Specifying Configurable Videos with Feature Models

Sebastian Lubos[1], Alexander Felfernig[1] and Viet-Man Le[1]

[1]Graz University of Technology, Graz, Austria

Abstract

Personalization of products is a popular aspect in various application domains, including videos. Enabling users to consume personalized learning videos that include only individually relevant content has the potential to deliver additional benefits in e-learning, for example, by making learning more efficient. In this paper, we present a practical approach to define configurable videos based on feature models, as well as an integrated solution to derive personalized videos by respecting given constraints. Additionally, the possibility to extend the configuration with interactive video elements is described to enable an improved user experience.

Keywords

Feature Models, Configuration, Interactive Video, Personalized Video, Video Summarizing,

1. Introduction

Configuration of software, services, and products fulfilling individual needs has been a popular topic of research in recent years [1, 2]. Feature models [3] have thereby proven to be an excellent approach to solving those challenges in a variety of domains [4], including videos [5]. The possibility to create personalized videos offers huge potential, especially in the domains of knowledge transfer and e-learning where videos are consumed to increase know-how or study new topics.

A challenge in this area is the availability of different learning videos on video platforms, for example, YouTube[1], covering various topics, explained to consumers with individual pre-knowledge. The large variety of options and poor possibilities to determine if videos are relevant before consumption make it complicated for users to find adequate videos [6]. Effective learning videos have the characteristics to include all relevant information for a user to understand and follow the video. At the same time, they reduce learning time by excluding unrelated or already known information [5].

The usage of natural language queries to retrieve video summaries [7], and more recently the integration of chatbots to query and interact with videos[2], have been published as a possibility to support users. While those approaches work well if users are able to specify what they are searching for, a knowledge-based configuration of videos [5] has been presented as a possibility to mitigate this weakness, by giving the user more assistance. User requirements are collected and used in a *Constraint Satisfaction Problem (CSP)* to determine a video fulfilling the user requirements.

Based on the findings in [5], we demonstrate a flexible and reusable approach to define configurable videos, and show an example instantiation that provides a personalized video using the *Choco* solver[3].

Previous work in the synthetic creation of videos has applied video processing techniques to change the visual content to generate artwork variants using variability management techniques [8], and to generate multiple video variants for algorithm test samples [9, 10, 11]. In our approach, we preserve the video content of existing videos and parts of videos while enhancing user experience by transforming it into a well-organized structure. This extends the work of an online video generator taking an initial selection of video clips as seed to create variants of humorous video segments [12], by applying it in the domain of learning videos with additional user requirements and more complex constraints.

The major contributions of the paper are the following. We extend our previous work on the problem definition of configurable videos [5], by demonstrating a practical implementation. A reusable and adaptable approach to defining the structure of configurable videos using feature model technologies is presented, including the integration of a solver to generate personalized videos with respect to specified user requirements. Furthermore, we explain how the solution can be extended to integrate decision points with interactive video elements [13] for an improved individual user experience.

The remainder of this paper is organized as follows. Our approach to specifying a configurable video is explained in Section 2. In Section 3, we present an example configuration with specified user requirements and the resulting video. In Section 4, the reusability and limita-

ConfWS'23: 25th International Workshop on Configuration, Sep 6–7, 2023, Málaga, Spain

slubos@ist.tugraz.at (S. Lubos); alexander.felfernig@ist.tugraz.at (A. Felfernig); vietman.le@ist.tugraz.at (V. Le)
0000-0002-5024-3786 (S. Lubos); 0000-0003-0108-3146 (A. Felfernig); 0000-0001-5778-975X (V. Le)

