Domain-specific language to design educational programs with the use of X-matrix approach

Artem Kuzmin
Faculty of Digital Transformation
ITMO University
Saint Petersburg, Russia
337762@niuitmo.ru

Alexey Dukhanov
Faculty of Digital Transformations
ITMO University
Saint Petersburg, Russia
dukhanov@itmo.ru

Semen Kraev
Faculty of Digital Transformations
ITMO University
Saint Petersburg, Russia
simon.kraev@itmo.ru

***Abstract*—This paper introduces a prototype of a domain-specific language for quick and efficient preparation of educational programs. Rapid changes in technology lead to constant modification process of the current program especially in scientific-intensive areas. As a result, teachers may need a tool representing a convenient, structural and formalized approach for learning program construction. The domain-specific language is devoted to serve as such an instrument and is based on concepts of X-matrix and problem areas map. In this paper functions of the language prototype and its syntax are represented. Key entities which comprise X-matrix and problem area matrix are explained. The implementation of operations on the entities in the markup language are described.**

***Keywords*—problem areas map, X-matrix, domain-specific languages**

I. Introduction

This work-in-progress is devoted to the development of domain-specific language prototype for building educational programs. Informational technologies more often lead to digital transformation in various areas including education. One of the tasks of informational technologies in education areas is construction of learning programs where technical issues could significantly influence on the end product. The problem is that team of course designers is faced short deadlines for creation and modification of educational programs. As a result, a little time is left for creativity and making qualitative content in terms of its perception. Strict deadlines is a feature of science-intensive fields where novel technologies and approaches are discovered monthly or annually. Moreover, new technologies lead to the changes in a list of required outcomes and employer's needs therefore it is vital to frequently update educational programs in these areas. Creating new or modifying old educational programs traditionally takes a lot of time, so a tool is needed to reduce the time spent on updating the structural elements and content of educational programs. One of the concepts that addressed to this problem is called the X-matrix, developed in [1]. It implies formation of a visual matrix representation of the program, taking into account the expectations of employers and the latest achievements in scientific and technical fields. This approach is designed to significantly accelerate the development of programs in collaboration with both employers' representatives and scientific experts. The matrix indicates the links between the modules of the educational program, scientific and technical tools, the learning outcomes of the graduate and the expectations of employers. But at the same time, the implementation of the concept is not demonstrated in the article, as well as the entities that the concept operates on were not formalized. Thus, the research question of this work is: How to automate the processes of educational programs creation based on the X-matrix model? We believe that in order to effectively use the X-matrix model, it is better to use domain-specific languages. That is because today algorithmic coding technology is available to a large number of specialists, teachers, including those in the humanities, due to high-level programming languages. Thus, the developed domain-specific language becomes available to an increasing number of educational program designers. In addition, a language with standard terminology and structure will facilitate communication between people in the learning program design team. A set of approaches dedicated to the creation of courses and programs have already used domain-specific languages. These languages are designed to simplify the work of the course designer and allow educators without much programming experience to create full-fledged curricula and courses. So, in [2], 4 subject-oriented languages were designed to solve the problem of effective curriculum development: to create units of educational content, to create evaluation modules, to set the sequence of created educational units and to create gamified units. It was stated that more than 95% of the program developers were satisfied with the ease of use of the language for creating the program, and that the development of e-learning courses using a domain-specific language is much faster than using conventional web applications. The work [3] presents a language for personalizing learning content in online courses that allows a teacher without experience in software development and data science to customize the collected data and how it is processed. Thus, it can be noted that the adaptability of learning can be customized through easy-to-learn tools and can be used to create adaptive curricula. The TESMA language presented in [4] demonstrated that the concepts, models and relationships used in the language are able to describe real curricula and cover many user cases during their development. LiaScript, explained in [5], is aimed to simplify integration with web technologies, reusability of course resources, which plays in favor of considering domain-specific languages when creating curricula.

