Knowledge-Based UML Activity Model Transformation Algorithm

Ilona Veitaite[a], Audrius Lopata[b]

[a] Institute of Social Sciences and Applied Informatics Vilnius University, Kaunas Facult Kaunas, Lithuania
[b] Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania

Abstract

The main scope is to introduce the transformation algorithm of UML Activity model generation from Enterprise model (EM). The transformation algorithm is described in details by showing full process through steps. Whole generation process steps illustrated by particular example of Paper submission for the publishing following the transformation algorithm step by step.

Keywords

Enterprise Model, IS Engineering, Knowledge-based, UML Activity Model

1. Introduction

Nowadays to insure business and IT alignment it is important to create communication between these two parts. It is often occasion that understanding how to adopt new technologies to influence business is understandable just for one side and talking about this kind of IT improvement does not always perform well.

Information systems (IS) become more complex and modelling methods and techniques are not sufficient to characterize all business and IT processes [1, 2, 3, 4, 5, 6, 7, 8]. Enterprise modelling has become an irreplaceable part of IS development process. Traditionally IS engineering stages from modelling to code generation are implemented empirically.

Moreover, nowadays, computer-based IS engineering, to avoid the empirical influence, is developing based on new knowledge-based methods [9, 10, 11]. Computer-based IS in knowledge-based IS engineering, is developed using stored enterprise knowledge base of the particular business domain, i.e., enterprise model, the composition which is defined by formal criteria [1, 2, 12, 13, 4, 14, 8, 15]. UML is one of the most common software specifications. It is a universal IS modelling language which is applied to a multitude of methodologists and used in the most popular modelling tools. The importance of UML in software development has become more significant since the appearance of model-driven architecture [16, 17, 18, 19].

The method of UML models generation from EM implements a knowledge-based design stage in the IS development cycle. UML dynamic models can be generated through transformation algorithms, when the proper knowledge is collected into knowledge repository, where it is already verified to insure automatically generated design models quality [18, 19].

2. Enterprise Model As Basis In Knowledge-based Is Engineering Process

EMM is formally defined EM structure, which consists of a formalized EM in line with the general principles of control theory. EM is the main source of the necessary knowledge of the particular business domain for IS engineering and IS re-engineering processes (Fig. 1) [20, 21, 22].

EM class model has twenty-three classes. Essential classes are Process, Function and Actor. Class Process, Function, Actor and Objective can have an internal hierarchical structure. These relationships is presented as aggregation relationship. Class Process is linked with the class MaterialFlow as aggregation relationship. Class MaterialFlow is linked with the classes MaterialInputFlow and MaterialOutputFlow as generalization relationship.

Class Process is linked with Classes Function, Actor and Event as association relationship. Class Function is linked with classes InformationFlow, InformationActivity, Interpretation, InformationProcessing and Realization as aggregation relationship. These relationships define the internal composition of the Class Function. Class InformationFlow is linked with ProcessOutputAtributes, ProcessInputAtributes, IPInputAttributes and IPOutputAttributs as generalization relationship.

IVUS 2020: Information Society and University Studies, 23 April 2020, KTU Santaka Valley, Kaunas, Lithuania
Ilona.Veitaite@knf.vu.lt (I. Veitaite); Audrius.Lopata@ktu.lt (A. Lopata)

