Journal of Information Systems Education

Volume 33
Issue 2
Spring 2022

Teaching Tip

Improving Student Performance by Introducing a No-Code Approach: A Course Unit of Decision Support Systems

Hai Wang and Shouhong Wang

Recommended Citation: Wang, H., & Wang, S. (2022). Teaching Tip: Improving Student Performance by Introducing a No-Code Approach: A Course Unit of Decision Support Systems. *Journal of Information Systems Education*, 33(2), 127-134.

Article Link: https://jise.org/Volume33/n2/JISE2022v33n2pp127-134.html

Initial Submission:	4 March 2021
Accepted:	12 May 2021
Published:	15 June 2022

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 33(2), 127-134, Spring 2022

Teaching Tip
Improving Student Performance by Introducing a No-Code Approach: A Course Unit of Decision Support Systems

Hai Wang
Sobey School of Business
Saint Mary's University
Halifax, NS B3H 2W3, Canada
hwang@smu.ca

Shouhong Wang
Charlton College of Business
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
swang@umassd.edu

ABSTRACT

No-code/low-code app development is transforming traditional information system development paradigms. This teaching tip uses an example of course unit of decision support systems (DSS) to demonstrate that introducing no-code system implementation assignments into a course as a replacement for computer programming language exercises can improve student learning. It presents the pedagogical design and the teaching method of no-code DSS implementation. The contents of the pedagogy include key concepts of no-code development, workshops of no-code DSS implementation, and assignments for students. This course unit demands about one-third-credit-hour workload, and can be embedded in a three-credit-hour business course. The preliminary evidence has indicated that the teaching method of no-code DSS implementation is useful for business students.

Keywords: No-code development, Decision support system, Computer programming

1. INTRODUCTION

After years of development of large-scale enterprise information systems in business organizations, such as ERP, CRM, and SCM systems, app development has become the mainstream in the software development sector (IBISWorld, 2019). An app is a small-scale computer program or software application designed to run on mobile or desktop devices. Apps are widely used for personal productivity assistance, games, and e-commerce. The education literature about apps has reported the use of apps for effective mobile education and mobile game app development to improve student engagement (Dekhane et al., 2013; Yan et al., 2014). On the technical side, no-code/low-code app development is transforming traditional information system development paradigms and has become a highly regarded app development paradigm in many leading organizations (Fryling, 2019; Statista, 2019). A no-code development platform allows developers to create application software using graphical user interfaces and templates without the technical skills of traditional computer programming languages (Lee et al., 2020). Information Systems professionals can use no-code app development platforms to design app prototypes rapidly for advanced apps (Hyun, 2019).

This pedagogical study investigates how the no-code system development approach can be relevant in business courses, and how introducing no-code system implementation assignments into a course as a replacement for computer programming language exercises can improve student learning. The paper reports an example of teaching the no-code method for decision support systems (DSS) implementation in a business course.

DSS is an essential subject in the information systems field (Topi, 2019; Topi et al., 2010) and is important for every business student as discussed in the next section. In comparison with large-scale enterprise information systems such as ERP, CRM, and SCM systems which are developed by using the typical software development life cycle (SDLC) model, DSS are small-scale systems which are developed by using rapid applications development (RAD) approaches (Arnott, 2004). Yet, the major tools for DSS implementation have been computer languages (Apigian & Gambill, 2019; Ragsdale, 2001). This study demonstrates that the no-code method is a good alternative technique for DSS implementation as well as

a useful approach to teaching DSS concepts to business students.

This paper describes the pedagogical design and implementation of a course unit of DSS with no-code development for business students. The rest of the paper is organized as follows. The next section provides an overview of the background of the study. The three subsequent sections explain the innovation of teaching of the no-code method for the DSS unit, present the evidence of the usefulness of the proposed approach based on preliminary practices, and discuss why teaching no-code system development can be useful for business students to expand their knowledge set and skill set. The final section reflects upon how this paper contributes to information systems education.

2. BACKGROUND OF THE STUDY

2.1 Basics of No-Code App Software Development

A no-code development platform allows developers to create application software using graphical user-interfaces and templates instead of traditional computer programming languages (Wang & Wang, 2021). Figure 1 shows an example. To use Microsoft Visual Basic for Applications (VBA) to load an Excel spreadsheet for a DSS, the DSS developer must specify detailed steps for the computer to select a range of the needed data, to copy the selected part, to select the destination location, and to paste the needed data for further process. On the other hand, using the Microsoft Power Apps (2020) no-code environment, the DSS developer can generate a screen that is linked to a needed Excel spreadsheet with just a few clicks of buttons and icons.