This work proposes a prototype of a high-level domain-specific programming language, which is a tool for formalized description of educational programs from advanced training courses to bachelors and masters' programs. The language is based on directories, dictionaries and lists reflecting scientific and technical fields, banks of educational and research objects, lists of employers' expectations, competencies, and events. Relationships between horizontal and vertical elements of the matrix are formalized based on translation algorithms for the corresponding resources. In particular, the relationship between scientific and technological tools and learning modules is formalized in the form of algorithms for using research objects together with learning objects in various learning activities. Here the basis is the programming language proposed in [6]. And the relationship between the tools and the expectations of employers is formalized in the form of a high-level pseudocode that represents an algorithm for applying appropriate approaches to solving production problems. In order to create a prototype of a domain-specific language, it is necessary to decompose the X-matrix models and identify the main entities, how they can be embodied, determine the operations on them, and also determine the principal functions of the language. As soon as these issues are considered, it is necessary to create a language syntax so that developers of educational programs can try to implement their use cases by writing scripts in a domain-specific language. Finally, a language interpreter integrated with Microsoft Excel that renders the script code, turning it into the appropriate X-matrix has to be implemented. Thus, this paper logically continues the idea of creating educational programs based on the X-matrix, which needs to be integrated with a domain-specific language to increase the convenience and efficiency of developing and amending the curriculum. Some of the functionality of the language, the basic concepts and entities on which the X-matrix abstractions and map of problem area are based will be presented. Finally, a real code example will be demonstrated for creation of X-matrix based on a university educational program.

II. Review of Relevant Works

The problem in knowledge-intensive areas such as information technology is the rapid development of approaches and technologies that can be radically different from previous cutting-edge options. In accordance with this, the requirements for workers in the industry are changing at the same time. As a result, when updating the programs of any course or university program, it is necessary to quickly prepare new materials. Moreover, the academic disciplines should be oriented to the new needs of employers. The traditional approach to educational program design is inefficient and time-consuming, as it leads to the creation of large works, which takes a lot of time [7]. The traditional approach does not meet the conditions of rapidly changing technologies and the requirements of employers. One of the alternative solutions is a group of methods based on reusable learning objects. Learning object is a small independent unit of knowledge that store certain content that represents them [8]. By breaking down large material into modules, educators can use them in a variety of contexts in different courses. Moreover, learning objects may not be created by course designers since they can use ready-made. As a result, the basis of the program can be reused as a set of learning objects so teachers can focus on the creation or search for new objects. To date, there are many approaches that use reusable abstractions, which are presented in a number of scientific papers [7], [9]–[14]. For instance, in [7], a model was presented that provided software for converting existing electronic materials into a set of reusable learning objects. Automatic transformation is convenient, but it takes time to review the received materials. In addition, such a system does not respond to changes in the scientific and technical world. In [9], an agile approach was published that suggests using versioning of learning objects therefore the work in a team on learning objects becomes convenient, which reduces the time to create them and makes it possible to quickly respond to changing scientific and technical trends. However, it will not be possible to significantly reduce the time since teamwork and a convenient presentation of the learning object are not a radical change. In [10], a recommender system was proposed that allows to customize the recommendations of learning objects for students based on their profiles. Thus, it is possible not only to use learning objects from the bank, but also to personalize them for each student. However, the paper does not consider the flexibility of banks of learning objects, which makes it possible to respond to the appearance of new technologies. In addition, a lot of time will have to be spent on verifying courses and assessing the balance of learning paths in order to understand the quality of how they develop competencies and explain certain technologies. The adaptive approach presented in [11] introduced the concept of an adaptive learning object with dynamic behavior that inherits from a normal static learning object. This approach also provides a solution for personalized learning but does not take into account changing technologies and production needs. In [12], a personal assistant was developed, which is customizable comments to learning objects for in-depth personalization. However, the time spent on creating comments can be disproportionately large if this process is not automated. In addition, new technologies can be described as comments, but this is not the most practical approach to modify learning objects, since you will have to create large comments.

Authoring tool is a popular method of visual course development today, some of which are presented in [13]. These tools are convenient and easy to understand, but they do not evaluate the course or the program created in the editors. In [14], a model of a hybrid learning resource is presented, which is a set of learning objects and contextual parameters that can be used to adapt a learning resource. Nevertheless, the model does not take into account the factor of changing production needs and does not allow to verify the balance of the educational program. At the same time, this model has found a logical continuation, as part of the X-matrix for describing the relationship between courses and competencies.

The presented sources propose the solution of problems related to the rapid, effective the development of courses, but it was not possible to find approaches that could allow presenting educational programs in code so that it would be possible not only to consider the interests of employers relevant to the development of technologies but also to assess the balance of educational programs.

