Int j simul model 22 (2023) 2, 187-198
Original scientific paper

AUTONOMOUS ENTITIES: A HYBRID MODEL AND ITS EFFECTS

Barbieri, J. P.*; Pinho, A. F.** & Montevechi, J. A. B.**
* Federal Institute of Education, Science and Technology of the South of Minas Gerais, Av. Vicente Simões, 1111, Pouso Alegre, Minas Gerais, Brazil
** Production Engineering and Management Institute, Federal University of Itajubá (UNIFEI), Av. BPS, 1303, Itajubá, Minas Gerais, Brazil
E-Mail: joao.barbieri@ifsuldeminas.edu.br, pinho@unifei.edu.br, montevechi@unifei.edu.br

Abstract
A good simulation model is one in which a real system is adequately represented. It is known that the level of accuracy of a simulation model is related to the complexity (scope and level of detail) that this model has. Given this, in the context of Modelling and Simulation (M&S), what are the effects when adding more detail to a simulation model? Are such effects the same when two different simulation approaches are combined? In order to answer these questions, this paper presents a method in which it is possible to (i) analyse the elements in a simulation model and (ii) analyse the nature of the connections among them. From the application of this method, the structure of three simulation models was represented by graphs. By analysing the graphs, the effects generated when adding a higher level of detail to the entities are perceptible. Such effects are more noticeable when comparing the simulation approaches. The use of Hybrid Simulation allows the creation of simpler and more robust connections.
(Received in August 2022, accepted in May 2023. This paper was with the authors 5 months for 1 revision.)

Key Words: Complexity, iDAV Method, Model Development, Hybrid Simulation

1. INTRODUCTION

A simulation project addresses problems involving real systems. In fact, a model fulfils the function of reproducing, in a different way, an object, a system or an idea [1]. A model is a simplified representation of reality built for a specific purpose [2]. Furthermore, in simulation models it is also possible to represent human behaviour [3].

The set of logical and causal relationships that occur in real systems is the heart of a simulation model. In a simulation model, the main characteristics of a real system are captured and reproduced virtually [3-5]. According to [1], a good simulation model needs to contain the following attributes: (1) Simple to be understood by the user; (2) Goal or purpose directed; (3) Robust, in that it does not give absurd answers; (4) Easy for the user to control and manipulate, i.e., it should be easy to communicate with; (5) Complete on important issues; (6) Adaptive, with an easy procedure for model modification or updating; (7) Evolutionary, in that it should start simply and become more complex, in conjunction with the user.

Considering the attributes listed by [1], the modeller needs to decide on the accuracy of the simulation model based on the reality. Such accuracy goes through the analysis of the complexity, i.e., the scope of the model and the level of detail. According to [2], a 100 % accurate model is utopian. One cannot know absolutely all the details of a real system. However, according to [2], the accuracy of the model increases when the scope and level of detail are increased.

In the context of Modelling and Simulation (M&S), what are the effects when adding more details to a simulation model? Are such effects the same when two different simulation approaches are combined? In order to answer these questions and discuss the level of detail in a simulation model, three simulation models will be created and analysed using a method. This method, whose name is iDAV (acronym of the terms: *identification*, *Definition*, *Analysis*, and

Verification), was created specifically to analyse the structure of a simulation model, using basic concepts present in Graph Theory (GT).

The three simulation models will represent a manufacturing line in which there is high manual labour. Due to this manual characteristic, the increase of detail in the models will happen on the degree of autonomy of each worker. Two of these models will be created using the Discrete-Event Simulation (DES) approach. The difference between these two models lies in the degree of workers' autonomy. A third model will be created using the combination of DES and Agent-Based Simulation (ABS) approaches. In this hybrid model, the workers also have autonomous characteristics. The intention of including a hybrid model lies in verifying that the effects of adding more details are the same, regardless of the type of simulation approach.

2. MATERIAL AND METHODS

2.1 The iDAV method

The method name is the acronym for the terms: *identification*, *Definition*, *Analysis*, and *Verification*. As shown in Fig. 1, the *identification* step consists of two actions: (1) **Component collection**; (2) **Connection collection**. In this step, the components in the simulation model are identified and registered. The same is done with the connections of these components; it is essential to document the relationship between each component. The result of this step is materialized in a matrix that can be built using electronic spreadsheets.

