Proceedings Paper:
Tsialiamanis, G.P., Wagg, D.J., Antoniadou, I. et al. (2020) An ontological approach to structural health monitoring. In: Dilworth, B. and Mains, M., (eds.) Topics in Modal Analysis & Testing, Volume 8 : Proceedings of the 38th IMAC, A Conference and Exposition on Structural Dynamics 2020. 38th IMAC, A Conference and Exposition on Structural Dynamics 2020, 10-13 Feb 2020, Houston, TX, USA. Springer International Publishing, pp. 51-59. ISBN: 9783030477165. ISSN: 2191-5644. EISSN: 2191-5652.

https://doi.org/10.1007/978-3-030-47717-2_6

An ontological approach to structural health monitoring

G. Tsialiamanis, D.J. Wagg, I. Antoniadou, & K. Worden
Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield
Mappin Street, Sheffield S1 3JD, UK

Abstract

In the current work, an ontological framework for structural health monitoring (SHM) is discussed. Ontologies are used in disciplines like knowledge engineering and natural language processing, but their structure and goals also fit the purposes of SHM. In SHM projects – as in all projects – many problems arise during knowledge sharing and application. Ontologies can deal with these problems and at the same time have more benefits for SHM processes, as their modularity may assist in extending and transferring knowledge. An SHM-specific ontology is constructed here and described; It contains many objects that can be used in the procedure of monitoring structures. The ontology can also be used as a database to store data acquired, but also serves as a knowledge-base for the current discipline's algorithms and methods. Further, having close connections to object-oriented programming, ontologies straightforwardly facilitate software development and reusability of their components. Certainly, the ontology can be used to save time during the application of SHM, but also can be applied to improve performance of existing methods, by finding within the ontology the best algorithm to fit the purpose of each method.

Key words: Structural health monitoring (SHM), ontologies, database, knowledge-base, knowledge engineering.

1 Introduction

In modern societies, everyday activities are becoming increasingly dependent on structural and mechanical systems. These systems have a design life which is heavily dependent on the external conditions that they will be subjected to throughout their operation. Since it is critical that they survive their design while remaining operational and safe, a framework to ensure both operability and safety is required. With this purpose in mind, *structural health monitoring* (SHM) can be employed. SHM refers to the process of implementing a damage detection strategy for aerospace, civil or mechanical engineering infrastructure [1]. Application of SHM to systems can be partitioned into the following steps [2]: (1) observation of the system during its operation, (2) data acquisition from the system and, (3) extraction of features that are sensitive to damage and determine the current state of system's health. The steps of SHM can be implemented in many ways. More specifically, the final two steps of feature extraction and current state evaluation have been implemented with various methods that come from different scientific disciplines like signal processing, machine learning, physical modelling, etc. All these methods interact with each other and have their advantages and disadvantages. Being motivated by this and by the fact that in many projects problems arise from poor communication between various project members and difficulty in knowledge sharing [3], this paper is proposing a connecting framework for these components and for sharing knowledge within the SHM process.

The proposed framework here is an *ontological* one. Ontologies are used in many fields including computer science, semantics and natural language processing. The Ontology's purpose of sharing knowledge, developing software modules and interoperability between different projects fits the needs of SHM and can be exploited to improve performance of such applications. In the current work, an SHM ontology is constructed and discussed in the context of using it to better understand its component functions: using it as a database, to apply SHM techniques more efficiently and to implement software according to the ontology.

Although applicable to SHM in a broad sense, an ontology could be particularly useful in a Population-based SHM (PBSHM) setting [4, 5, 6, 7, 8], where the goal is to develop general inference tools across a population. Here an ontology would identify useful, and perhaps overlooked, connections between objects such as Irreducible Element (IE) models and Atribbuted Graph (AG) representations of structures [5, 6] and appropriate knowledge transfer methods like transfer learning [7] and 'forms' [7, 4]. This may provide benefits in highlighting appropriate methods for each data source such that destructive phenomenon like negative transfer in transfer learning are avoided.

A similar approach has been outlined in previous work [9] to exploit the advantages of using ontologies in the scope of verification and validation (V&V) and system identification, fields that also have many interacting components.

2 Components of the Ontology

Ontologies have many definitions. The one that is preferred here is given in [10]: "An ontology is a specification of a conceptualization". This means that the ontology is a description of knowledge on a specific domain that can be helpful in sharing and explaining, storing and reusing/transferring it to similar projects and domains.