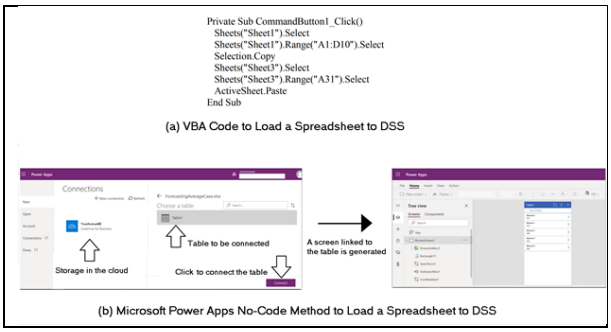


Figure 1. Comparison of Code versus No-Code Methods

In comparison with general tools of computer programming languages, a no-code app development platform would be application-specific and less flexible. Each no-code app development platform has its own configuration and specific templates, and the developed apps have limited expandability and maintainability in the software development life-cycle. Nevertheless, a good no-code development platform allows end-users to develop apps to meet wide-ranging business needs. More importantly, information systems professionals and entrepreneurs can use no-code app development platforms to design app prototypes for sophisticated apps.

2.2 No-Code Development Platform

The no-code business app development platform used in the present case is Microsoft Power Apps (2020) because Microsoft Power Apps is a part of the Microsoft Office 365 environment which is widely available for students and provides seamless connections to Microsoft Access and Excel for DSS for small business organizations. Microsoft Power Apps is one of many commonly used no-code development platforms on the software market, such as Salesforce Lightning (https://www.salesforce.com/lightning/), Appy Pie (https://www.appypie.com), Bubble (https://bubble.io), and DaDaBIK (https://dadabik.com). It is easy to integrate data from various sources, and has mobile support. Nevertheless, Microsoft Power Apps' capacity for data processing and workflows is weak in comparison with many other no-code development platforms. Also, many of its functionalities are not straightforward for beginners. As Microsoft Power Apps is a member of Microsoft Office 365 which is licensed in our institutions, and has seamless connections with Microsoft Excel and Microsoft OneDrive for Business, it was selected for this course despite these drawbacks.

2.3 Decision Support Systems

In today's global economy, business organizations count on business intelligence and analytics to maintain or increase business growth (Amabile1 et al., 2013; Ortiz & Lombardo, 2009). Business students need to know DSS to understand how business intelligence and big data analytics can be relevant to business decision making (Arnott & Pervan, 2014; Guarda et al., 2013). DSS are particularly important to business education because organizations commonly overlook decision-making beyond day-to-day operations (Benson & Dresdow, 2015; Chaudhry et al., 1996; Jamaluddin & Dickie, 2011; Katzman et al., 2009; Newell & Marabelli 2015).

A DSS is an information system that supports business or organizational decision-making activities. The major emphasis of DSS is to help decision makers solve semi-structured decision-making problems which are elements of ill-structured decision-making problems. An ill-structured problem may be rapidly changing and not easily specified in advance, and tends to have a "satisfactory solution." Simon's (1969) decision-making model describes the common decision-making process which has four phases: intelligence, design, choice, and implementation. As a decision-making process to deal with ill-structured problems in business is problem-specific, commercialized DSS for business are rare. Business organizations must develop their own DSS to help the decision makers solve semi-structured decision-making problems which are transformed from unique ill-structured decision-making problems. The easier-to-use the available end-user DSS development tools are, the more decision-makers (management) are able to develop their own DSS without depending on the IT staff. No-code and low-code development, end-user computing, and DSS development have synergistic relationships; that is, the innovation of no-code and low-code development can empower end-users to develop and adapt systems (Lieberman et al., 2006) which would in turn improve the end-user computing satisfaction (Domínguez-Escrig et al., 2018) as well as end-users' capabilities of DSS development (Nygård et al., 2020).

Early researchers in DSS (Keen, 1980; Sprague, 1980) have argued that a DSS would change the decision-making process as the decision-maker learns a new way to make decisions. For example, adequately designed graphical user-interfaces of DSS can be more effective than narratives (text data) and numerical

tables for decision-makers to process information (Montazemi & Wang, 1988). The graphical user-interface of DSS allows the decision maker to play an important role in an interactive development process, allowing the decision-maker to choose from a narrowed list of alternatives, and to carry out "what-if" sensitivity analyses.

