz, F. (2017). Identification of Sustainability Key Factors Based on Capturing Dominant Feedbacks of Behavioural Stereotypes in Socio-Economic Systems. *Systems, 5*(2), 42. https://doi.org/10.3390/systems5020042.

Bureš, V., Jašíková, V., Otčenášková, T., Kolerová, K., Zubr, V., & Marešová, P. (2012). A Comprehensive View on Evaluation of Cluster Initiatives. In J. Politis (Ed.), *Proceedings of the 8th European Conference on Management Leadership and Governance (ECMLG) 2012* (pp. 74-79), Neapolis University Pafos.

Carlin, B. I. (2009). Strategic price complexity in retail financial markets. *Journal of Financial Economics*, *91*(3), 278–287. https://doi.org/10.1016/j.jfineco.2008.05.002.

Clarke, B., Kwon, J., Swinburn, B., & Sacks, G. (2021). Understanding the dynamics of obesity prevention policy decision-making using a systems perspective: A case study of Healthy Together Victoria. *Plos One, 16*(1), e0245535. https://doi.org/10.1371/journal.pone.0245535.

Černá, S., & Svobodová, L. (2017). Internet and social networks as a support for communication in the business environment - Pilot study. In P. Jedlička, P. Marešová & I. Soukal (Eds.), *15th International Scientific Conference on Hradec Economic Days 2017* (pp. 120-126), Hradec Králové, Czech Republic.

Eshaghpour, S., Hosseini, S.H., Aghaei, S.A., & Seif, M.S. (2021). A hybrid systems approach to determine effective factors on the growth of marine industries in developing countries. *International Journal of Business and Systems Research, 15*(1), 124-142. https://doi.org/10.1504/IJBSR.2021.11175.

Forrester, J.W. (2003). Economic theory for the new millennium. *System Dynamics Review, 29*(1), 26–41.

Guo, J., Chen, M., Sun, X., Wang, Z., & Xue, J. (2021). Leveraging industrial-technological innovation to achieve sustainable development: A systems thinking perspective. *Plos One, 15*, e0242981. https://doi.org/10.1371/journal.pone.0242981.

Haines, E. (2001). Point in Polygon Strategies. In P.S. Heckbert (Ed.), *Graphics Gems* (pp. 24-46). Academic Press.

Jung, J. U. (2017). Reducing Subjectivity in the System Dynamics ModelingProcess: An Interdisciplinary Approach. *Lecture Notes in Computer Science, 10585*, 365–375.

Kamble, R., & Wankhade, L. (2021). System dynamics model to improve productivity in manufacturing industries. *International Journal of Productivity and Quality Management, 32*(1), 23-52. https://doi.org/10.1504/IJPQM.2021.112018.

Kelly, R.A., Jakeman, A.J., Barreteau, O., Borsuk, M.E., ElSawah, S., Hamilton, S.H. et al. (2013). Selecting among five common modelling approaches for integrated environmental assessment and management. *Environmental. Modelling & Software, 47*, 159–181. http://doi.org/10.1016/j.envsoft.2013.05.005.

Kline, J.R. (1942). What is the Jordan Curve Theorem? *The American Mathematical Monthly, 49*(5), 281-286.

Loyo, H.K., Batcher, C., Wile, K., Huang, P., Orenstein, D., & Milstein, B. (2013). From model to action: Using a system dynamics model of chronic disease risks to align community action. *Health Promotion Practice, 14*(1), 53–61. https://doi.org/ 10.1177/1524839910390305.

Mavrommati, G., Bithas, K., & Panayiotidis, P. (2013). Operationalising sustainability in urban coastal systems: A system dynamics analysis. *Water Research, 47*(20), 7235–7250. https://doi.org/ 10.1016/j.watres.2013.10.041

Mikulecký, P., Olševičová, K., Bureš, V., & Mls, K. (2011). Possibilities of Ambient Intelligence and Smart Environments in Educational Institutions. In N.Y. Chong, & F. Mastrogiovanni (Eds.), *Handbook of research on ambient intelligence and smart environments: Trends and perspectives* (pp. 620-639). IGI Global. https://doi.org/10.4018/978-1-61692-857-5.ch029.

Mohelská, H., & Sokolová, M. (2010). The Creation of the qualitative scenarios in the virtual three-dimensional environment Second Life. In A. Karahoca, & S. Kanbul (Eds.), *Proceedings of the 1st World Conference on Information Technology (WCIT) 2010*. Istabul, Turkey. https://doi.org/10.1016/j.procs.2010.12.053.

Muflikh, Y.N., Smith, C., & Aziz, A.A. (2021). A systematic review of the contribution of system dynamics to value chain analysis in agricultural development. *Agricultural Systems, 189*, 103044. https://doi.org/10.1016/j.agsy.2020.103044.

Nasirzadeh, H., Amin-Tahmasbi, H., & Amoozad Khalili, H. (2021). Investment analysis in privatisation of National Iranian Drilling Company using systems dynamics and BWM technique. *Energy Policy, 148*, 111963. https://doi.org/10.1016/j.enpol.2020.111963.

Oliva, R. (2004). Model structure analysis through graph theory: partition heuristics and feedback structure decomposition. *System Dynamics Review, 20*, 313-336. https:/doi.org/10.1002/sdr.298.

Radzicki M.J. (2009). System Dynamics and Its Contribution to Economics and Economic Modeling. In R. Meyers (Ed.), *Encyclopedia of Complexity and Systems Science*. Springer. https://doi.org/10.1007/978-0-387-30440-3_539

Režný, L., & Bureš, V. (2018). Adding Feedbacks and Non-Linearity to the Neoclassical Growth Model: A New Realm for System Dynamics Applications. *Systems, 6*(2), 8. https://doi.org/10.3390/systems6020008.

Richardson, G. (1986). Problems with causal-loop diagrams. *System Dynamics Review, 2*, 158-170.

Sahin, O., Salim, H., Suprun, E., Richards, R., MacAskill, S., Heilgeist, S., Rutherford, S., Stewart, R.A., & Beal, C.D. (2020). Developing a Preliminary Causal Loop Diagram for Understanding the Wicked Complexity of the COVID-19 Pandemic. *Systems, 8*, 20. https://doi.org/10.3390/systems8020020.

Schoenberg, W., Davidsen, P., & Eberlein, R. (2020). Understanding model behavior using the Loops that Matter method. *System Dynamics Review, 36*(2), 158-190. https://doi.org/10.1002/sdr.1658.