Ontology construction is usually carried out using ontology languages like OWL [11]. To facilitate ontology construction, software with user interfaces exist e.g. Protege [12] and GATE. The specific ontology construction software used herein was Protege, developed at Stanford University in collaboration with the University of Manchester; it was chosen for its convenience in editing the ontology and defining the components. In Figure 1 the interface of Protege is shown, on the left various pieces of the ontology are listed and on the right a specific class is being edited. Protege has many capabilities and flexibility in editing and creating ontologies like defining variable types according to one's needs, automatic clustering of objects following axioms, etc.

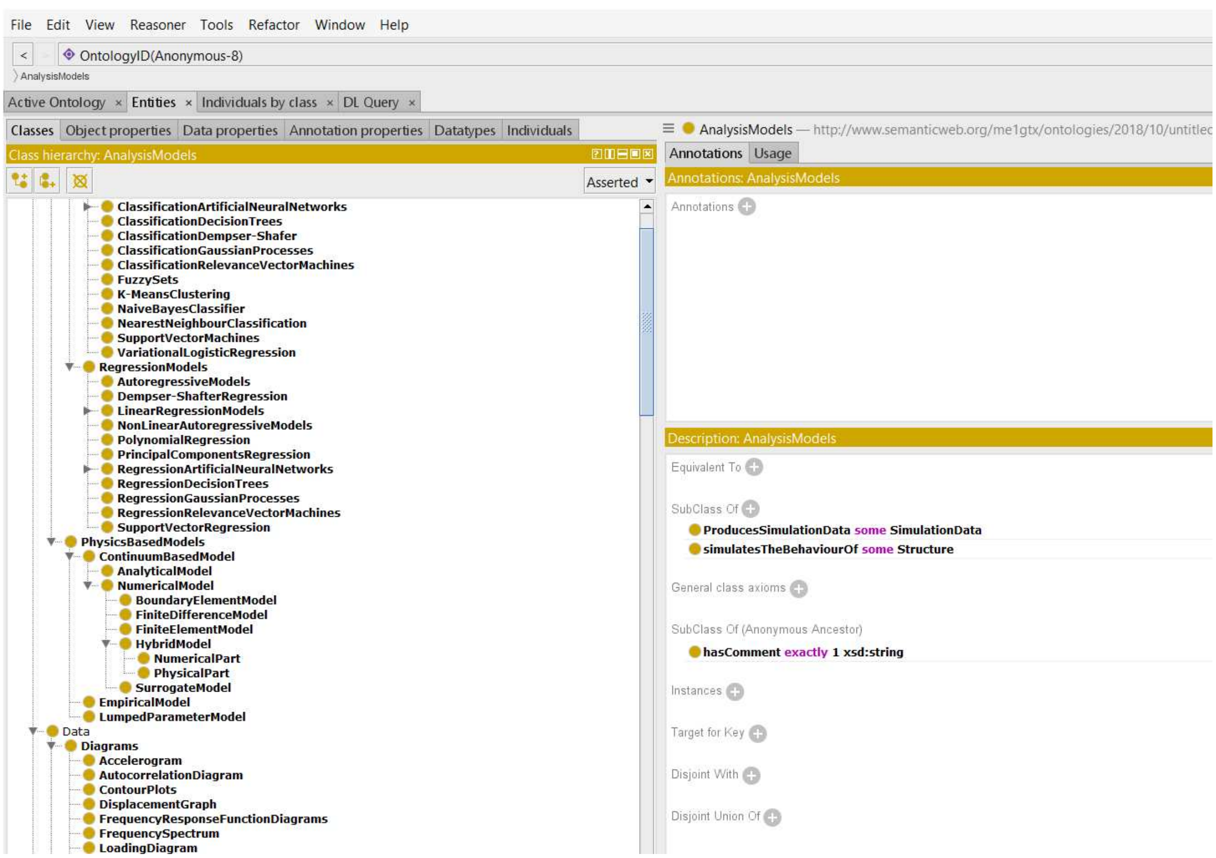

Figure 1: Protege ontology editor.

The procedure of constructing the ontology is executed through construction of its components. The components constituting an ontology are:

1. **Individuals**: are members of the ontology. Individuals are instances of the classes that they belong to and have corresponding characteristics.

2. **Classes**: are sets that include instances with similar characteristics. Classes are partitioned in subclasses and all the instances belonging to them share theproperties of the superclass.

3. **Connections**: are the relationships connecting classes. The defined connections show the way that two or more classes may be related. The name of the connection isthe semantic definition of the interaction between connected classes.

4. **Attributes**: are characteristics of individuals; they refer to a quantitative characteristic of an individual and their values are not always the same between individuals of the same class.