[1]https://www.youtube.com

[2]e.g., https://www.ortusbuddy.ai/

[3]https://choco-solver.org

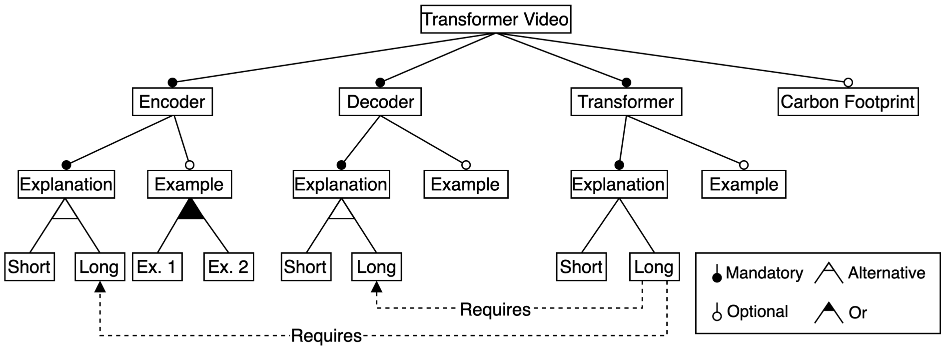

Figure 1: Feature model of an example configurable video about the transformer model.

tion of this approach are discussed. Finally, open research issues are discussed in Section 5 before the paper is concluded with Section 6.

2. Specification of Configurable Videos

Developing feature models is a complex task for persons not experienced with the notation and technologies. In order to ease the problem definition, we enable users to define the structure of configurable videos using the JSON notation provided in [14]. The JSON format is heavily used in different applications, and many software developers have experience with it. For this reason, we expect that it will ease the future implementation of a GUI-based editor for configurable videos, such that they can be configured by everyday users.

Within this paper, we use the configuration of a learning video explaining the *transformer* model in machine learning [15] as a running example. A transformer model is a type of deep learning architecture designed to process sequential data, such as text or speech, by leveraging self-attention mechanisms. It is a popular and powerful model for a variety of natural language processing (NLP) applications, including, machine translation, language understanding, and text generation.

For our example, we use videos from the *Hugging Face* tutorial on NLP[4] hosted on YouTube. Using those tutorial videos, we designed a configurable video with the structure presented as feature model [3] in Figure 1. Leaves of the model refer to single video segments and parent features define the category. Video segments are interpreted as the property values of the configured video. Following the structure, we see that each video includes at least an explanation of the individual components of the transformer model (*encoder* and *decoder*), followed by an explanation of the overall model. Examples of each part can be included optionally, as well as a digression on the carbon footprint of transformer models.

Using the formatting described in [14] the feature model can be described in a JSON format using a straightforward approach. The nested JSON structure enables the user to define the model following a top-down approach, where for each node, an id, the type (mandatory, optional, root), child nodes as well as sibling-relation (alternative, or) can be defined. Furthermore, exclusions and required properties can be specified using their id. Following the algorithms described in [14], the JSON structure can be translated into a valid feature model. In Figure 2, a part of the configuration is shown as an example.

In addition to this generic approach to describing the structure of the configurable video, information on the included video segments is required. More specifically, an *URL* where the video is available is needed. Additionally, the duration of the segments in seconds is required to define constraints for the overall duration of the generated video. This is also defined using a JSON structure, where each key references an id of the model described as JSON. An example is shown in Figure 3.

The definition of the configured video is then used to instantiate a model for the CSP. We used *PyCSP3*[5] for this purpose, which is a *Python* framework, to describe models for *CSPs* in a declarative manner. It includes the possibility to choose between the solver of *ACE (AbsCon Essence)*[6] and *Choco*. The generic code to use provided JSON files for the instantiation of the configurable video, as well as the complete example, are available in our repository[7].

Besides the description of a configurable video, user requirements need to be collected in order to personalize the video. While different possibilities, including the assessment of pre-knowledge, are possible, we restrict

[4]https://huggingface.co/learn/nlp-course

[5]http://pycsp.org
[6]https://github.com/xcsp3team/ace
[7]https://github.com/slubos/specifying-configurable-videos

```
{
   "id":"TransformerVideo",
   "type":"root",
   "parent":"",
   "relation":"",
   "requires":[],
   "excludes":[],
   "children":[
      {
         "id":"Encoder",
         "type":"mandatory",
         "parent":"TransformerVideo",
         "relation":"",
         "requires":[],
         "excludes":[],
         "children":[
            {
               "id":"EncoderExplanation",
               "type":"mandatory",
               "parent":"Encoder",
               "relation":"",
               "requires":[],
               "excludes":[],
               "children":[
                  {
                     "id":"EncoderExplanationShort",
                     "type":"optional",
                     "parent":"EncoderExplanation",
                     "relation":"alternative",
                     "requires":[],
                     "excludes":[],
                     "children":[]
                  },
                  ...
               ]
            },
            ...
         ]
      },
      ...
   ]
}
```