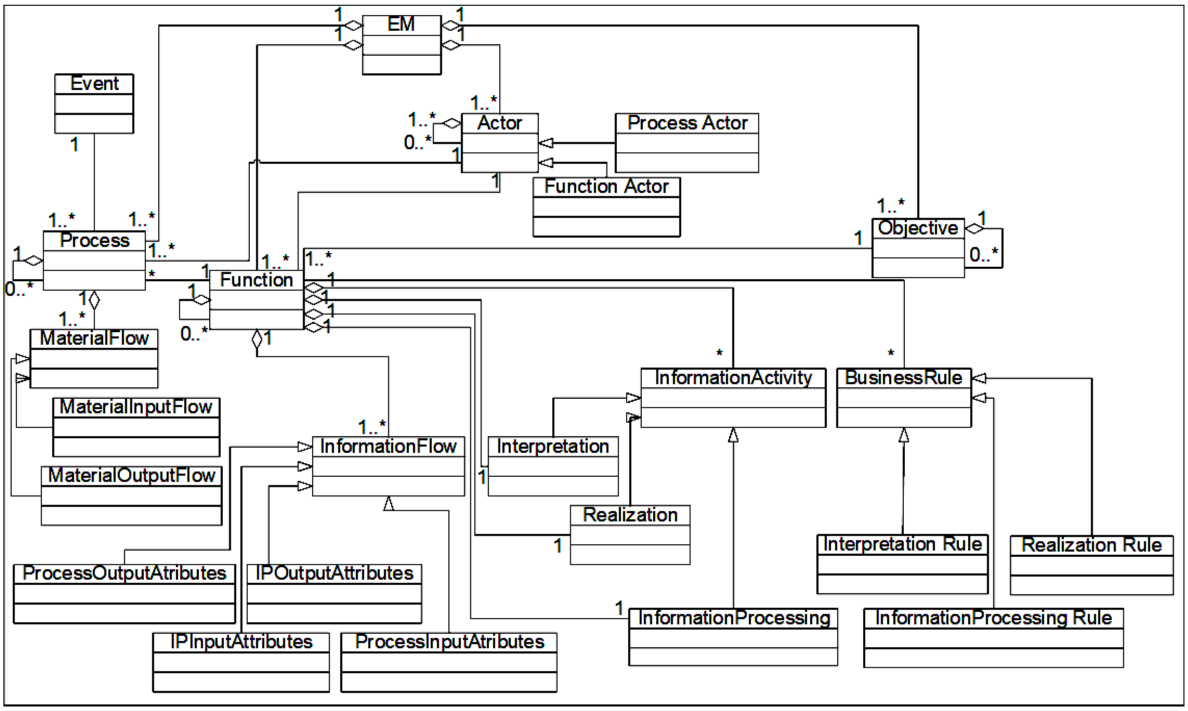


Figure 1: EMM class diagram [20],[21],[22]

Class InformationActivity is linked with Interpretation, InformationProcessing and Realization as generalization relationship. Class Function linked with classes Actor, Objective and BusinessRule as association relationship. Class BusinessRule is linked with Interpretation Rule, Realization Rule, InformationProcessing Rule as generalization relationship. Class Actor is linked with Function Actor and Process Actor as generalization relationship [18, 19, 23].

3. Variations Of Enterprise Model Elements Role

Information systems design methods indicates the arrangement of systems engineering actions, i.e. how, in what order and what UML model to use in the IS development process and how to implement the process. Majority of them are based on different types of models describing varying aspects of the system qualities. Meaning of each model can be defined separately, but more important is the fact that each model is the projection of the system. An unexperienced specialist can use UML models inappropriately and the description of the system will supposedly be insufficient [23, 24, 25, 26]. Identifying specific UML model and selecting the initial model element is reasonably meaningful, because further generating process relies on it. Many UML model elements iterates in different UML model, but these elements describe different aspects of the system. Table 1 presents how Enterprise model element Business rule Actor, Process and Function can have different significance in different UML models [23, 24, 25, 26].

4. UML Models Transformation Algorithms

All UML models: static and dynamic can be generated from Enterprise model using transformation algorithms [24, 25]. Figure 2 presents transformation algorithm of UML model generation from EM process and is described by following steps [24].

- Step 1: Particular UML model for generation from EM process is identified and selected.
- Step 2: If the particular UML model for generation from EM process is selected then algorithm process is continued, else the particular UML model for generation from EM process must be selected.

Table 1
Enterprise Model Business Rules Elements Role Variations In Part Of Uml Dynamic Models [24], [25]

EM	UML Model element	UML Dynamic Model
Business Rule	Extend	Use Case Model
	Include	Use Case Model
	Association	Use Case Model
	Control Nodes	Activity Model
	Time Constraint	Timing Model
	Destruction Occurrence	Timing Model
	…	…
	Pseudostate	State Machine Model
	…	…
Actor	Actor	Use Case Model
	Subject	Use Case Model
	Partition	Activity Model
	Lifeline	Sequence Model
	Lifeline	Communication Model
	…	…
	Lifeline	Timing Model
	…	…
Process/Function	Use Case	Use Case Model
	Activity	Activity Model
	Frame	Communication Model
	…	…
	Message	Sequence Model
	…	…

- Step 3: First element from EM is selected for UML model, identified previously, generation process.
- Step 4: If the selected EM element is initial UML model element, then initial element is generated, else the other EM element must be selected (the selected element must be initial element).
- Step 5: The element related to the initial element is selected from Enterprise model.
- Step 6: The element related to the initial element is generated as UML model element.
- Step 7: The element related to the previous element is selected from Enterprise model.
- Step 8: The element related to the previous element is generated as UML model element.