III. Proposed Approach

In this paper, a markup language model is proposed, by means of which it will be possible to formalize an educational program (EP) not only in the traditional form: structure, competencies (learning outcomes). The language will make it possible to describe formally the EP's connections with the scientific methods and technologies that will be taught in the program, as well as with the needs of potential enterprises where EP graduates will implement their learning outcomes. Therefore, the map of subject areas and the X-Matrix proposed in [1] are taken as a basis.

A. *The main functions of the markup language*

The language has two main functions: description of the data forming the model of the educational program in the information space {"Education", "Science and Technology", "Personnel"} and checking the balance of the program.

B. *Description of the problem area matrix*

The first function involves creating a matrix of problem areas and an X-matrix. It is known that the process of forming the X-matrix of the educational program begins with the construction of a matrix of problem areas. Thus, the compilers of the educational program need to initialize this matrix (TechMatrix matrix), which contains cross-cutting technologies vertically (list of CCT class elements) and life cycle stages horizontally (list of LCS class elements). The original matrix is initialized with empty lists, which can be supplemented with new elements. TechMatrix supports sorting operations by fields of CCT's and lifecycle stages (for example, SortBy(CCT,Name)) and filtering operations of lists by their fields (for example, Find(CCT, Weight>2)). Methods for adding an element (Add) and removing an element (Remove, RemoveAt) are defined for lists. CCT is an object with a set of fields: "Name" is the name of the technology, "CrossAll" is a logical variable that determines whether this technology is cross cutting technology for all others, "Desc" is a textual description of the CCT, "Weight" is an integer value that reflects the importance of the technology from the point of view of enterprises. The "CrossAll" field is used as an indicator that this technology covers all the others in the list, so it may not be listed in the list of X-matrix technologies. A lifecycle stage is also an object with the "Name" field.

As soon as all CCT's and life cycle stages have been added to the appropriate lists, it becomes necessary to fill in the matrix. The element of the technical matrix is an object with two fields: "Name" is a description of the requirements for an employee working with the appropriate technology at this stage of the life cycle, "Trj" is a list of educational trajectories that contain this requirement, in the form of trajectory indexes. Access to the element of the technical matrix is carried out by two indexes in the order of "cross cutting technology index" - "life cycle stage index".

C. *Description of the X-Matrix using the markup language*

Having completed the design of the matrix of problem areas, the compilers of the curriculum should begin to work on the X-matrix. When initializing the matrix (Xmat matrix), four empty lists are created, reserved for sequential filling and four cross-matrices. Each list supports the operations of adding an element (Add), deleting (Remove, RemoveAt) and updating (Update) as well as cross-matrices. The list of employers' needs consists of EmpNeed objects. This object has fields: "Name" -the name of the need, "Desc" - a text description of the need. The needs are formulated based on the fields of the matrix of problem areas, so the sequence of stages of creating a curriculum is important. In addition, a list of technologies covered by this curriculum is explicitly formulated from the matrix of problem areas. The "Technology" object also consists of the fields "Name", "Desc". Then lists of learning outcomes and courses are formed sequentially. The structure of both objects "LrnOut" (learning outcome), "Course" is identical and is a set of three fields: "Name", "Desc", "Trj" - a list of identifiers of trajectories that cover this competence or course. Thus, 4 fundamental entities are implemented in the X-matrix: the needs of the employer, technologies, educational competencies and courses.

The interaction between entities is described by four matrices: CrossEmpTechs, CrossEmpLearns, CrossCourseTechs, CrossCourseLearns. The elements of the matrices are objects of the Cross type with two fields: "Weight", "Desc" - the textual justification of the specified Weight value. Although the range of Weight values is the same for all matrices {"", "x", "xx", "xxx"}, this field has a different meaning in each cross-matrix. In CrossEmpTechs, Weight means a degree of technology use for a specific employer need. In CrossEmpLearns, this field shows how competence solves issues related to this need. So, "x" means that competence allows to identify issues in the area of a given need and give general recommendations for their solution, "xx" indicates that competence allows to solve individual issues in the area of a given need, and "xxx" means that competence allows to build organizational solutions to cover the problem fully. In CrossCourseTechs, the Weight field reflects the level at which this technology is used in this course. An empty field means that the technology is not covered by the course, "x" – the technology is taken into account, "xx" – the technology is actively used, "xxx" – additional changes may be made to the technology. The CrossCourseLearns matrix stores information about the depth of teaching. An empty field indicates that the event does not allow to master this competence, "x" - allows to master the competence at the initial level, "xx" - allows to master the competence at the basic practical level, "xxx" - allows to master the competence at an in-depth level (an ability to solve broader tasks, including extending the proposed methods). Thus, a ready-made X-matrix is obtained, representing an educational program that can be flexibly changed for modification and creation of new programs.