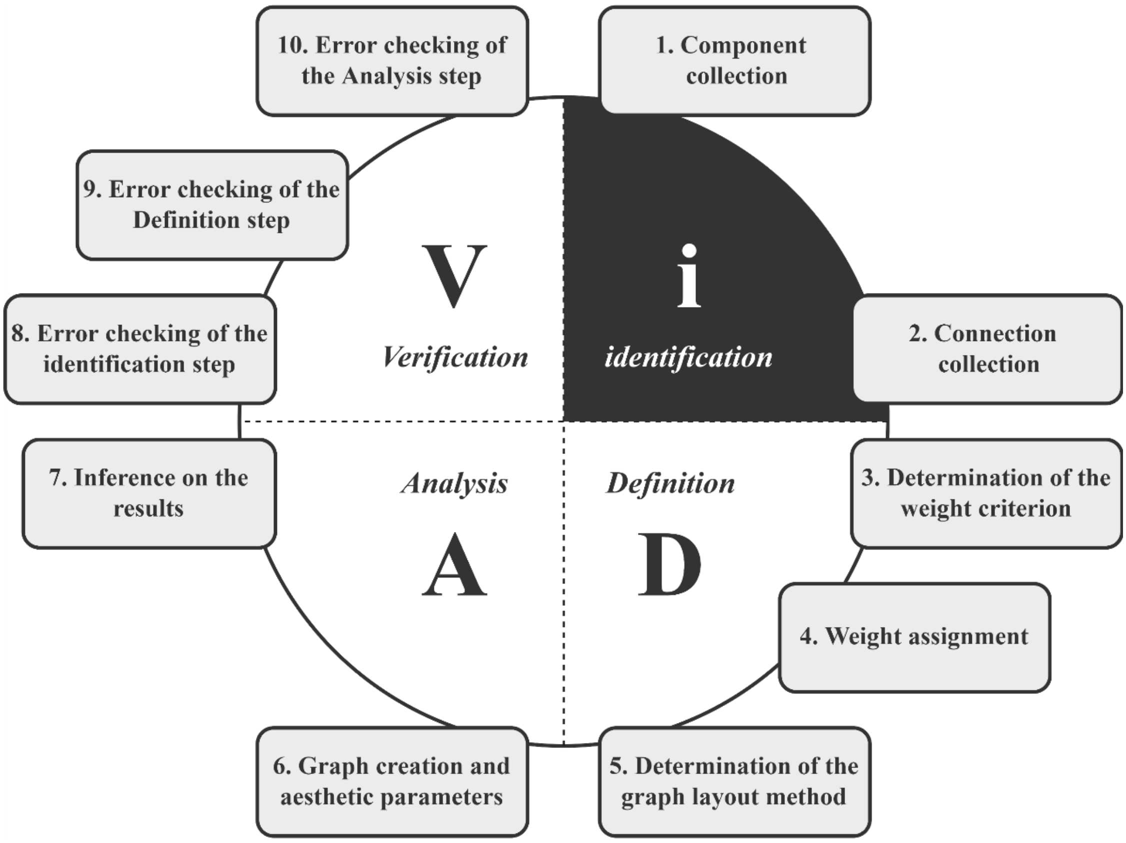

Figure 1: The iDAV method.

In the *Definition* step, there are three actions: (3) **Determination of the weight criterion**; (4) **Weight assignment**; (5) **Determination of the graph layout method**. The adoption of criteria to assign weights to connections is important, because in the simulated environment there may be connections made by *drag-and-drop* or by programming blocks. Thus, adopting a weighting criterion and applying it helps to distinguish the types of connections that exist in a simulation model. At this step, the graph layout method that will portray all components and their connections, i.e., the structure of the simulation model, is also determined.

In the *Analysis* step are two actions: (6) **Graph creation and aesthetic parameters**; (7) **Inference on the results**. There are tools that help in creating and defining the aesthetic parameters of graphs. One of these tools is the Gephi® – an open-source software used for analysing and visualizing networks. The Gephi® was chosen due to its mechanisms that facilitate communication with electronic spreadsheets. With the graph created, the action of inference about the results begins. This inference is made based on statistical concepts and methods.

The *Verification* step is the last of the iDAV method. This process is composed of three activities that are: (8) **Error checking of the identification step**; (9) **Error checking of the Definition step**; (10) **Error checking of the Analysis step**. Keeping in mind the continuous improvement of the simulation model, the *Verification* step exists to help to identify possible errors that occurred in the previous steps.

The operation of the iDAV method is cyclic, i.e., the *identification*, *Definition*, *Analysis,* and *Verification* steps can be performed multiple times. The letter "*i*", in its lowercase form, indicates the beginning of the process.

2.2 The computational models

Using the IDEF-SIM (Integrated Definition Methods – Simulation) modelling technique [6], the conceptual model was developed. Fig. 2 represents the conceptual model of an assembly line composed of three manual operations. Based on Fig. 2, the simulation models DES_Individual and DES_Autonomous were built, as well as the simulation model HYB_Autonomous. The construction of the simulation models was made by the multimethod simulation modelling tool AnyLogic®.

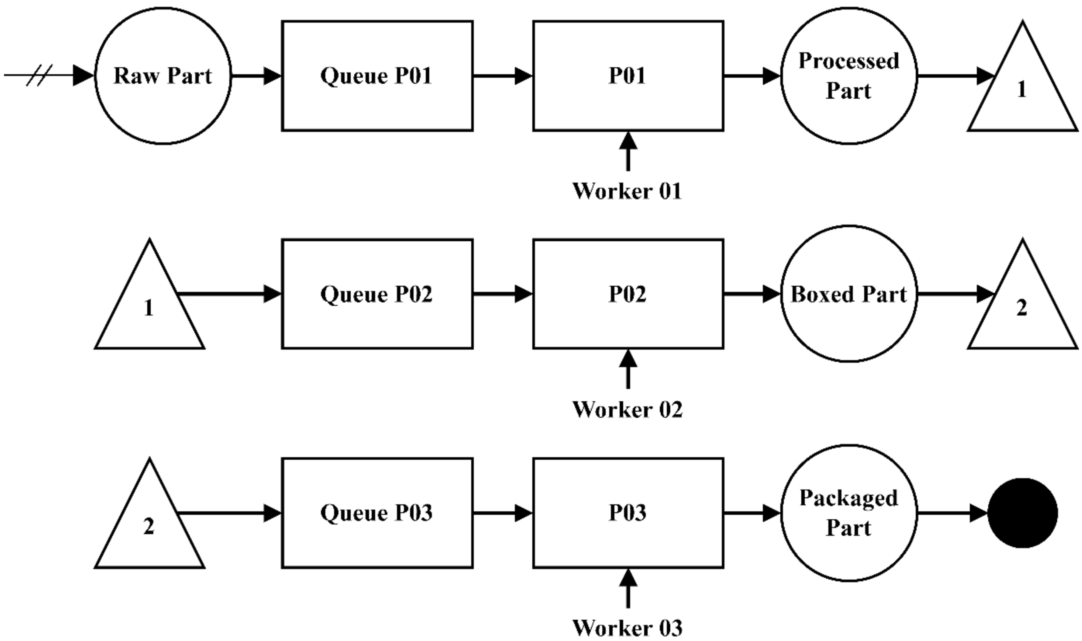

Figure 2: Conceptual model of a manufacturing line.