5. **Properties**: are the connections between individuals and attributes. A property of an individual reveals an attribute that may differentiate it from similar individuals. As with the connections, the property name defines the semantics of the individual-attribute relationship.

6. **Annotations**: are strings including definitions of objects. They can be thought as an attribute being a string that describes the functionality of an individual or a class.

Having defined an ontology through the components above, a hierarchy of classes and subclasses is created. This is also called a taxonomy and is the result of the ontology having connected different classes only through connections defining the relationship of subclass and superclass. Following this path, an initial tree structure was accomplished. This is the fundamental structure of the in-hand ontology. Connections linking classes with different relationships were defined thereafter. In the following sections, both the fundamental structure and the complete structure will be described.

3 Main Ontology Structure

In order to facilitate the shaping of the ontology, a taxonomy was initially defined comprising of the most important superclasses of instances used in SHM. These superclasses were chosen to be:

1. analysis models;
2. data;
3. data processing;
4. physical parts;
5. SHM methods.

The first four classes contain algorithms and data types that are used in many disciplines and can be easily transferred into other ontologies. The final class contains the SHM methods that may be used in the procedure of monitoring structures and must be connected with instances in other classes, since these methods use instances of all other superclasses. One can think of the final class as the main goal and connecting component of the ontology.

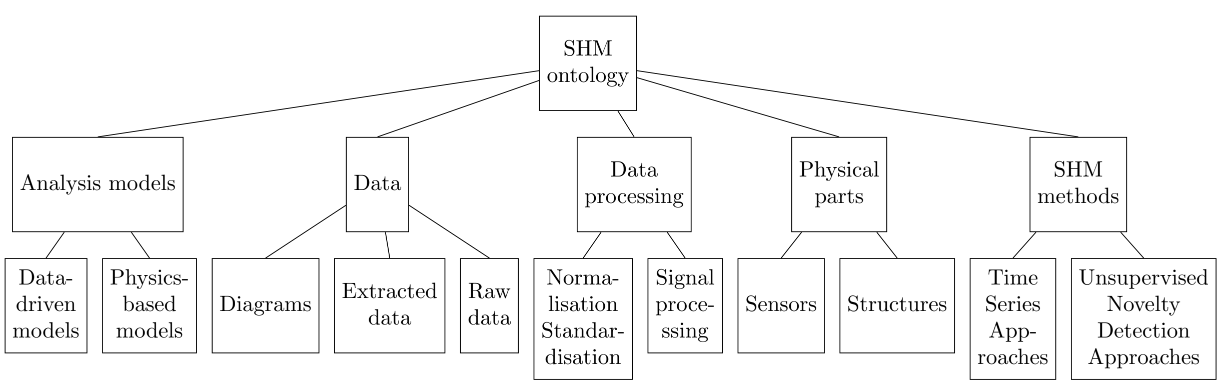

Figure 2: The five superclasses and some of their subclasses.

To further explain the content of the aforementioned classes, their description and some of their subclasses will be presented subsequently. The "analysis models" class contains anything that is used to analyse a structure. Since many types of models are used in various structural health monitoring techniques, like monitoring the modal assurance criterion (MAC) [13] or neural networks [14] to classify/localise damage. The first partition is into the classes of "physical models" and "data-driven models". "Data-driven models" is a class that mainly contains machine learning techniques. These are models that simulate physics, but infer their output using existing data from structures. These models are partitioned mainly into *regression* models and *classification* models. The first of these classes is composed of models predicting values of variables (e.g. natural frequencies, displacements, etc.) and are more commonly used in simulating the behaviour of structures, whilst the second one is often used to discriminate data corresponding to damaged or undamaged states of the structure. Some of the included models are neural networks [15], K-means clustering [16], support vector machines [17], autoassociative neural networks [18] etc. Physics-based models are constructed by studying and explaining the physics of problems. Two major subclasses of this class are the analytical and numerical models. Numerical models contain finite element models, surrogate models, lumped mass models etc; they are the most often-used ones, since they can be simulated using computers. On the other hand, the analytical models are solved using calculations and are not perhaps as common as the numerical ones, since engineering problems often preclude analytical solutions; however, they are still included in the ontology for completeness.

The "Data" superclass contains all the data that are acquired from structures. The data are partitioned in classes according to their type. For example, acceleration, displacement and velocity data, which belong to the time domain, but also data in the frequency domain. Furthermore, the data are also separated into data from sensors ("raw data"), data from processing the sensor data ("processed data") and data from any models ("simulation data"). "Data" is probably the class that can be more easily transferred into ontologies describing other domains, since most processes nowadays produce data and their analyses are based on data. Additionally, data can be transferred from one SHM application to another. Analysing data from a structure and making inference about them, can help in understanding the behaviour of similar structures or materials used in another SHM application. Some subclasses included in this class are "accelerograms", "contour plots", "mode shapes", "displacement simulation data", "displacement sensor data", etc.