In the future, depending on the type of interaction (communication), the corresponding descriptions will be detailed. For example, CrossCourseTechs connections will be described using technology ontologies, as well as language tools for describing the application of the results of scientific and technical activities in the educational process. One of the variants of the markup language is proposed in [1].

D. *The operator basis of the markup language*

The following operators are offered in the language: (1) assignment operator ("="): assigns an object of the same type to a variable of a certain type; (2) comparison operators ("<=", ">=", "<", ">", "==", "is"): allows you to compare two objects of the same type and return a value of type bool.

E. *The function of evaluating the balance of the program*

This function reveals one of the main applied purposes of the markup language: assessing the balance of the program in terms of meeting the expectations of employers, the level of involvement and teaching of scientific and technical means, as well as the correspondence of the levels of competence development to the levels of their implementation in production.

Since the interaction between fundamental entities is described by cross-matrices, this check is a validation for the values of matrix elements. Verification of the correctness of the educational program consists of two processes: verification of compliance with the degree of use of technology and the depth of its teaching and verification of the depth of teaching technology and the level of development of relevant competence. In particular, for the first case, if the discipline is taught at a higher level than employers need (the maximum value of CrossEmpTechs[tech] < the maximum value of CrossCourseTechs[tech]), then the system should display an appropriate message about possible optimization (for example, "Perhaps the level of training in {tech} technology is higher than the employer needs"). An example of validation for the second case is when technology in a discipline is taught at an insufficient level to form a certain competence of the student (CrossCourseTechs[course][tech] < CrossCourseLearns[course][learnOutcome]). Then the system should display the message "It is possible that the depth of consideration of the {tech} technology at the {course} event is insufficient to develop the {learnOutcome} competence at a given level".

F. Syntax for Notepad++ text editor

To facilitate the use of the language, a proposed syntax is installed in the widely used Notepad++ editor. The template for the markup language is available at the link [https://drive.google.com/file/d/1NS6Iahx7peJIuF1ULWsJZ6lGPAR7TV_n/view?usp=sharing].

An example of the language features is shown in Figure 1. In the illustration X-matrix initialization is depicted and its name. Then the main matrix lists are filled with the corresponding objects (for instance, xm.EmpNeeds – a list of employer's needs filled with the objects of EmpNeeds type). Thus, in the example presented all four lists which describe the educational program are initialized.

IV. Application of the Approach in the Educational Program of the National Center for Cognitive Research

The prototype of domain-specific language was tested on an educational program "Big Data and Machine Learning" proposed by National Center for Cognitive Research. The program was reformulated in terms of X-matrix entities and the connections between them were built by the experts. As the program was created, the code in domain-specific language was written where X-matrix was initialized as well as its lists with employer's needs, technologies, learning outcomes and courses. The last piece of code is dedicated to cross entities which bind these entities. The code fragment is shown in Figure 2. The whole code file, the excel file with X-matrix and the interpreter are available at the GitHub repository [https://github.com/ShadowTechiner/DSL_xmatrix.git].