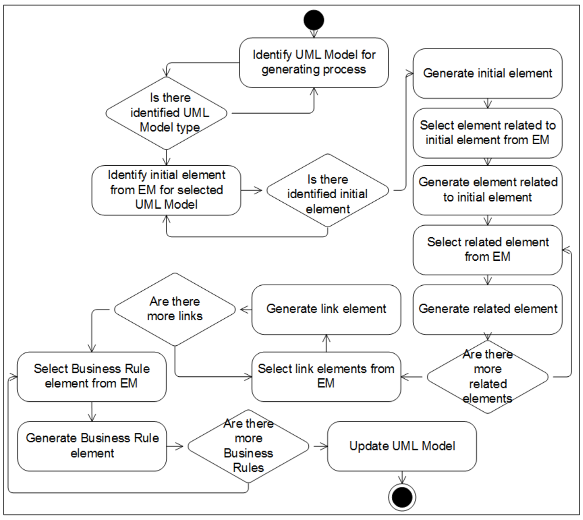

Identify UML Model for generating process
Is there identified UML Model type
Identify initial element from EM for selected UML Model
Is there identified initial element
Generate initial element
Select element related to initial element from EM
Generate element related to initial element
Select related element from EM
Generate related element
Are there more related elements
Select link elements from EM
Generate link element
Are there more links
Select Business Rule element from EM
Generate Business Rule element
Are there more Business Rules
Update UML Model

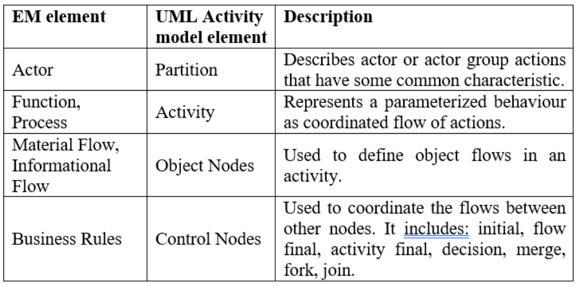
Figure 2: The top level transformation algorithm of UML models generation from EM process [24], [25]

- Step 9: If there are more related elements, then they are selected from EM and generated as UML model elements one by one, else the link element is selected from Enterprise model.
- Step 10: The link element is generated as UML model element.
- 11: If there are more links, then they are selected from EM and generated as UML model elements one by one, else the Business Rule element is selected from Enterprise model.
- Step 12: The Business Rule element is generated as UML model element.
- Step 13: If there are more Business Rules, then they are selected from EM and generated as UML model elements one by one, else the generated UML model is updated with all elements, links and constraints.
- Step 14: Generation process is finished.

4.1. UML Activity Model Description

Activity model is one of the UML dynamic models, which shows flow of control or object flow with underlining the sequence and conditions of the particular flow. The actions which are coordinated by activity models can be initiated because other actions finish executing because objects and data become available, or because some events external to the flow occur [16, 17, 19].

Table 2
presents elements from Enterprise model – input elements and elements generated to UML Activity model – output elements.[24, 25].



EM element	UML Activity model element	Description
Actor	Partition	Describes actor or actor group actions that have some common characteristic.
Function, Process	Activity	Represents a parameterized behaviour as coordinated flow of actions.
Material Flow, Informational Flow	Object Nodes	Used to define object flows in an activity.
Business Rules	Control Nodes	Used to coordinate the flows between other nodes. It includes: initial, flow final, activity final, decision, merge, fork, join.