The probability distributions of workstations P01, P02, and P03 are, respectively, Normal (12.26, 61.33), Normal (14.03, 70.14), and Normal (10.19, 50.97). These probability distributions represent processing times in seconds. Each workstation has a queue capacity equal to 1.

The simulation models become more realistic when intelligent agents reproduce human behaviour [7]. There are four individual characteristics: behaviour, emotion, personality and perception [8]. In this paper, two characteristics will be considered: behaviour and perception. In the first simulation model, the behaviour is scripted. In the second and third simulation model, the entities have a behaviour based on the perception of variables belonging to the manufacturing system.

The human factor is represented individually. In summary, the simulation model starts with the workers in the *Resting* state. At 8:00, the workers go to the *Working* state. At 12:00, the workers leave for lunch and returns to the *Working* state at 13:00. At 17:00, the workers move to the *Resting* state. Table I shows the main characteristics of workers in the simulation models.

Table I: The main characteristics of the models.

Model	**Behaviour**	**States**
DES_Individual	Scripted	*Resting; Working; Lunching*
DES_Autonomous HYB_Autonomous	Autonomous	*Resting; Working; Lunching;* *Checking; Normal; Fast; Slow*

In the DES_Autonomous and HYB_Autonomous models, the workers have autonomous behaviour and are able to act according to the conditions present in the simulated environment. Thus, the workers have autonomous characteristics since they can change their own work rhythm during the simulation on the model. Such change occurs when each worker checks their production and compares whether this production is above, below, or within the previously established production goal. The modeller does not know, a priori, the decisions of each worker during the simulation. Therefore, in addition to the *Resting*, *Working*, and *Lunching* states, each worker has the *Checking*, *Normal*, *Fast*, and *Slow* states.

In the case of the DES_Autonomous and HYB_Autonomous models, an event that changes the work rhythm was added for each worker. These changes occur through stochastic data from seventeen values related to worker effort and work conditions [9]. Table II shows the values used in the two parameters associated with the rhythm of work.

Table II: The parameters associated with the rhythm of work.

Effort		**Conditions**	
Positive	*Negative*	*Positive*	*Negative*
0.13	0.04	0.06	0.03
0.12	0.08	0.04	0.07
0.10	0.12	0.02	-
0.08	0.17	0.00	-
0.05	-	-	-
0.02	-	-	-
0.00	-	-	-

In models where entities have greater autonomy, the production target is checked by each worker. It also checks how much has been produced. By comparing this information, each worker changes or maintains its own work rhythm. According to the parameters in Table II, this rhythm can be positive, null, or negative. Thus, the rhythm of work can be fast, normal, or slow.

3. RESULTS AND DISCUSSION

In this paper, the evolutionary attribute of a simulation model is explored in order to understand the effects of giving greater autonomy to workers, entities in the model. Starting from the idea that a computational model consists of elements with different functions that are intrinsically connected, the structure of a simulation model is based on three fundamental aspects: (i) the number of components, (ii) the connection among the components, and (iii) the essence of the connections.

The *identification* step, the first one of the iDAV method, was applied to the simulation models. Thus, 82 components were counted in the DES_Individual model, 184 components in the DES_Autonomous model, and 225 components in the HYB_Autonomous. Also, in the *identification* step, 132 connections were counted in the DES_Individual model, 276 connections in the DES_Autonomous model, and 477 connections in the HYB_Autonomous. Table III shows a summary of the data collected from the simulation models in the identification step.

Table III: The components and connections of the models.

Models	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$
DES_Individual	82	132
DES_Autonomous	184	276
HYB_Autonomous	225	477

The *Definition* step, the second one of the iDAV method, was applied to the simulation models. The Graph Theory [10, 11] served as a conceptual basis for the definition of weights and the layout of graphs. Thus, the Cyclomatic Complexity (CC) equation [12], Eq. (1), was used to analyse the essence of each connection among the components in the computational models. The CC equation is one of the most used metrics in computational measurement [13]. In practical terms, Cyclomatic Complexity measures the logical decision-making power of a system [12].

$$v\ (G) = e - n + 2(p) \tag{1}$$

$$\omega = \beta - \alpha + 2(\rho) \tag{2}$$

Based on Eq. (1), Eq. (2) was created. In this paper, ω is the number of logical actions that are made in a connection among the components; this number is the weight adopted in each connection among the components of the simulation model. The term α represents the number of components involved in a connection. The term β means the number of logical sequences among the components involved in a connection, i.e., the term e depicts the flow of information that occurs when each line of code is triggered. The term ρ represents the number of routines present in a connection.

In the simulation models, the connections have different natures. Thereby, there are connections among components in which there is no need to write lines of code. In such cases, *drag-and-drop* action is commonly adopted. However, there are also connections created from writing lines of code.

The application of Eq. (2) also occurred for the other existing connections between the components of each simulation model. The result of the *Definition* step is in the Table IV.

For *drag-and-drop* connections, ω value is equal to 1. For connections created by simple lines of code, ω value is equal to 2. For connections made by blocks of lines of code, which include routines as if-then-else, ω value is greater than 2.