Figure 2: A part of the JSON definition of the configuration aspects for the example video on transformer models.

those to the maximum video duration for our example, assuming that the video is suitable for a beginner level. Especially in preparation for exams, students often follow the utility maximization problem [16], and try to learn as much as possible in a limited amount of time. To capture this requirement, the maximum acceptable video duration of a user is collected and translated to a maximum duration constraint. The duration is determined by summing the duration of each included video segment.

The personalized video is then generated following the *configuration task* described in [5], consisting of a feature

```
{
    "EncoderExplanationShort": {
        "url": "https://youtu.be/H39Z...",
        "duration": 45
    },
    "EncoderExplanationLong": {
        "url": "https://youtu.be/MUqN...",
        "duration": 141
    },
    ...
}
```

Figure 3: A part of the JSON definition of video sources for the example video on transformer models.

model and a defined set of user requirements [17]. The solution of this task is a *configuration*, i.e., an assignment of variables in the CSP, such that the constraints of the model and user requirements are fulfilled [5].

In our presented approach, the variables V state which of the individual video segments are included in the configured video. The respective variable domain is $\{true, false\}$, describing the inclusion or exclusion of a video segment. Nodes of the feature model described in the JSON file are defined as variables. Furthermore, for all video segments, a variable describing the duration is defined within the domain $\{0, videoduration\}$, where *videoduration* is the length of the video in seconds defined in the JSON file. Using a constraint, the value of this variable is restricted to 0 or the defined *videoduration* depending on the inclusion of the segment in the configured video. Further knowledge base constraints are directly derived from the feature model described as JSON file, following the algorithms described in [14].

3. Video Configuration Results

A configured video can be interpreted as a simple playlist, i.e., the included video segments will be played in an ordered fashion. Table 1 shows an example of the minimal and maximal video configuration in terms of video duration of the transformer model example. Depending on the maximum acceptable video duration of the user, different video segments are included or excluded.

Since multiple versions can be the result of a configuration task, the user has the choice to select one of the options. Considering, for example, 250s as the maximum acceptable duration, 15 configurations have been found. To enable the choice, the total duration could be shown, such that the user can select if they want to use most of their available time or not. Alternatively, a further explanation alternative might describe the content in a contrastive way, for example, *video A* contains more detailed explanations, while *video B* has more examples.

	Transformer Video										
	Encoder				Decoder			Transformer			Carbon Foot-print
	Explanation		Example		Explanation		Example	Explanation		Example	
	Short	Long	Ex. 1	Ex. 2	Short	Long		Short	Long		
Min. Config	X				X			X			
Max. Config		X	X	X		X	X		X	X	X

Table 1

The examples demonstrate the minimal and maximal video configurations in terms of video duration for the transformers model example. An **X** indicates that the respective video segment is part of the configuration.

A more advanced solution is based on *interactive videos* which offer an extension to classical videos by offering several interactivity features [13]. This approach is used to enable the user during the video consumption if parts of the video should be included, as long as they still fulfill the duration requirement. In terms of alternative videos, this means a choice is presented which path is followed. After each choice, the selection is included as a constraint, and new paths are configured on the fly. For ”or” video segments, the user can have the option to choose one or both. In the case of optional segments, the user is asked if they want to watch it.

Figure 4 sketches the possible path flow including the decision points for the transformer video example described with the feature model in Figure 1, given the example requirement of 250s as the maximum acceptable duration. Decision points in the workflow diagram are shown with the diamond symbol. For an interactive video, this can be implemented as a question, with choices shown by the outgoing arrows, labeled with their description. The rectangle with rounded corners indicates which video is played. A circle indicates the start, while a double-edged circle represents the end.