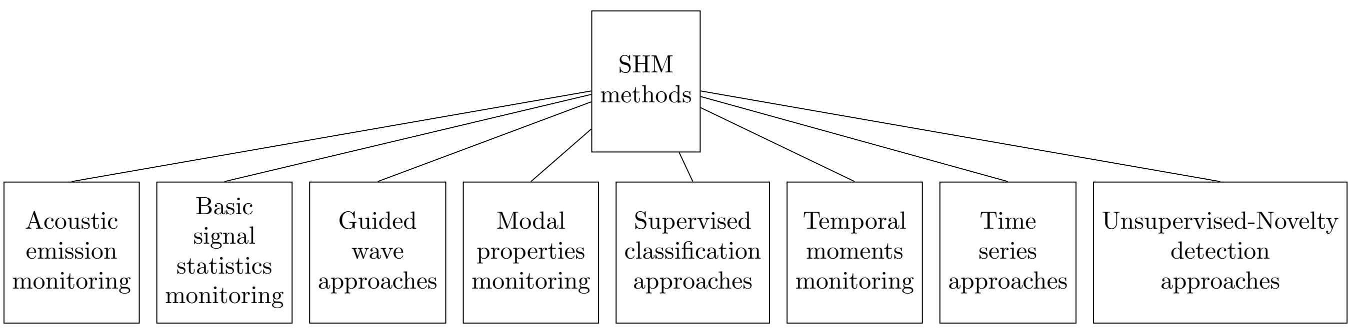

Figure 3: SHM "methods" class.

The next superclass is related to methods that are used to process the data. These methods mainly belong to the discipline of signal processing, from the simple and ubiquitous Fourier transform [19] and signal statistics extraction, to more complicated methods like wavelet decomposition of signals [20]. These methods are vital for SHM, because they may reveal unseen features of the signals that are sensitive to damage. It is quite common that a damaged state can be spotted by observing a spectrum (which is produced by performing a Fourier transform on the acceleration signal of a sensor) rather than the acceleration time-history itself. Methods included in this class are: "signal smoothing algorithms", "Hilbert transform", "principal component analysis", etc.

The fourth superclass is the "physical parts" that the all methods refer to. This class simply contains instances of the structures that are monitored and the sensors placed on them. The instances are just objects referring to an existing structure. The structures belonging to this class can be partitioned in substructures for a more detailed and modular representation of a greater object. For example a wind turbine can be partitioned into its blades and the tower and even deeper, the blade can be partitioned into the cell and the stiffeners existing inside. This class serves the purpose of registering all monitored physical objects but also separating them into classes to facilitate search for data from a specific structure or type of structures.

Last but not least, is the class containing all methods that are used in SHM. This class is the most important one, as it uses components from every other class and combines them into methods that monitor a structure. Some of the methods here are trivial, such as monitoring the maximum value of a sensor signal; but there are also much more sophisticated methods, like monitoring the modal force error on a structure, that require data from the "data" class, finite element models from the corresponding class and also data processing methods. All methods included refer to a specific structure, use data from sensors, process them with a processing method and make inference about the current situation of the structure according to a model from the class "analysis models".

Many types of SHM methods are included in the current ontology. They are partitioned into subclasses according to the type of data they use and the types of algorithms/methods used to make inferences. For example, there are the "acoustic emission monitoring" [21] and "guided wave approaches" [22] whose data come from receivers trying to "hear" cracks in materials and ultrasound receivers correspondingly. Other classes of methods are the "novelty detection methods", which are unsupervised ones [23], trying to identify changes in the behaviour of structures, having as inputs only normal condition/undamaged state data. Major classes of this superclass can be seen in Figure 3.

4 Connections

Having defined a tree structure for the ontology and included all the components that are used in SHM procedures, further connections have to be defined in order to specify more analytically the interaction of these components. This is achieved by adding connections in the ontology. The name of the connection, as mentioned above, is the semantic definition of the connection; it explains how two objects are connected within the ontology for the purpose of SHM. So far the defined connections are only of type "is a subclass of". For example, "data-driven models" is a subclass of "analysis models".