Table 2 presents elements from Enterprise model – input elements and elements generated to UML Activity model – output elements [23, 24, 25]

Figure 3 presents UML Activity model generation from Enterprise model transformation algorithm. Transformation algorithm is described by following steps:

- Step 1: The initial element Actor from Enterprise Model for UML Activity model generation is selected.
- Step 2: Partition element is generated.
- Step 3: Process element from Enterprise model, which is related with the initial Partition element is selected.
- Step 4: If Process element is Activity element related to Partition, then Activity element is generated, else Function element is selected.
- Step 5: Function element is generated as Activity element.
- Step 6: Partition element is linked with Activity element.
- Step 7: There is checking if there are more Processes in Enterprise Model related to UML Activity model. In case, there are, algorithm goes back to step 3.
- Step 8: Material Flow element from Enterprise model, which is related with the link to the Partition element is selected.
- Step 9: If Material Flow element is an Object Node element related to Partition and Activity elements, then Object Node element is generated.

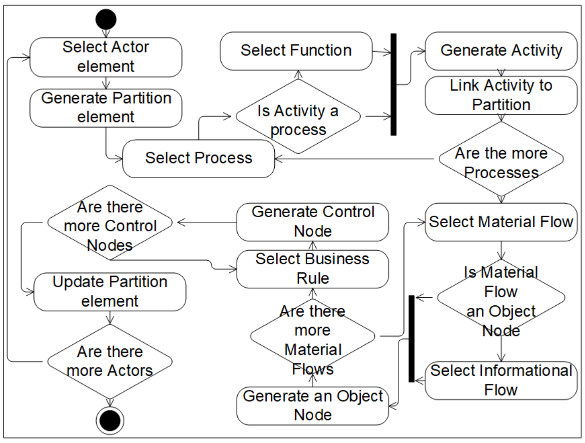


Figure 3: UML Activity Model transformation algorithm [24], [25]

- Step 10: Else Informational Flow element is selected.
- Step 11: Object Node element is generated.
- Step 12: There is checking if there are more Material Flow elements in Enterprise Model related to UML Activity model. In case, there are, algorithm goes back to step 8.
- Step 13: Business element is selected.
- Step 14: Control Node element is generated.
- Step 15: There is checking if there are more Business Rules in Enterprise Model related to UML Activity model. In case, there are, algorithm goes back to step 13.
- Step 16: Partition element is updated.
- Step 17: There is checking if there are more Actor elements in Enterprise Model related to UML Activity model. In case, there are, algorithm goes back to step 1.

4.2. An Example of Paper Submission for Publishing Process

In the Enterprise model there is stored data about submitting the paper for the publishing. This information consists of actors, processes, functions, material and informational flows and business rules. Regarding stored date it is possible to claim, that this data is enough for UML Activity model generation process. The example presents Paper submission for the publishing process, where are three participants: author,

Table 3

Step 1 And Step 2 In UML Activity Model Generation Process

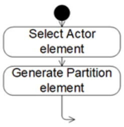
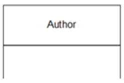
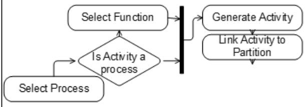
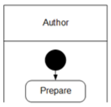
Transformation algorithm part	Enterprise model element	Generated UML Activity model element
	Actor	

Table 4

Steps 3, 4, 5, 6 in UML activity model generation process

Transformation algorithm part	Enterprise model element	Generated UML Activity model element
 Select Function Generate Activity Link Activity to Partition Is Activity a process Select Process	Process Function	

who prepares and submits the paper, also updates the paper after the review; reviewer, who reviews the paper, makes the decision regarding rejection, prepares review report and recommendations for the update; Editor, who receives review report, makes the decision regarding acceptance and publishes the final version of the paper. Table 3 presents first two steps of UML Activity model transformation algorithm, where Actor element is selected from Enterprise model and generated as UML Activity model's Partition element. Author – first participant of the Paper submission for publishing example is generated.

Table 4 presents next four steps of UML Activity model transformation algorithm, where Process/function element is selected from Enterprise model and generated as UML Activity model's Activity element. Paper preparation – first activity of the author of the Paper submission for publishing example is generated.

Table 5 presents seventh step of UML Activity model transformation algorithm, where next where Process-/function element is selected from Enterprise model and generated as UML Activity model's Activity element. Paper submission – second activity of the author of the Paper submission for publishing example is generated.