Each time a user takes a decision, the value is added as a constraint to the CSP, such that the remaining paths and options are computed dynamically while the user is consuming the video.

4. Discussion

This paper presents a reusable approach for creating configurable videos adaptable to any topic. It requires the availability of manually structured videos by the creator, and the specification of video sources must be updated accordingly (see Figure 3). While YouTube videos were used as an example, any video source could be utilized. The video creator is responsible for adapting the model for the configurable video to represent their desired structure and constraints (see Figure 2). This approach is versatile and can represent any video structure, also including more videos and constraints. We expect the approach to scale well for more complex videos, as the constraints are rather simple and the number of variables is manageable. Yet, we leave this experiment open for future work.

To simplify the process of specifying configurable videos, a GUI-based editor could be used instead of relying on the video creator’s knowledge of JSON files and feature models. The editor would allow the user to add videos by pasting links, then organize them into categories using a drag-and-drop approach. Constraints within categories could be specified using different group types, indicating whether they are alternatives or multiple options to include. Additionally, the video creator could designate videos as mandatory or optional. Cross-tree constraints could be added additionally to specify the requirement of videos from other categories. We expect this user-friendly approach to be easily understandable, eliminating the need for understanding feature models. The translation of the GUI input to JSON is handled by the application.

5. Open Issues for Future Work

One topic for future work is the implementation of the interactive video approach described in this paper. Frameworks for this purpose, e.g., FrameTrail[8] or H5P[9], offer the possibility to define the interactive elements and use them for playout. As we expect that this kind of video consumption improves learning effectiveness, conducting a user study to examine this assumption is planned. A between-subject study could be conducted, where one group uses interactive videos, while the other views the complete video without interaction. Questionnaires about the video topic immediately after the video and after a few weeks could be used to analyze the short- and long-term learning effectiveness of the interactive approach.

Further topics include the support of users in the definition of configurable videos. While our approach offers the possibility to describe the structure of those, it is still a lot of manual annotation work to describe the video,

[8] https://frametrail.org
[9] https://h5p.org/

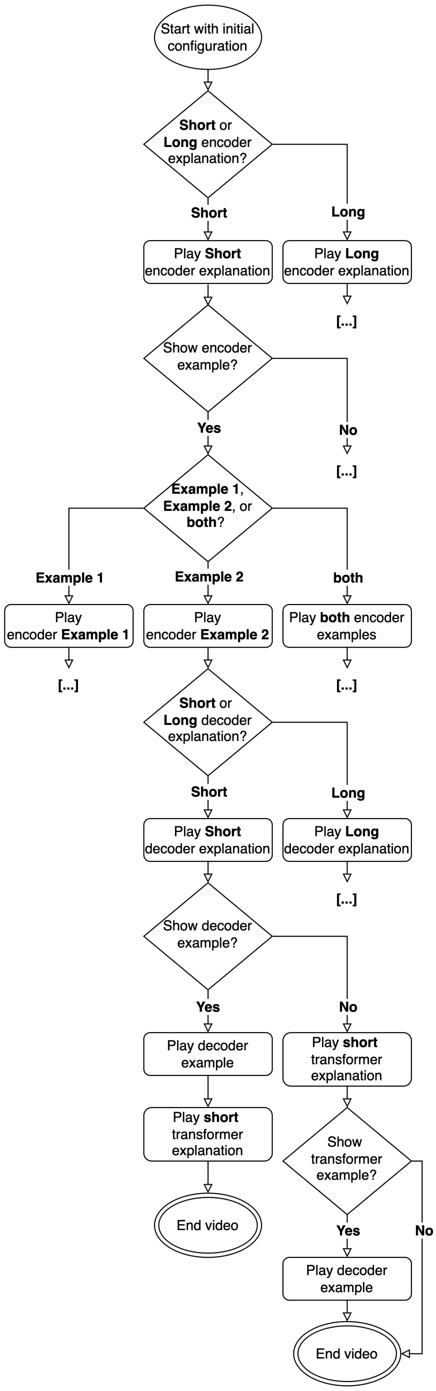

Figure 4: Possible paths of the configured video example with interactive elements. Diamonds indicate decisions users take during video consumption, while rectangles resemble played video segments. Not all paths are shown for simplicity.

for this purpose, options to automatize parts of this work should be considered. This includes the automated indexing of video content to ease recognition of what is included [18, 19], as well as their semantic segmentation defining the individual video segments that can be included ([20, 21]). Also, the possible inclusion of recommendation technologies [22, 23] to support the definition of those videos, e.g., by recommending which options could further be included.