Also, in the second step of the iDAV method, the layout of the graphs is defined. This definition of the graph layout was based on the works of [10, 11, 14-19]. There are several layouts for graphs, e.g., Circular layout, Radial Axis layout, Frushterman-Reingold layout, ForceAtlas layout, and Yifan Hu layout.

Table IV: The weights and number of connections.

Model	**DES_Ind.**		**DES_Aut.**		**HYB_Aut.**	
ω	β	%	β	%	β	%
1	110	83.3	110	39.9	176	37.1
2	13	9.9	97	35.1	301	62.9
5	9	6.8	9	3.3	0	0
9	0	0.0	60	21.7	0	0
Total	**132**	**100.0**	**276**	**100.0**	**477**	**100.0**

The construction of the graphs, which represent the simulation models, was based on the Gephi® software – open-source software used for analysis and visualization of networks. In this paper, the ForceAtlas and Circular layouts were adopted. Theses layouts were chosen because they highlight the parts of the simulation models and the importance of each component. The graphs produced from theses layouts help us to understand the structural aspects of the simulation models and the impacts of attributing greater autonomy to the entities.

Before the creation of the graphs that represent the simulation models, it was necessary to elaborate two matrices.

$$D = [d_{ij}] \in \Re^{m \times n} \quad (3)$$

Based on Eq. (3), the matrices were created using the Excel® tool, a spreadsheet editor produced by Microsoft. All existing components and connections in the three models are depicted, respectively, by Eqs. (4), (5), and (6). The weights of each connection were also added to the matrices. After creating these matrices, they were transferred to the Gephi® software.

$$DES\_Ind = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,82} \\ \vdots & \ddots & \vdots \\ \alpha_{82,1} & \cdots & \alpha_{82,82} \end{bmatrix} \quad (4)$$

$$DES\_Aut = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,184} \\ \vdots & \ddots & \vdots \\ \alpha_{184,1} & \cdots & \alpha_{184,184} \end{bmatrix} \quad (5)$$

$$HYB\_Aut = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,225} \\ \vdots & \ddots & \vdots \\ \alpha_{225,1} & \cdots & \alpha_{225,225} \end{bmatrix} \quad (6)$$

The *Analysis* step, the third one of the iDAV method, was performed and it was applied aesthetic parameters for the graphs. These parameters are in the Gephi® software. Figs. 3, 4, and 5 were created based on the matrices – Eqs. (4), (5), and (6).

In the ForceAtlas layout, the colours represent parts of the simulation model. The components related to 3D presentation are represented by yellow colour. The red colour depicts the workstations on which the workers are located. The workers and all their parameters and variables are represented in blue. The diameter of each component is defined by the quantity of connections and their weight. It should also be noted that each component is represented by an identification number. In Circular layout, components are ordered clockwise according to their influence on the model, from smallest to largest. Each line represents a connection. The thickness of each line corresponds to the weight assigned to it.

The third step of the iDAV method has two actions, and the last one is the inference about the results. So, the first result achieved is shown in Table III and Table IV. Looking at the DES models, the attribution of greater autonomy to the workers entities, who are present in the manufacturing line, generated an increase in the number of components and lines. The number of components and lines grew, respectively, in the order of 124.4 % and 109.1 %, i.e., the attribution of greater autonomy to entities doubled the size of the simulation model. Comparing the DES_Individual model with the HYB_Autonomous model, the number of components and lines also increased; respectively, this increase was in the order of 174.4 % and 261.4 %.

As shown in Table IV, there was also a change regard to the types of connections existing in the simulation model. For example, in the DES_Individual model, 83.3 % of the connections are *drag-and-drop*. This percentage drops to 39.9 % in the DES_Autonomous model. In the HYB_Autonomous, this percentage is 37.1 %. In addition to this drop, there is an increase in the connections with lines of code, i.e., 60.1 % of the DES_Autonomous model has more complex connections than *drag-and-drop* connections. In the HYB_Autonomous, the percentage is 62.9 %.

The increase of the components and connections, and the proportional decrease of the *drag-and-drop* connections generate impacts on the characteristics of the graphs. The thickness of the connections increased as did the diameter of the components. This can be confirmed by comparing the Figs. 3, 4, and 5.

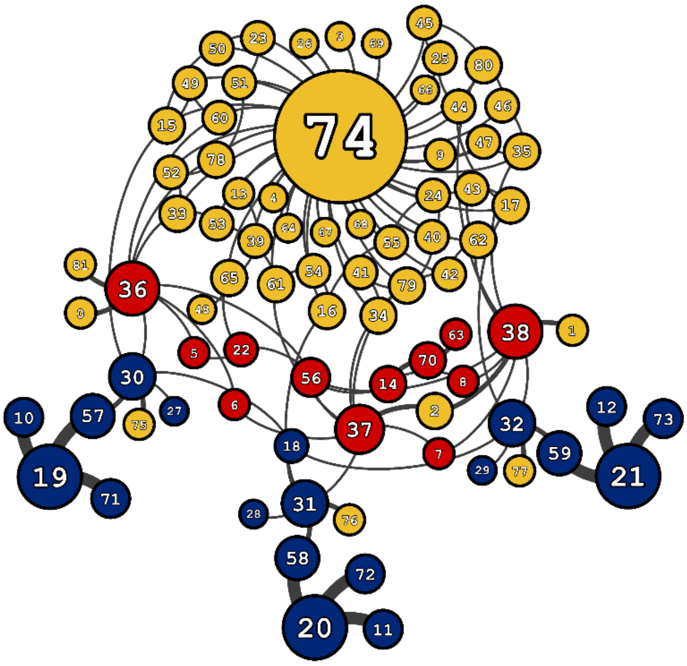

Figure 3: The structure graph of the DES_Individual model.