Connections are added to explain the functionality of components. Data-driven models use data so they are connected to the type of data that they use with a connection named "uses data of type". Furthermore, all types of models produce data and are connected to them with a connection called "produces data". Following this logic, all the components gradually get connected to other ones and form a more complicated diagram that represents the existing knowledge about SHM. The result can be seen in Figure 4. It is clear that adding these connections to the architecture makes it much more complicated than the tree that was the initial structure (part of it is shown in Figure 2).

As expected, the "SHM methods" class is connected to most of the other classes. All methods have to use one or more analysis models, they all refer to a structure, use sensors, exploit data taken from sensors and process the data using a data processing method. Finally, each method outputs a result concerning the current state of a structure. This explains both why this class can be considered the goal of the ontology and the connective factor of it.

An important aspect of ontology connections is the possibility and generation of inverse connections. Inverse or reverse connections are defined as the inverse property of a direct connection of classes A and B. For example if class A is a subclass of class B, then class B is a superclass of class A. This type of connection integrates the semantics of the ontology and defines a more detailed description of the components; they can be really useful when a new component is to be added into the ontology. In that case, connections are defined linking it to existing elements and through reverse connections the current elements' functionality is updated. As an example, one may consider the addition of a new SHM method in the ontology. Machine learning method A is used by the new object and therefore, a direct connection with name "uses method" connects the newcomer and machine learning method A. The inverse property in this case is named "is used by method" and extends the definition of machine learning method A by connecting it to the new SHM method. In Figure 5 a novelty detection SHM data can be seen as the orange circle. It is connected with direct connections (white arrows) to other classes (yellow ellipses) and an attribute (red ellipse). Each direct connection has a corresponding inverse one (red arrows) that defines how the classes in yellow circles affect the class under examination.

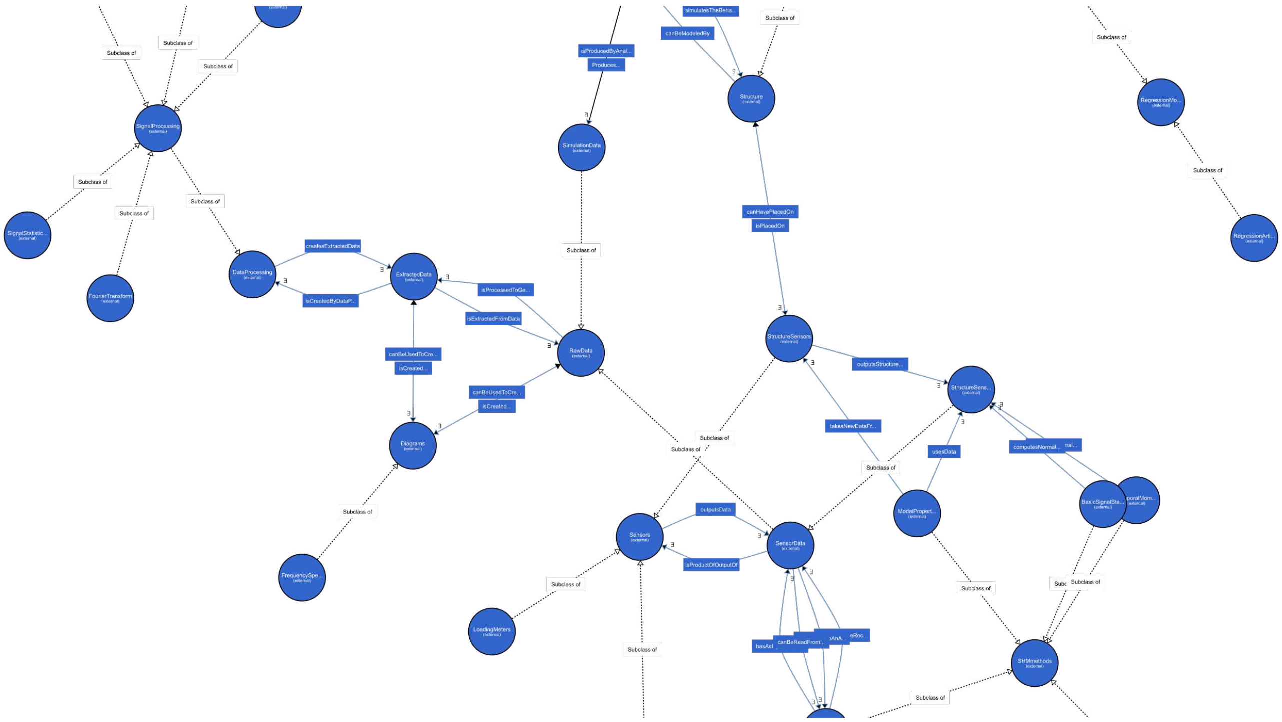

Figure 4: Overview of the SHM ontology.