Finally, the inclusion of diagnoses [24, 25] should be considered to relax situations where no solution can be found for given user requirements. Those can help to find a configuration that takes into account as much as possible of the original user requirements.

6. Conclusions

With this paper, we present an initial implementation to define and generate personalized videos using feature models. Using an easy-to-use JSON notation, a solution was presented that is able to add additional benefit to knowledge transfer with videos by reusing already existing material. Following a practical example learning video, we showed how the approach can be used, and further extended to enable its usage with interactive videos, which is part of our future work.

Acknowledgments

The presented work has been developed within the research project Streamdıver which is funded by the Austrian Research Promotion Agency (FFG) under the project number 886205.

References

[1] A. Felfernig, L. Hotz, C. Bagley, J. Tiihonen, Knowledge-based Configuration - From Research to Business Cases, Elsevier, 2014.

[2] D. Sabin, R. Weigel, Product configuration frameworks - a survey, IEEE Intelligent Systems 13 (1998) 42–49.

[3] K. Kang, S. Cohen, J. Hess, W. Novak, S. Peterson, Feature-oriented Domain Analysis (FODA) – Feasibility Study, TechnicalReport CMU – SEI-90-TR-21 (1990).

[4] J. Martinez, W. K. G. Assunção, T. Ziadi, Espla: A catalog of extractive spl adoption case studies, in: Proceedings of the 21st International Systems and Software Product Line Conference - Volume B, SPLC '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 38–41. URL: https://doi.

org/10.1145/3109729.3109748. doi:10.1145/3109729.3109748.

[5] S. Lubos, M. Tautschnig, A. Felfernig, V.-M. Le, Knowledge-based configuration of videos using feature models, in: Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume B, SPLC '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 188–192. URL: https://doi.org/10.1145/3503229.3547052. doi:10.1145/3503229.3547052.

[6] A. Imran, F. Alaya Cheikh, S. Kowalski, Automatic annotation of lecture videos for multimedia driven pedagogical platforms, Knowledge Management and E-Learning 7 (2015).

[7] M. Vahedi, M. M. Rahman, F. Khomh, G. Uddin, G. Antoniol, Summarizing relevant parts from technical videos, in: 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 434–445. doi:10.1109/SANER50967.2021.00047.

[8] J. Martinez, G. Rossi, T. Ziadi, T. F. D. A. Bissyandé, J. Klein, Y. Le Traon, Estimating and predicting average likability on computer-generated artwork variants, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO Companion '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 1431–1432. URL: https://doi.org/10.1145/2739482.2764681. doi:10.1145/2739482.2764681.

[9] M. Acher, M. Alférez, J. A. Galindo, P. Romenteau, B. Baudry, Vivid: A variability-based tool for synthesizing video sequences, in: Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools - Volume 2, SPLC '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 143–147. URL: https://doi.org/10.1145/2647908.2655981. doi:10.1145/2647908.2655981.

[10] J. A. Galindo, M. Alférez, M. Acher, B. Baudry, D. Benavides, A variability-based testing approach for synthesizing video sequences, in: Proceedings of the 2014 International Symposium on Software Testing and Analysis, ISSTA 2014, Association for Computing Machinery, New York, NY, USA, 2014, p. 293–303. URL: https://doi.org/10.1145/2610384.2610411. doi:10.1145/2610384.2610411.

[11] M. Alférez, M. Acher, J. A. Galindo, B. Baudry, D. Benavides, Modeling variability in the video domain: Language and experience report, Software Quality Journal 27 (2019) 307–347. URL: https://doi.org/10.1007/s11219-017-9400-8. doi:10.1007/s11219-017-9400-8.