Recently, Sharda et al. (2020) use decision support and DSS as a conceptual starting point for the understanding of decision support and then describe business intelligence and analytics by building on DSS. Olavsrud (2020) considers that DSS are a subset of business intelligence aimed at helping organizations make informed business decisions based on vast troves of analyzed data. On the other hand, Power and Heavin (2017) note that businesses organizations still need information systems for decision support, but for a variety of reasons are loath to use the term DSS, describing it as a legacy term that might even refer to earlier failed projects. Watson (2018) argues that technologies for DSS development are continuously updating to fit the changing business environment; however, the traditional framework for developing DSS (Sprague, 1980) is still relevant in the big data era.

2.4 The Course Unit of DSS

Because of the wide-ranging business subjects and a restricted number of courses in a business program, an independent DSS course seems to be infeasible and unnecessary. Not many business programs offer standalone three-credit DSS courses these days. In the present study, a course unit of DSS is included in a three-credit business elective online course Information Technology for Small Business in the AACSB accredited college of business of a medium-sized Higher Research Activity (R2) university in the US. The course is offered once or twice each year for students in all majors who have taken the prerequisite introductory information systems course. The typical sizes of the course are about fifteen. The subject of DSS is taught in this course because small business organizations face diversified decision-making problems frequently but often overlook the importance of DSS. More importantly, as DSS are always problem-specific, small business organizations often have to develop DSS without IT professional assistance. The course covers seven subjects: end-user computing and systems architectures for small business, off-the-shelf software systems for small business, DSS, business intelligence for small business, social media for small business, cloud computing for small business, and open source software for small business. Each of the above subjects represents a particular type of information technology applications.

The central teaching/learning theme of this course is instructional information technology planning for small business; that is, students learn information technologies in the context of small business management through projects of small business information technology planning. The course requires students to conduct their real-world projects. Each student selects a local small business organization for the project to design a comprehensive business plan that recommends appropriate information technology solutions for the selected small business organization to improve the business strategic practices. Each student is required to have general knowledge about the selected small organization based on their work experiences and observations, but is not required to receive permission and/or any private information of the selected small business organization for the project. A project includes two parts: a general plan that describes how the learned seven types of information technologies can be applied to the selected small business organization, and an in-depth plan of detailed steps of implementation of one or two of the seven types of information technologies for the small business organization. The assignments for each subject require students to prepare the project for the related subject. Upon the completion of all assignments, students should be able to develop the course project. The online business elective course has been offered six times since it was developed four years ago.

The DSS unit teaches key concepts of DSS and provides illustrative examples of DSS for small business organizations. Before this no-code DSS implementation method was introduced, the unit taught Microsoft Excel Visual Basic for Applications (VBA) for DSS implementation. Microsoft Excel is a commonly used tool in small business organizations, and the embedded VBA are relatively easy to use in comparison with other computer languages. The DSS unit included an illustrative example of DSS implemented by using VBA. It also included the basics of VBA programming materials so that students would be able to implement a simple DSS beyond the textbook example for their course project. Two assignments are used for the assessment of this unit. The first assignment asks students to identify an ill-structured decision-making problem for the small business organization, to decompose it into structured decision-making problems, to discuss possible database and model bases used to solve the structured decision-problems, and to articulate the role of the decision-maker in solving the decision-making problem using "what-if" trials. The second assignment asks students to sketch the user-interface screens for the prospective DSS, and to explain how the user-interface supports the decision-maker to make "what-if" trials using the database and model bases discussed in the first assignment.

3. THE NEW NO-CODE APPROACH TO DSS IMPLEMENTATION

The new tool of DSS implementation taught in the DSS unit is Microsoft Power Apps. The DSS unit delivers the following essential concepts of no-code DSS implementation for business students.

3.1 A Diagramming Tool for No-Code DSS Implementation

Diagramming tools are visual languages for communication and documentation. In the information systems field, diagrams have been used as effective tools for systems analysis and design in all types of systems development activities. For example, structure diagrams, object-oriented diagrams, entity-relationships diagrams, network diagrams, and data flow diagrams are widely used in different areas of information systems analysis and design. The Unified Modeling Language (UML) (Booch et al., 2005) provides various types of diagramming tools for software systems development. The present DSS unit raised an important issue: What diagramming tool can be used to support no-code DSS development for effective collaboration and documentation? After a thorough review of the major diagramming tools in UML, including object diagram, activity diagram, interaction overview diagram, sequence diagram, and use case diagram, we decided to use so-called DSS design diagram, a variation of UML use case

diagram with explicit images and icons in the context of DSS development, to facilitate no-code DSS development. Students spend about an hour learning and drawing DSS design diagrams for discussion during the design process.