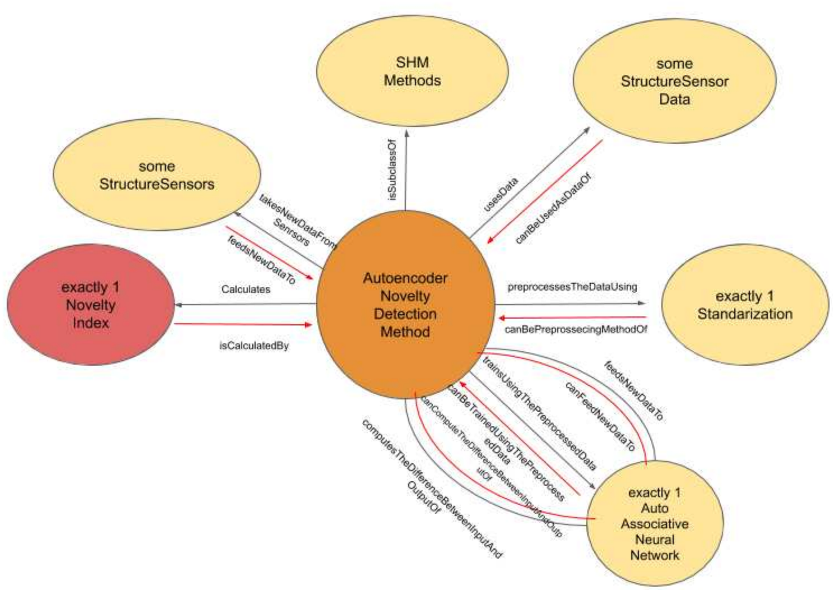

Figure 5: An SHM novelty detection method.

5 Aspects of the Ontology

Having constructed the ontology, some of its uses or aspects will be examined. The first and most straightforward one is using the ontology to share knowledge. It is clear that the fabricated network is really effective in explaining the functionality of its components and their interaction. Apart from the annotations explaining each object's function, one is lead by the connections to other components directly connected to the one that is of interest, and learns their functionality too, gaining further intuition in the methods. Furthermore, within a project, collecting knowledge in the ontology and sharing it facilitates everyone's understanding in each other's part. For example, if a method is developed by a member of a group and it gets included in the ontology by defining connections with existing pieces of knowledge, methods and structures, it gets automatically explained to other members by observing it as a part of the ontology. This can solve difficulties in communication, misunderstandings and boost a group's productivity.

Further to helping groups in understanding and communicating, ontologies can be used in the same scope to increase the efficiency of SHM applications, to automate them and expand existing knowledge. It is clear that the ontology is quite modular. Using a regression algorithm for an SHM method means that another one can be used and maybe different algorithms can yield better results than the initial one. A method's connection to a specific algorithm almost certainly means that it can be connected to algorithms in the same superclass as the existing one. Following this scheme algorithms that optimise SHM methods can be found, and new pieces of knowledge are produced, since the newly assembled SHM method may be something that has never been applied before. An example of the above is having an algorithm of "supervised classification approaches" class connected to "classification neural networks" to perform the classification. The neural network class is a subclass of "classification models" just like "classification support vector machines". The neural networks then can be replaced by support vector machines and an alternative way of applying the existing "supervised classification approaches" method has been created. This procedure is shown in Figure 6.

Another aspect of the ontology is that it can assist in developing software about the topic it describes. Ontologies are tightly connected to object-oriented programming [24]. Classes exist in both cases and objects of a class inherit properties form their parents. Implementing knowledge from an ontology in combination with an object-oriented scheme is much easier than trying to directly implement all methods as different software pieces. Clustering in classes encourages reusability and makes software more understandable. Different modules created in a project can be transferred and used in other projects, both in the case of software modules and as ontological classes. Connections define clearly the interaction between software objects and help developers understand the way all components should be combined to achieve a specific result. This approach can be considered as another way of increasing productivity within a group, by facilitation of all software implementations needed.

The database aspect of the ontology is also a straightforward one. The "Data" class contains all data coming from both real-life structures and simulation models. Connections between these data and the structures and sensors they come from form a filing system or a database. Users of the ontology can at anytime find data coming from objects of interest, since connections exist between them and are produced by their data. At the same time, data are partitioned into categories according to what they represent; for example displacement, acceleration, frequency response functions, probability density functions of events, etc. Moreover, data are connected to processing methods that potentially transform them and users can have access directly to these methods and process the data in hand. This detailed description of data also assists exchanging data in different projects and disciplines.