[12] G. Bécan, M. Acher, J.-M. Jézéquel, T. Menguy, On the variability secrets of an online video generator, in: Proceedings of the Ninth International Workshop on Variability Modelling of Software-Intensive Systems, VaMoS '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 96–102. URL: https://doi.org/10.1145/2701319.2701328. doi:10.1145/2701319.2701328.

[13] A. Palaigeorgiou, George and Papadopoulou, I. Kazanidis, Interactive video for learning: A review of interaction types, commercial platforms, and design guidelines, in: M. Tsitouridou, J. A. Diniz, T. A. Mikropoulos (Eds.), Technology and Innovation in Learning, Teaching and Education, Springer International Publishing, Cham, 2019, pp. 503–518.

[14] H. Shatnawi, H. C. Cunningham, Encoding feature models using mainstream json technologies, in: Proceedings of the 2021 ACM Southeast Conference, ACM SE '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 146–153. URL: https://doi.org/10.1145/3409334.3452048. doi:10.1145/3409334.3452048.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[16] A. Mas-Colell, M. D. Whinston, J. R. Green, et al., The Utility Maximization Problem, volume 1, Oxford university press New York, 1995.

[17] L. Hotz, A. Felfernig, M. Stumptner, A. Ryabokon, C. Bagley, K. Wolter, Configuration Knowledge Representation and Reasoning, 1 ed., Elsevier B.V., Netherlands, 2014, pp. 41–72.

[18] Y. Deldjoo, Enhancing Video Recommendation Using Multimedia Content, Springer International Publishing, Cham, 2020, pp. 77–89. URL: https://doi.org/10.1007/978-3-030-32094-2_6. doi:10.1007/978-3-030-32094-2_6.

[19] M. Elahi, F. Bakhshandegan Moghaddam, R. Hosseini, M. H. Rimaz, N. El Ioini, M. Tkalcic, C. Trattner, T. Tillo, Recommending Videos in Cold Start With Automatic Visual Tags, Association for Computing Machinery, New York, NY, USA, 2021, p. 54–60. URL: https://doi.org/10.1145/3450614.3461687.

[20] T. Tuna, M. Joshi, V. Varghese, R. Deshpande, J. Subhlok, R. Verma, Topic based segmentation of classroom videos, in: 2015 IEEE Frontiers in Education Conference (FIE), 2015, pp. 1–9. doi:10.1109/FIE.2015.7344336.

[21] P. A. Co, W. R. Dacuyan, J. G. Kandt, S.-C. Cheng, C. L. Sta. Romana, Automatic topic-based lecture video segmentation, in: Innovative Technologies and Learning: 5th International Conference, ICITL 2022, Virtual Event, August 29–31, 2022, Proceedings, Springer-Verlag,

Berlin, Heidelberg, 2022, p. 33–42. URL: https://doi.org/10.1007/978-3-031-15273-3_4. doi:10.1007/978-3-031-15273-3_4.

[22] A. Falkner, A. Felfernig, A. Haag, Recommendation Technologies for Configurable Products, AI Magazine 32 (2011) 99–108.

[23] A. Felfernig, V.-M. Le, A. Popescu, M. Uta, T. N. T. Tran, M. Atas, An overview of recommender systems and machine learning in feature modeling and configuration, in: 15th International Working Conference on Variability Modelling of Software-Intensive Systems, VaMoS'21, Association for Computing Machinery, New York, NY, USA, 2021. URL: https://doi.org/10.1145/3442391.3442408. doi:10.1145/3442391.3442408.

[24] A. Felfernig, M. Schubert, C. Zehentner, An efficient diagnosis algorithm for inconsistent constraint sets, AI for Engineering Design, Analysis, and Manufacturing (AIEDAM) 26 (2012) 53–62.

[25] A. Felfernig, R. Walter, J. Galindo, D. Benavides, M. Atas, S. Polat-Erdeniz, S. Reiterer, Anytime Diagnosis for Reconfiguration, Journal of Intelligent Information Systems 51 (2018) 161–182.