Table 5

Step 7 In UML Activity Model Generation Process

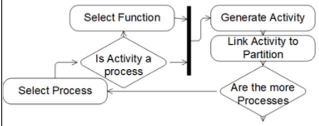
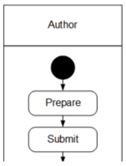
Transformation algorithm part	Enterprise model element	Generated UML Activity model element
 Select Function Generate Activity Link Activity to Partition Is Activity a process Select Process Are the more Processes	Process Function	

Table 6

Step 8. 9, 10 And 11 In UML Activity Model Generation Process

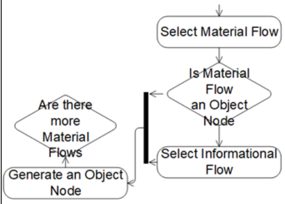
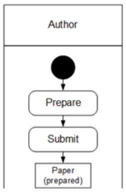
Transformation algorithm part	Enterprise model element	Generated UML Activity model element
 Select Material Flow Is Material Flow an Object Node Are there more Material Flows Select Informational Flow Generate an Object Node	MaterialInputFlow InformationFlow	

Table 6 presents next four steps of UML Activity model transformation algorithm, where Informational Flow element is selected from Enterprise model and generated as UML Activity model's Object Node element. Prepared paper – first object node of the author of the Paper submission for publishing example is generated.

Figure 4 presents full UML Activity model generated from Enterprise model of an example of Paper submission for publishing process.

Transformation algorithm steps implementation confirms, that data of Paper submission for publishing process stored in Enterprise model can be generated to UML Activity model and as this data is already verified and validated its accuracy and fullness is indeed enough for the UML models generation process.

5. Conclusions

The first part of the paper deals with the presentation of the Enterprise model, EM elements role variations possibilities in UML dynamic models generating process and top level of transformation algorithm. In the next part the explanation of UML Activity model transformation algorithm, which is described by steps,

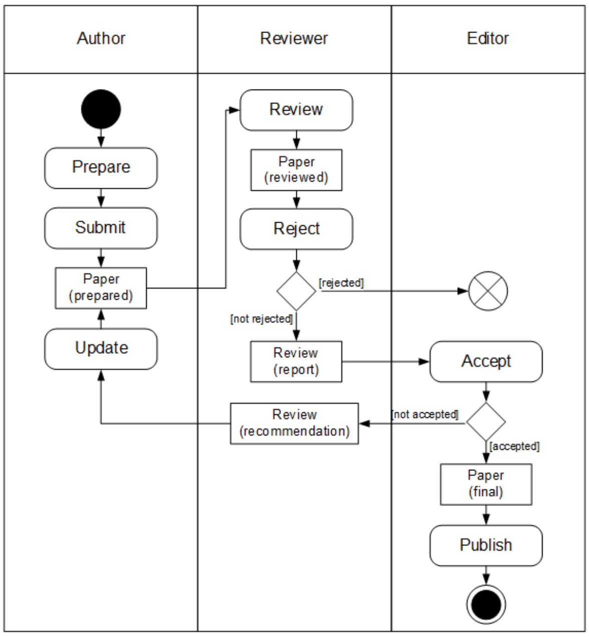


Figure 4: Full UML Activity model generated from Enterprise model of an example of Paper submission for publishing process

is presented. The next part presents particular example, which data is stored in knowledge-based Enterprise model and is used in generation process. There also all the stages of the example are described.

Final part describes transformation algorithm steps for the UML Activity model generation from Enterprise model and this process is presented with graphical schemes. The illustrated example shows that data stored in Enterprise model is enough for generation process and it is possible to confirm, that each element of UML dynamic models can be generated from the Enterprise model. Transformation algorithms usage accomplishes knowledge-based IS development cycle design phase.

References

[1] C.-K. Chen, Construct model of the knowledge-based economy indicators., Transformations in Business & Economics 7 (2008).

[2] R.-S. Chen, C.-M. Sun, M. M. Helms, W.-J. K. Jih, Aligning information technology and business strategy with a dynamic capabilities perspective: A longitudinal study of a taiwanese semiconductor company, International Journal of Information Management 28 (2008) 366–378.

[3] J. Dunkel, R. Bruns, Model-driven architecture for mobile applications, in: International Conference on Business Information Systems, Springer, 2007, pp. 464–477.

[4] J. C. Henderson, H. Venkatraman, Strategic alignment: Leveraging information technology for transforming organizations, IBM systems journal 38 (1999) 472–484.