The simulation model must be easy for the user to control and manipulate [1]. By giving greater autonomy to entities, the number of code blocks increased. This fact can make it difficult for stakeholders to control and manipulate the variables present in the model. The lack of knowledge about the logical functioning of the computational language applied in the simulation model can also be a barrier for stakeholders.

A simulation model must be easy to be understood by the user [1]. In a simulation model, the graphical part composed of 3D components allows the stakeholders to have a clearer understanding of the workflow, actions, and behaviour of entities within the simulation model. For this reason, the 3D presentation elements were adopted in this paper. However, the effect of adding 3D components to the simulation model is notable in the graph of the DES_Individual model. If all components related to the 3D presentation were removed, the most influential components would become those related to the workers.

The components 19, 20, and 21 also standout in Fig. 3. These components bring together variables responsible for the scripted actions of the workers during the simulation of the

DES_Individual model. These actions are related to changes among *Resting*, *Working*, and *Lunching* states. These changes show up in the 3D presentation. Furthermore, the thickness of the lines connecting these components is greater, as such connections were made through a block of codes.

Still in Fig. 3, the components whose numbers are 36, 37, and 38 are also highlighted. They represent the workstations in which the workers are allocated. In addition to the components related to manufacturing line workers, there are also graphic components associated with the production and packaging of the parts. On average, there are nine components associated with the components 36, 37, and 38.

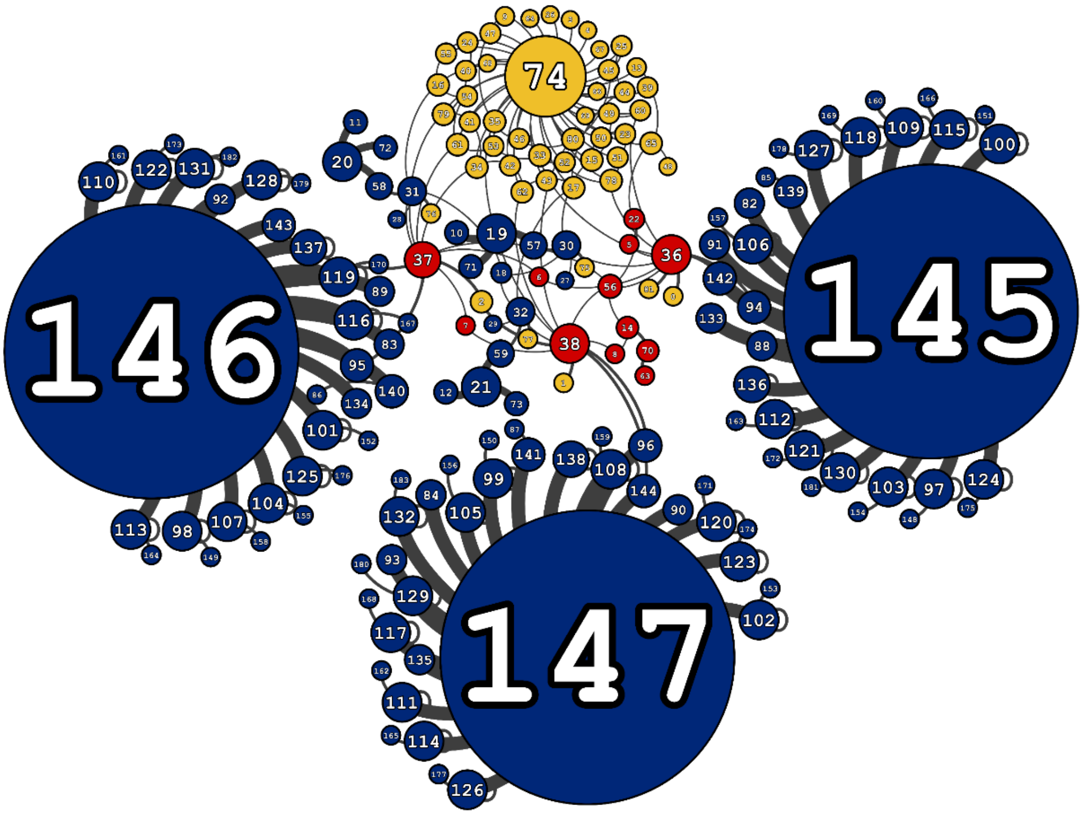

Figure 4: The structure graph of the DES_Autonomous model.

As shown in Figs. 4 and 5, the addition of variables and parameters, which allow greater decision-making capacity of entities within the simulation model, generated an increase in complexity. A good simulation model can be robust but should not produce absurd answers [1]. Thus, the three simulation models produce similar results. Even when it is added variables and parameters that give greater autonomy to the workers entities.

Applying the evolutionary principle [1] – a principle in which a simulation model should start simple and become more complex –, the *Checking*, *Normal*, *Fast*, and *Slow* states were added. In the DES_Autonomous model, the autonomy of the worker is associated with the addition of 34 components for each entity. Therefore, in total there was the addition of 102 components.

According to Fig. 4, for example, the component 147 is connected to the other 33 components through blocks of lines of code, whose logic is *if-then-else*. This type of connection increases the line thickness. This is why the components 145, 146, and 147 have a larger diameter. With the application of the iDAV method, it is evident the importance of the 145, 146, and 147 components in the DES_Autonomous model. Therefore, the component 74, that is related to all the 42 graphical components, the yellow ones, become less important in the model.