Apart from direct consideration of the ontology as a database described in the previous paragraph, it can also be considered a database for instances of algorithms and methods. Every individual in the ontology is an object that interacts with other objects and has attributes. These objects are implemented and stored during a project. Trying to find an object can be easier if it is searched through values of it's characteristics. For example, looking for a finite element model with more than 10000 degrees of freedom can be a search within the class of finite element models for objects with "degrees of freedom" attribute value greater than 10000. Furthermore, the filing algorithms used makes the ontology a knowledge-base, including every possible method that can be used in the scope of SHM. Users can look for them and read about them and their interactions with other instances.

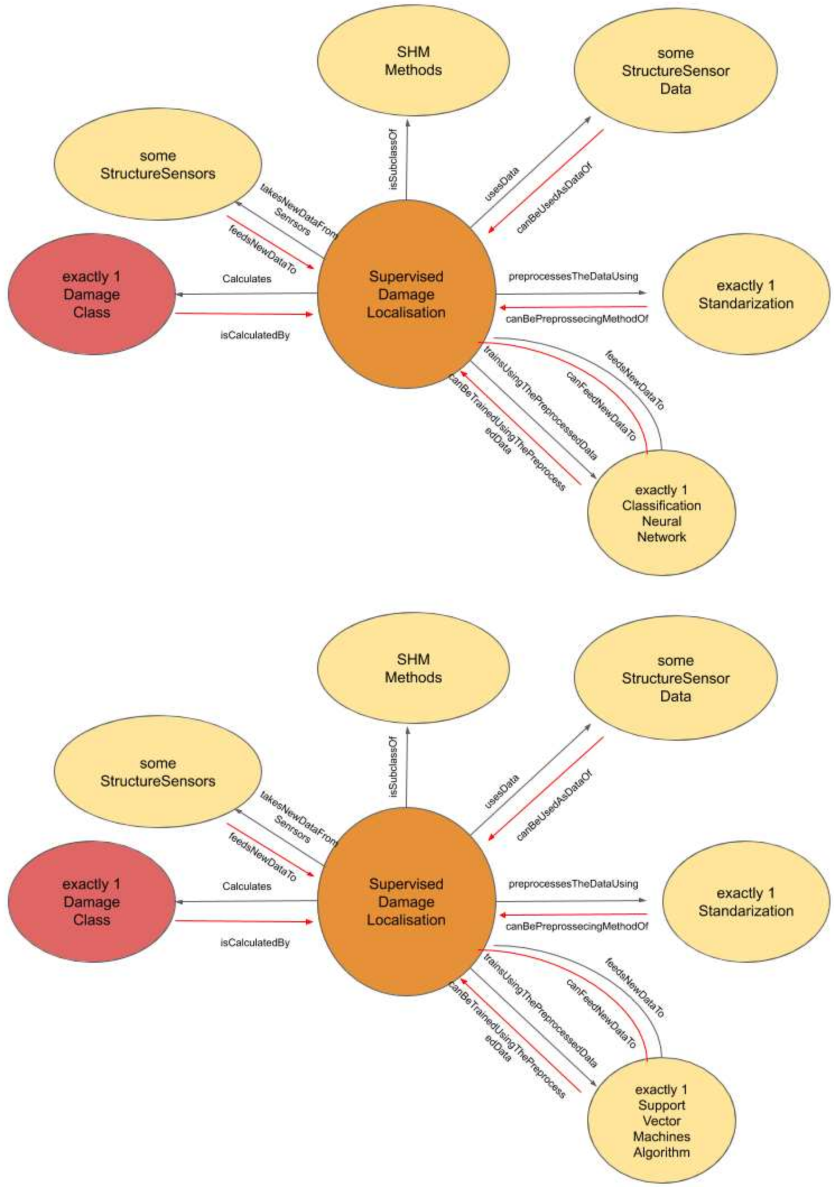

Figure 6: Initial method of supervised damage classification using neural networks (top) and alternative method created by swapping neural networks for support vector machines (bottom).

6 Discussion and Conclusions

The ontology is clearly a way of describing one's knowledge about a subject. It is a diagramatisation of intuition within a discipline, connecting different parts of it and explaining the semantics of connections. The semantics of components and those of their interactions are well defined, trying to connect every piece of knowledge one has about a subject. It is well suited for SHM projects, as methods from different disciplines have to be used and interact. The ontological framework proposed can assist in sharing knowledge within one or many projects, as parts of it are transferable and may be fit for other purposes as well. The ontology is structured in such a way so that it can be expanded by addition of more elements used under the scope of SHM. Moreover, expanding of the ontology, following the proposed scheme of inverse properties, will connect existing elements to new ones updating even current knowledge description.