[5] D. A. Peak, C. S. Guynes, V. R. Prybutok, C. Xu, Aligning information technology with business strategy: An action research approach, Journal of Information Technology Case and Application Research 13 (2011) 16–42.

[6] F. Silvestri, S. Acciarito, G. Cardarilli, G. Khanal, L. Di Nunzio, R. Fazzolari, M. Re, Fpga implementation of a low-power qrs extractor, Lecture Notes in Electrical Engineering 512 (2019) 9–15.

[7] E. Perjons, Model-Driven Process Design: Aligning Value Networks, Enterprise Goals, Services and IT Systems, Ph.D. thesis, Department of Computer and Systems Sciences, Stockholm University, 2011.

[8] L. Plazaola, J. Flores, N. Vargas, M. Ekstedt, Strategic business and it alignment assessment: A case study applying an enterprise architecture-based metamodel, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), IEEE, 2008, pp. 398–398.

[9] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, M. Woźniak, A novel training method to preserve generalization of rbpnn classifiers applied to ecg signals diagnosis, Neural Networks 108 (2018) 331–338.

[10] C. Napoli, F. Bonanno, G. Capizzi, An hybrid neuro-wavelet approach for long-term prediction of solar wind, Proceedings of the International Astronomical Union 6 (2010) 153–155.

[11] G. Capizzi, F. Bonanno, C. Napoli, Hybrid neural networks architectures for soc and voltage prediction of new generation batteries storage, 2011, pp. 341–344.

[12] H. Eichelberger, Y. Eldogan, K. Schmid, A comprehensive analysis of UML tools, their capabilities and their compliance, Inst. für Informatik, 2009.

[13] F. Gailly, S. Casteleyn, N. Alkhaldi, On the symbiosis between enterprise modelling and ontology engineering, in: International Conference on Conceptual Modeling, Springer, 2013, pp. 487–494.

[14] J. Jenney, M. Gangl, R. Kwolek, D. L. Melton, N. Ridenour, M. Coe, Modern methods of systems engineering: with an introduction to pattern and model based methods, Joe Jenney, 2010.

[15] S. Russo, C. Napoli, A comprehensive solution for psychological treatment and therapeutic path planning based on knowledge base and expertise sharing, in: International Conference for Young Researchers in Informatics, Mathematics, and Engineering (ICYRIME), volume 2472 of *CEUR Workshop Proceedings*, 2019, pp. 41–47.

[16] G. Booch, J. Rumbaugh, I. Jacobson, Unified modeling language user guide,(the 2nd edition), 2005.

[17] O. UML, Unified modeling language version 2.5. 1. unified modelling, 2019.

[18] P. S. Sajja, R. Akerkar, Knowledge-based systems for development, Advanced Knowledge Based Systems: Model, Applications & Research 1 (2010) 1–11.

[19] C. Napoli, E. Tramontana, An object-oriented neural network toolbox based on design patterns, in: International Conference on Information and Software Technologies, Springer, 2015, pp. 388–399.

[20] S. Gudas, et al., Enterprise knowledge modelling: domains and aspects, Technological and economic development of economy (2009) 281–293.

[21] S. Gudas, Architecture of knowledge-based enterprise management systems: a control view, in: Proceedings of the 13th world multiconference on systemics, cybernetics and informatics (WMSCI2009),) July, 2009, pp. 10–13.

[22] S. Gudas, Informacijos sistemų inžinerijos teorijos pagrindai, Vilnius University (2012) 384.

[23] I. Veitaite, A. Lopata, Additional knowledge based mof architecture layer for uml models generation process, in: International Conference on Business Information Systems, Springer, 2015, pp. 56–63.

[24] I. Veitaite, A. Lopata, Transformation algorithms of knowledge based uml dynamic models generation, in: International Conference on Business Information Systems, Springer, 2017, pp. 59–68.

[25] I. Veitaite, A. Lopata, Problem domain knowledge driven generation of uml models, in: International Conference on Information and Software Technologies, Springer, 2018, pp. 178–186.

[26] I. Veitaite, A. Lopata, Knowledge based uml information flow model transformation algorithm, in: CEUR Workshop Proceedings, volume 2152, 2018, pp. 30–36.