Thus, a correct code was created and given to language interpreter written in Python. As a result, a Microsoft Excel file was created with the corresponding X-matrix (Figure 3). Apart from matrix-generation, the interpreter has a function which performs balance verification. In the illustration the matrix has blue cells warning that the course in the corresponding row constructed is not balanced since the technologies connected with blue cells in south-western corner are not explained in such a way to develop learning outcomes bound with blue cells in south-eastern corner of the matrix.

```
Xmat xm
xm.EmpNeeds=[
            EmpNeed {Name="Development of projects of digital transformations and development of informatization of the enterprise/industry",
                    Desc="The highest priority"},
            EmpNeed {Name="Development of requirements, terms of reference for the design and development / purchase of high-tech software"}
            ]
EmpNeed need = "Conducting scientific research, analytical work using big data technologies and machine learning"

xm.EmpNeeds.Add(need)

Technology big_data = {
                Name="Big Data",
                Desc=None
                }
xm.Technologies.Add(big_data)
Technology ML = {
                Name="Machine Learning",
                Desc="Classic approaches"
                }
xm.Technologies.Add(ML)


LrnOut outcome = {Name="Able to manage the processes of assessing the complexity, labor intensity, terms of work",
                Desc=None,
                Trj=2}
LrnOut important = {Name="Able to apply algorithms of machine learning in order to build effective models",
                Desc=None,
                Trj=1}
xm.Outcomes = outcome
xm.Outcomes.Add(important)

xm.Courses = [Course {Name="Linear Algebra",
                Desc="Fundamentals in linear algebra including matrices, vectors, analytical geometry",
                Trj=5}]
```

Fig. 1. A demonstration of capabilities of the markup language.

```
Xmat xm

xm.EmpNeeds = [ EmpNeed {Name= "The development of requirements, terms of reference for the design and development / purchase of high-tech software"},...]

xm.Technologies = [ Technology {Name="Distributed computing and database technologies"},...]
xm.Outcomes = [ LrnOut {Name="Able to manage analytical work"},...]
xm.Courses = [ Course {Name="Methodology of translational research"},...]

Cross oneX = {Weight = "x", Desc = "Weak connections"}
Cross doubleX = {Weight = "xx", Desc = "Medium connections"}
Cross tripleX = {Weight = "xxx", Desc = "Strong connections"}

//Filling CrossEmpTechs cross-matrix
xm.CrossEmpTechs[0][0] = oneX
xm.CrossEmpTechs[0][1] = oneX
...

//Filling CrossEmpLearns cross-matrix
xm.CrossEmpLearns[0][0] = doubleX
xm.CrossEmpLearns[0][1] = doubleX
...

//Filling CrossCourseTechs
xm.CrossCourseTechs[0][0] = oneX
xm.CrossCourseTechs[0][1] = oneX
...

//Filling CrossCourseLearns cross-matrix
xm.CrossCourseLearns[0][0] = oneX
xm.CrossCourseLearns[0][1] = oneX
...
```

Fig. 2. Shortcut code descriptoin of X-matrix realated to the educational program "Big Data and Machine Learning"

In order to determine why a warning occurred, each blue cell of the south-western corner has a comment describing which learning outcomes caused this warning.

V. Conclusion

In this paper, a markup language model is proposed. This language makes possible to formalize some key concepts of the X-matrix and matrix of problem areas. In the work, the functions of the language, the operator basis, as well as key classes that display entities that can be operated by designers of educational programs were identified. For the developed language, a syntax is proposed that can be installed in the popular Notepad++ text editor. An example of the code implemented in this editor using syntax is given.

In future works, we will develop a language interpreter that will allow us to evaluate the balance of the educational program. This language will also be expanded in three directions. The first of them will consist in the integration of the educational program with the taxonomies of scientific directions, scientific methods, as well as technologies. The second direction is devoted to the integration of descriptions of the educational program and the courses included in it. The tools being developed in this direction should facilitate the solution of issues of filling and updating the training courses of the developed program in terms of content and conducting training events. The third direction will consist in detailing the descriptions of the graduate's competencies and their application to solve production tasks.

Acknowledgment

This work is financially supported by the National Center for Cognitive Research of ITMO University

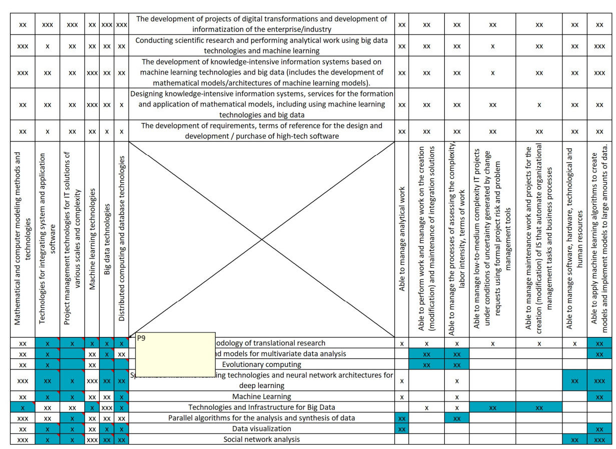

Fig. 3. X-matrix of the educational program "Big Data and Machine Learning" created with domain-specific language

REFERENCES

[1] A. Dukhanov, A. Lutsenko, A. Klimova, and S. Kraev, "A Systemically Synergistic Approach to the Design of Integrative Advanced Training Courses in the Field of Cross-Cutting Technologies," in 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1–9. doi: 10.1109/FIE49875.2021.9637425.