Different aspects of the ontology were discussed. A quite important and useful one is the database aspect. An ontology potentially is a database and a knowledge-base for projects. Following this approach, databases for scientific projects can be created and data sharing is made easier, since the ontology gives a detailed description of the data, the source of data and also the algorithms that may be used to process them. Searching for data through special characteristics that one needs is facilitated by looking for elements in the ontology with specific attribute values.

As future work, it should be noted that automatic ontology generation would be a really useful tool in the current framework. Generating automatically (or semi-automatically) ontological classes and incorporating them within the ontology will increase its capabilities in boosting productivity of teams. Extending the ontology in such a way will also assist in optimising algorithm performance since ontological components may be connected in more different ways as it extends, and more efficient algorithms can be found for SHM problems (or other disciplines). Finally, knowledge extension would be benefited by automatic generation of ontological components by allowing users to understand new methods and how they are used and combine them with existing ones.

7 Acknowledgement

The authors would like to acknowledge the support of the Engineering and Physical Science Research Council (EPSRC) and the European Commission (EC). This work was started while I. A., D.J.W and K. W. were supported by EPSRC grant EP/K003836/2, and finished under the EC Marie Skłodowska-Curie ETN grant "DyVirt" (764547) which directly supports G.T.

References

[1] C.R. Farrar and K. Worden. *Structural Health Monitoring: a Machine Learning Perspective.* John Wiley & Sons, 2012.

[2] C.R. Farrar and K. Worden. An introduction to structural health monitoring. In *New Trends in Vibration Based Structural Health Monitoring*, pages 1–17. Springer, 2010.

[3] M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11:93–136, 1996.

[4] L.A. Bull, P.A. Gardner, J. Gosliga, T.J. Rogers, M. Haywood-Alexander, N. Dervilis, E.J. Cross, and K. Worden. Towards a population-based structural health monitoring. Part I: Homogeneous populations and forms. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[5] J. Gosliga, P.A. Gardner, L.A. Bull, N. Dervilis, and K. Worden. Towards a population-based structural health monitoring. Part II: Heterogeneous populations and structures as graphs. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[6] J. Gosliga, P.A. Gardner, L.A. Bull, N. Dervilis, and K. Worden. Towards a population-based structural health monitoring. Part III: Graphs, networks and communities. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[7] P.A. Gardner and K. Worden. Towards a population-based structural health monitoring. Part IV: Heterogeneous populations, matching and transfer. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[8] W. Lin, K. Worden, and E.J. Cross. Towards a population-based structural health monitoring. Part VII: EoV fields: environmental mapping. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[9] I. Antoniadou, R.J. Barthorpe, and K. Worden. Towards an ontology for verification and validation in structural dynamics. In *Proceesdings 26th International Conference on Noise and Vibration Engineering (ISMA2014), Leuven, Belgium*, pages 1171–1178, 2014.

[10] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[11] H. Knublauch, D. Oberle, P. Tetlow, and E. Wallace. A semantic web primer for object-oriented software developers. W3C Working Group Note 9th March 2006, W3C, 2006.

[12] M.A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1:4–12, 2015.

[13] R.J. Allemang. The modal assurance criterion (MAC): Twenty years of use and abuse. *Sound and Vibration*, pages 14–21, 2003.

[14] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[15] C.M Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[16] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5^{th} Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[17] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[18] M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[19] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2002.

[20] C.K. Chui. *An Introduction to Wavelets*. Academic Press, 1992.

[21] J.H. Hensman, K. Worden, M. Eaton, R. Pullin, K.M. Holford, and S.L. Evans. Spatial scanning for anomaly detection in acoustic emission testing of an aerospace structure. *Mechanical Systems and Signal Processing*, 25:2462–2474, 2011.

[22] R. Fuentes, E.J. Cross, N. Ray, N. Dervilis, T. Guo, and K. Worden. In-process monitoring of automated carbon fibre tape layup using ultrasonic guided waves. In *Special Topics in Structural Dynamics, Volume 6*, pages 179–188. Springer, 2017.

[23] N. Dervilis, M. Choi, S.G. Taylor, R.J. Barthorpe, G. Park, C.R. Farrar, and K. Worden. On damage diagnosis for a wind turbine blade using pattern recognition. *Journal of Sound and Vibration*, 333:1833–1850, 2014.

[24] E. Kindler and I. Krivy. Object-oriented simulation of systems with sophisticated control. *International Journal of General Systems*, 40:313–343, 2011.