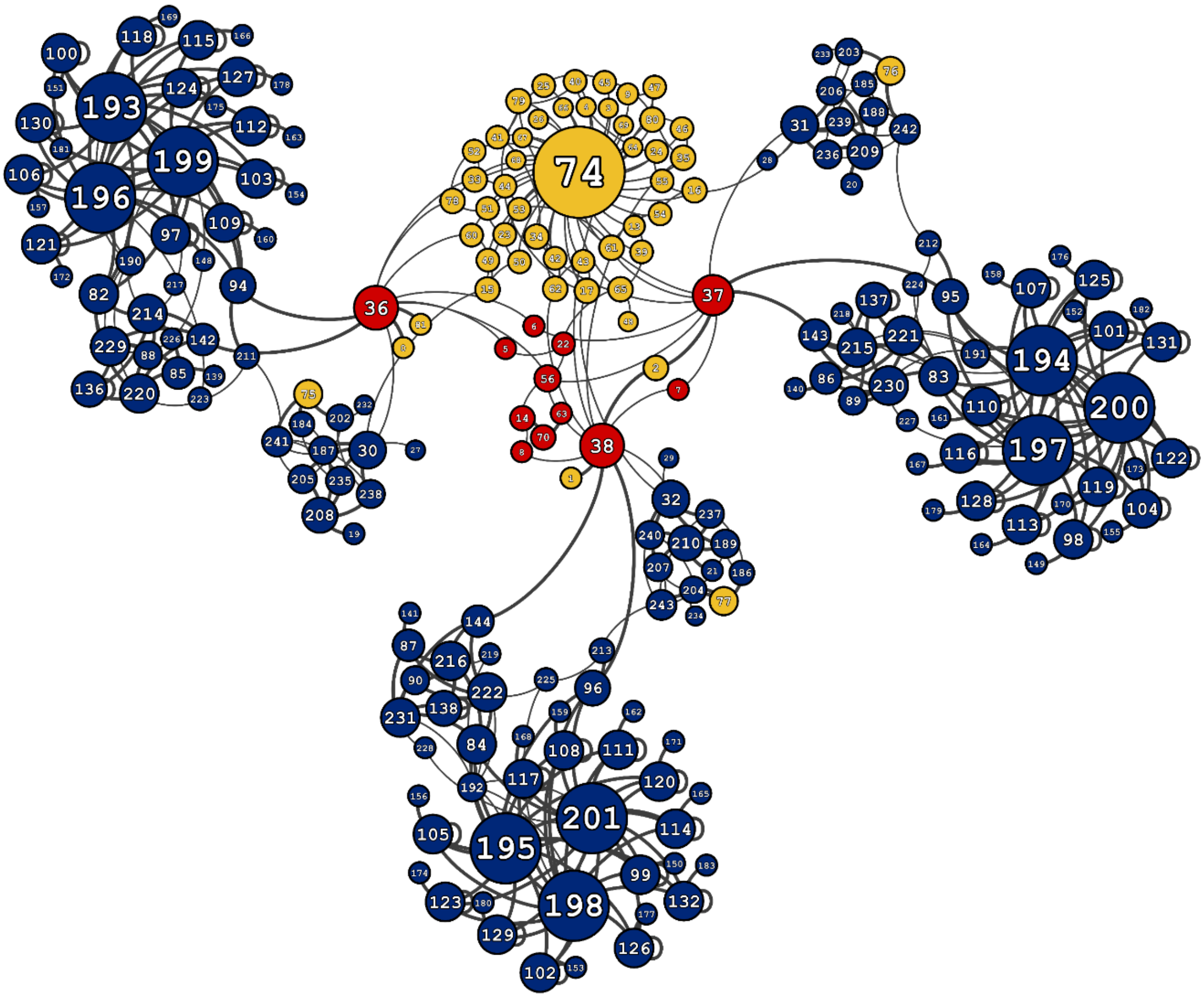

Figure 5: The structure graph of the HYB_Autonomous model.

The HYB_Autonomous model is represented by Fig. 5. Contrary to what happened in the DES_Autonomous model, the component 74 is the most influential component. This is because, despite of the apparent complexity, the HYB_Autonomous model has only two types of ω value, i.e., 1 and 2. In ABS modelling, the state diagram minimizes the use of routines, e.g., *if-then-else*. In each transition there are simpler lines of code. Despite this simplicity, the number of components and connections increase significantly as shown in Fig. 5, which reflects the Table III.

In the graph theory, making use of the handshaking lemma, each component has a value. This value was generated based on the weight and the number of connections of each component. After creating the graphs, it is possible to extract the average weighted degree (Ω) of each simulation models. Thus, the Ω value of the DES_Individual model, DES_Autonomous, and the HYB_Autonomous model are, respectively, 4.4, 9.6, and 6.9. This means that, for example, the components of the DES_Individual model have the average value of 4.4. Visually, Fig. 6 shows how the connections influence the Ω value of each simulation model.

In Fig. 6 are the essence graphs whose layout is circular. The components of each essence graph were ordered from smallest to largest. Such ordering is relevant because, with the data organized, it is easy to identify that the most complex connections are on the left and the simplest connections are on the right. Thus, through the essence graphs, it is possible to identify the simulation model in which the connections are more elaborate. This is the case of the DES_Autonomous model.