[2] D. Perez-Berenguer and J. Garcia-Molina, "INDIeAuthor: A metamodel-based textual language for authoring educational courses," IEEE Access, vol. 7, pp. 51396–51416, 2019, doi: 10.1109/ACCESS.2019.2911884.

[3] S. Meacham, V. Pech, and D. Nauck, "AdaptiveVLE: An integrated framework for personalized online education using MPS jetbrains domain-specific modeling environment," IEEE Access, vol. 8, pp. 184621–184632, 2020, doi: 10.1109/ACCESS.2020.3029888.

[4] N. Guelfi, B. Jahić, and B. Ries, "TESMA: Requirements and design of a tool for educational programs," Information (Switzerland), vol. 8, no. 1, Mar. 2017, doi: 10.3390/info8010037.

[5] S. M. Reyad, A. Razzaque, S. Badawi, A. Hamdan, R. Khamis, and Abdalmuttaleb Al-Sartawi, "The impact of E-learning on learner knowledge sharing quality," in Multi Conference on Computer Science and Information Systems, MCCSIS 2019 - Proceedings of the International Conference on e-Learning 2019, 2019, pp. 26–32. doi: 10.33965/el2019_201909f004.

[6] A. Dukhanov, L. Gorokhovatsky, S. Koniukhov, and A. Pershutkin, "Object-Oriented Approach for Quick Preparation of Learning Materials in Conditions of Digital Transformations and Rapid Updates of Scientific Results."

[7] Z. Putnik, Z. Budimac, and M. Ivanović, "From Legacy Teaching Material to Learning Objects-Possible Conversion Model Graphs in Space: Graph Embeddings for Machine Learning on Complex Data View project BLEMAT-Advanced Indoor Positioning Systems for fog computing View project From Legacy Teaching Material to Learning Objects-Possible Conversion Model," 2009. [Online]. Available: https://www.researchgate.net/publication/255527762

[8] C. Barritt and F. Lee Alderman Jr, "Creating a Reusable Learning Objects Strategy : Leveraging Information and Learning in a Knowledge Economy."

[9] R. P. Salas, "Reusable Learning Objects: An Agile Approach," in Proceedings - Frontiers in Education Conference, FIE, Oct. 2020, vol. 2020-October. doi: 10.1109/FIE44824.2020.9273947.

[10] F. A. Dorça, R. D. Araújo, V. C. de Carvalho, D. T. Resende, and R. G. Cattelan, "An automatic and dynamic approach for personalized recommendation of learning objects considering students learning styles: An experimental analysis," Informatics in Education, vol. 15, no. 3, pp. 45–62, 2016, doi: 10.15388/infedu.2016.03.

[11] H. Al-Chalabi and H. K. Majeed Al-Chalabi, "ADAPTIVE E-LEARNING OBJECTS MODEL." [Online]. Available: https://www.researchgate.net/publication/341120284

[12] R. M. Harden, I. H. Gessner, M. Gunn, S. B. Issenberg, S. D. Pringle, and A. Stewart, "Creating an e-learning module from learning objects using a commentary or 'personal learning assistant,'" Medical Teacher, vol. 33, no. 4, pp. 286–290, Apr. 2011, doi: 10.3109/0142159X.2011.557104.

[13] A. Gordillo, E. Barra, and J. Quemada, "An easy to use open source authoring tool to create effective and reusable learning objects," Computer Applications in Engineering Education, vol. 25, no. 2, pp. 188–199, Mar. 2017, doi: 10.1002/cae.21789.

[14] K. Bochenina, A. Dukhanov, M. Karpova, and V. Shmelev, "An Approach to Hybrid Learning Resource Design for Training Professionals in Computational Science," in Procedia Computer Science, 2016, vol. 101, pp. 439–448. doi: 10.1016/j.procs.2016.11.051.