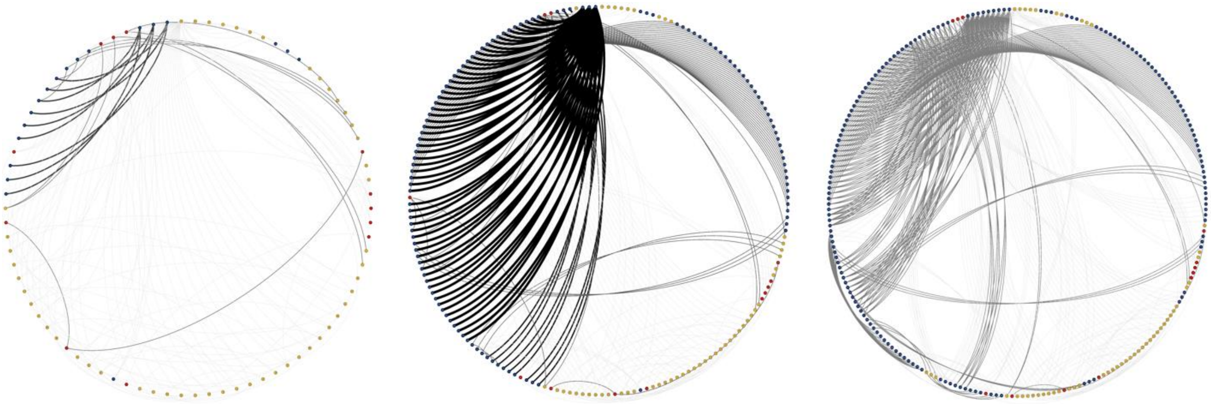

Figure 6: The essence graphs of the models.

The level of complexity of the computational models is determined by the level of detail of the real system that the modeler wants to capture. In this paper, there was no increase in the number of workstations or in the number of workers. The only increase that occurred was in the degree of decisions made by the entities. This autonomy caused an increase in the complexity of the computational models.

After finishing the third step of the iDAV method, the entire process performed so far is verified. This ensures the analysis is improved. Therefore, the actions present in the *Verification* step, the last one step of the iDAV method, helped to identify, for example, faults and programming excesses.

From the point of view of visual communication, after the *Verification* step, the graphs faithfully represent the simulation models. From this representation, the modeler is able to identify the most critical sectors and simplify the model. The graphs help the modeler to verify that [1] the simulation model has a directed objective and addresses important issues of the real system.

In fact, the *Verification* step was carried out systematically to ensure the quality of the results of the previous steps. This step can be understood as a filter in which the improvement of each model and, consequently, of each graph occurs. After the *Verification* step, the understanding of the details of each model was improved. This better understanding [1] is very important for potential stakeholders.

In manufacturing systems, the human factor is an important element. The human labour is [20] an essential element in modern manufacturing systems and, therefore, it must be reliably expressed in the modelling process of such systems. The human behaviour is a theme of modelling and simulation [21-23] and it is still a traditional challenge for the simulation area itself and its application in the business areas [24]. Indeed, identifying and conceptualizing the structure of human behaviours and interactions is a common problem in modelling complex systems involving human factor.

In Discrete-Event Simulation, the inclusion of human performance generates an opportunity to know about the impact and importance of the human factor in the system [25]. But it is a challenge to perform a more detailed modelling of the human factor through the Discrete-Event Simulation. Despite being a challenge, it is possible to attribute some human characteristics to entities in a Discrete-Event Simulation model. However, such inclusion generates effects as shown through the graphs.

The simulation model must be easy to be controlled and manipulated by the user [1]. Furthermore, the simulation model must also be adaptive, with easy procedures for modification or update. These characteristics guided the development of the simulation models in this paper. Dividing the simulation model in three sectors, and building graphs that emphasize this division, create the opportunity for improvement.

4. CONCLUSIONS

This paper sought to answer the two questions. The first one was: in terms of modelling, what are the effects when adding more details to a simulation model? The effects are: (i) Increase in the number of components and connections related to workers; (ii) Increased connections with lines of code; (iii) Increased variability of the results generated by the model. The increase in the number of components and connections was expected. However, the simulation model was doubled in size, and this was not expected. Adding autonomy to entities generated more *if-then-else* connections. Proportionally, almost 60 % of the connections in the models with autonomous entities were made through lines of code. This fact was not expected either. The second question was: Are such effects the same when two different simulation approaches are combined? The answer is negative. As shown through the essence graphs, the combination of DES and ABS approaches simplifies modelling. In the hybrid approach, despite the increase in the number of components due to the state diagram, the connection between the components is simpler. It is characteristic of the ABS approach to make use of state diagrams in order to explain the logical and causal relationships between the possible actions of an entity. This characteristic associated with the DES approach generated simplifications in the model connections. The iDAV method was created to analyse the structure of simulation models. Such method makes it easy to identify the importance of components and their connections in a simulation model. Furthermore, the iDAV method also helps to understand the structural part of the model and the areas that can be improved. For future papers, we suggest applying the iDAV method to models built with other simulation approaches, e.g., Dynamic Systems Simulation. We also suggest choosing other graph layouts to specifically analyse other aspects such as the nature of connections.

ACKNOWLEDGEMENT

The authors of this paper would like to express their gratitude to UNIFEI, IFSULDEMINAS, CAPES, CNPq, and FAPEMIG for their support to this research.

REFERENCES

[1] Shannon, R. E. (1975). *Systems Simulation: The Art and Science*, Prentice-Hall, Englewood Cliffs
[2] Robinson, S. (2008). Conceptual modelling for simulation Part I: definition and requirements, *Journal of the Operational Research Society*, Vol. 59, No. 3, 278-290, doi:10.1057/palgrave.jors.2602368
[3] Hlupic, V.; De Vreede, G.-J. (2005). Business process modelling using discrete-event simulation: current opportunities and future challenges, *International Journal of Simulation and Process Modelling*, Vol. 1, No. 1-2, 72-81, doi:10.1504/IJSPM.2005.007115
[4] Sargent, R. G. (2014). Verifying and validating simulation models, *Proceedings of the 2014 Winter Simulation Conference*, 118-131
[5] Brailsford, S. C.; Eldabi, T.; Kunc, M.; Mustafee, N.; Osorio, A. F. (2019). Hybrid simulation modelling in operational research: a state-of-the-art review, *European Journal of Operational Research*, Vol. 278, No. 3, 721-737, doi:10.1016/j.ejor.2018.10.025
[6] Montevechi, J. A. B.; Leal, F.; Pinho, A. F.; Costa, R. F. S.; Oliveira, M. L. M.; Silva, A. L. F. (2010). Conceptual modeling in simulation projects by mean adapted IDEF: an application in a Brazilian tech company, *Proceedings of the 2010 Winter Simulation Conference*, 1624-1635
[7] Bruzzone, A. G.; Tremori, A.; Tarone, F.; Madeo, F. (2011). Intelligent agents driving computer generated forces for simulating human behaviour in urban riots, *International Journal of Simulation and Process Modelling*, Vol. 6, No. 4, 308-316, doi:10.1504/IJSPM.2011.048011
[8] Du, X.; He, W.; Mao, Y.; Rao, Y. (2018). Crowd evacuation simulation based on emotion contagion, *International Journal of Simulation and Process Modelling*, Vol. 13, No. 1, 43-56, doi:10.1504/IJSPM.2018.090270

[9] Barnes, R. M. (1991). *Motion and Time Study: Design and Measurement of Work*, 7th edition, John Wiley & Sons, New York

[10] Godehardt, E. (1988). *Graphs as Structural Models: The Application of Graphs and Multigraphs in Cluster Analysis*, book series *Advances in System Analysis*, Vol. 4, Vieweg, Braunschweig

[11] Wallis, W. D. (2007). *A Beginner's Guide to Graph Theory*, 2nd edition, Birkhäuser, Boston

[12] McCabe, T. J. (1976). A complexity measure, *IEEE Transactions on Software Engineering*, Vol. SE-2, No. 4, 308-320, doi:10.1109/TSE.1976.233837

[13] Polančič, G.; Cegnar, B. (2017). Complexity metrics for process models – a systematic literature review, *Computer Standards & Interfaces*, Vol. 51, 104-117, doi:10.1016/j.csi.2016.12.003

[14] Di Battista, G.; Eades, P.; Tamassia, R.; Tollis, I. G. (1994). Algorithms for drawing graphs: an annotated bibliography, *Computational Geometry*, Vol. 4, No. 5, 235-282, doi:10.1016/0925-7721(94)00014-X

[15] Herman, I.; Melançon, G.; Marshall, M. S. (2000). Graph visualization and navigation in information visualization: a survey, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, 24-43, doi:10.1109/2945.841119

[16] Kaufmann, M.; Wagner, D. (Eds.) (2001). *Drawing Graphs: Methods and Models* (Lecture Notes in Computer Science), Springer, Berlin

[17] Koren, Y. (2005). Drawing graphs by eigenvectors: theory and practice, *Computers & Mathematics with Applications*, Vol. 49, No. 11-12, 1867-1888, doi:10.1016/j.camwa.2004.08.015

[18] Jacomy, M.; Venturini, T.; Heymann, S.; Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software, *PLOS ONE*, Vol. 9, No. 6, Paper e98679, 12 pages, doi:10.1371/journal.pone.0098679

[19] Tamassia, R. (Ed.) (2014). *Handbook of Graph Drawing and Visualization*, CRC Press, Boca Raton

[20] Digiesi, S.; Kock, A. A. A.; Mummolo, G.; Rooda, J. E. (2009). The effect of dynamic worker behavior on flow line performance, *International Journal of Production Economics*, Vol. 120, No. 2, 368-377, doi:10.1016/j.ijpe.2008.12.012

[21] Oliveira, M. L. M.; Montevechi, J. A. B.; Pinho, A. F.; Miranda, R. C. (2017). Using hybrid simulation to represent the human factor in production systems, *International Journal of Simulation Modelling*, Vol. 16, No. 2, 263-274, doi:10.2507/IJSIMM16(2)7.378

[22] Vilela, F. F.; Leal, F.; Montevechi, J. A. B.; Piedade, D. D. C. (2020). Effect of human factor performance on the productivity of a manual assembly line, *International Journal of Simulation Modelling*, Vol. 19, No. 3, 365-374, doi:10.2507/IJSIMM19-3-508

[23] Guo, X. Y.; Zeng, Z.; Li, M. X.; Fu, S. (2022). Simulation of aircraft cabin evacuation strategy based on exit flow equilibrium, *International Journal of Simulation Modelling*, Vol. 21, No. 2, 261-272, doi:10.2507/IJSIMM21-2-601

[24] Bruzzone, A. G.; Briano, E.; Bocca, E.; Massei, M. (2007). Evaluation of the impact of different human factor models on industrial and business processes, *Simulation Modelling Practice and Theory*, Vol. 15, No. 2, 199-218, doi:10.1016/j.simpat.2006.09.020

[25] Baines, T.; Mason, S.; Siebers, P.-O.; Ladbrook, J. (2004). Humans: the missing link in manufacturing simulation?, *Simulation Modelling Practice and Theory*, Vol. 12, No. 7-8, 515-526, doi:10.1016/S1569-190X(03)00094-7