A formal DSS diagram includes screenshots of user-interface, database and model base needed, and "what-if" trial results. DSS diagrams can describe the following properties of DSS using the corresponding symbols.

- Database (Symbol: Data icon)
- Model base (Symbol: Chart icon)
- User-interface (Symbol: Screenshot)
- Interaction between the user-interface and database/model base (Symbols: Line and text notation for data flow)

As the users are default actors of DSS, the actor notation commonly used in the UML use case diagrams is not needed. The iterative decision-making process is represented by all linked notations in the diagram. The next section exhibits an example of DSS diagram for no-code DSS development.

3.2 Teaching Materials

As no suitable teaching material for no-code DSS implementation can be found, a new set of teaching materials has been developed. The teaching materials include a teaching note, lecture PPT, and video clips to present the core of the no-code DSS implementation method. The teaching note provides a tutorial of Microsoft Power Apps along with the artifacts of an illustrative DSS example in Microsoft Power Apps, step-by-step instructions of DSS implementation in the aspects of DSS elements, and four Excel spreadsheets for the model base and database used for the DSS example. Following the tutorial, students are able to re-produce the example by themselves to learn the implementation of DSS using the no-code development approach. The teaching note is oversized for a journal paper, but the note and other related course materials posted in the learning management system for students can be obtained from the authors upon request. The DSS unit in the current form demands about one-third-credit-hour workload and can be embedded and integrated in a three-credit-hour business course.

The example of no-code DSS implementation is a mobile app for salesforce budgeting decision-making. Budgeting is an ill-structured decision problem because it involves many uncertain factors, such as future sales, workforce wages, and available financial resources. In this example, the salesforce budgeting decision-maker uses sales data, sales forecasting model, and salesforce foresting model to make "what-if" trials to reach a satisfactory decision. The app accesses four Excel tables to represent the two models and the corresponding database used in the DSS. The database has monthly sales data and salesforce expenses data. One of the two models displays a trend of monthly sales as well as estimated sales in the coming month in different scenarios (e.g., best, normal, and worst cases). The other model indicates the no-linear relationship between monthly sales and salesforce expenses which include salesperson salaries and overhead expenses. The Excel tables are stored on OneDrive for Business. The scenario of decision-making in this DSS example is described as follows.

The decision-makers (i.e., budgeting managers), are able to retrieve the Excel tables to view or revise the sales records, and is allowed to click on buttons on the mobile device to view statistical graphs of sales analysis. They then retrieve the other Excel table to view the non-linear relationship between sales and salesforce expenses which include salespersons' salaries and overhead expenses. They are able to leap across the display screens to reach their decision based on their judgement on the current situation. They can input trial data of estimated sales using dropdown menus, and perform a "what-if" analysis to come up with a "satisfactory" estimated salesforce expense for the budgeting. In the construction aspect, the example introduces basic elements of the user-interface, such as Screen, Label, Text, Button, Chart, Dropdown, and others. It demonstrates the use of parameters for these objects. Figure 2 is a DSS design diagram that depicts the design of the DSS example of salesforce expenses budgeting.

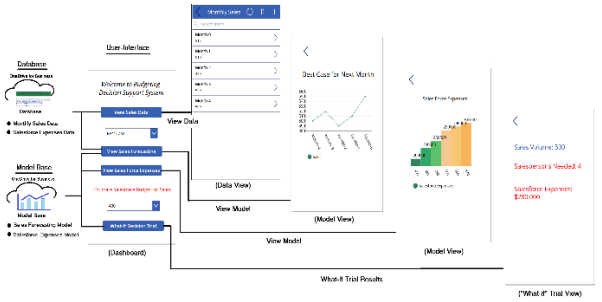


Figure 2. DSS Design Diagram of the Example of Salesforce Budgeting

As shown in Figure 2, the database and model base used in the DSS contain sales data, salesforce expenses data, sales forecasting model, and salesforce expenses model. The user-interface includes dashboard, data view, model view, and "what-if" trial view. The dashboard of the DSS user-interface can activate these views through buttons and dropdown menus. The user is able to navigate the data and models and to make "what-if" trials to reach a decision for the salesforce expenses budgeting. The key learning points of this example include:

- The general structure of DSS that includes database, model base, and user-interface
- The nature of decision-making for ill-structured decision problems
- The decision-maker's role and decision-making process
- The environment of Microsoft Power Apps for DSS construction, and basic elements of user-interface design, such as Label, Button, Dropdown, and Chart
- Use of Excel in the cloud to store databases and model bases in the cloud
- Connection to One Drive for Business
- Data presentation and graph design for data visualization for decision process in Microsoft Power Apps

3.3 Core Competency Components of the DSS Unit

The learning goal of this DSS unit is to help students to expand the knowledge set of DSS and to develop DSS implementation technical skills. The learning objectives of this DSS unit are consistent with the learning objectives of the course. Nevertheless, this DSS unit has its own assessment instruments and rubrics which are different from that of other units of the course. The assessment instruments of this DSS unit include two required assignments for every student and an optional DSS

project for any students who choose the DSS topic for their course project. The first assignment asks students to discuss an ill-structured decision-making problem the small business organization needs to deal with, how the ill-structured decision-making problem can be decomposed into structured problems to solve, what data are needed, what models are involved, and the role of the decision-maker. The second assignment is a design of the user-interface for the DSS discussed in the first assignment. Figure 2 is a typical example of design of DSS user-interface for this assignment. Specifically, students need to describe the following components of graphical DSS user-interface.

- The main menu screen with command buttons to activate subsequent screens and what-if trials, dropdown menus for inputs for what-if trials, and text boxes for what-if results
- Screens of data visualization
- Screens of model visualization
- Screens of intermediate results of what-if trials
- A DSS project is assessed in three aspects of competency: decision-making problem identification and formulation, DSS scope, and self-regulation, as described below

3.3.1 Decision-Making Problem Identification and Formulation. Decision-making problem identification is the thinking ability of business decisions. The real-world small business organization as the background of the student's project may not have clear definitions of decision-making problems or a well-defined "what-if" decision-making process that are ready for the student to develop the project. Students must go beyond the teaching materials to identify the right target decision-making problem for their course project creatively, and decompose the ill-structured decision-making problem into semi-structured decision-making problems. Students are asked to develop a business app in the context of DSS to solve the formulated structured decision-making problems.

3.3.2 DSS Scope. The scope of constructed DSS indicates the ability of being effortful and creative. Students are requested to set high standards, challenges, novelty, and achievable goals on their own to demonstrate motivation. DSS scope is measured in two aspects: database and model base support, and user-interface design. The decision-making problem to be solved should be clearly defined by using a DSS diagram to indicate the database, model base, and user-interface screens for interactions between the decision-maker and the system. The DSS diagram of a project must be checked by the instructor to ensure that the identified problems are meaningful and achievable by using Microsoft Power Apps. The documentation of the project should be professional as a tool of communication of decision-making problem solving.

3.3.3 Self-Regulation Skills. Self-regulation skills are the ability of self-monitoring and learning from positive experiences as well as mistakes. Learning no-code DSS implementation strongly depends upon individual learning styles and unique characteristics of cognition. The DSS unit promotes the development of conscious and quick learning of new concepts and techniques. Uniform lectures, general hints, and illustrative examples are the start points of learning, but students are required to find the best way of learning no-code DSS implementation by fitting their own learning strategies and thinking structures. The submission time of the project indicates whether the student can master time management.

The competency components as the measurable learning objectives of the DSS unit are summarized in the rubrics for assessment, as shown in Table 1. All of the ten factors in Table 1 are graded based on the instructor's subjective judgment depending upon the specific case and then averaged to determine a student's grade of the assignment or the project.

4. EVIDENCE

As discussed in the previous section, the assessment for the DSS unit includes two required assignments and one optional DSS project. The majority of students do not choose DSS for their course projects to avoid technical work. Before the new no-code DSS implementation method was introduced, only two of the total seventy-nine business students chose to use VBA to develop DSS for their course projects, indicating that computer programming is a challenging subject for business students (Watkins & Hufnagel, 2007). After introducing the new no-code DSS implementation method, two of ten students in this course chose the DSS topic for their course projects. The upsurge of the participation rate of optional DSS projects might indicate the increase in students' interest in DSS.

The DSS projects and assessment data were collected and analyzed. A comparison of projects of the two groups indicates that the difference in project quality of the two groups is insignificant, although the VBA group has slightly higher grades than the no-code app group. To investigate the effect of introducing the no-code DSS implementation method on the overall learning of the DSS unit, samples of assignments of two groups with the two DSS implementation methods were collected. The first group of thirteen students learned the VBA method in two different semesters before the no-code DSS implementation method was introduced, and the second group of ten students learned the no-code DSS implementation method. As the entire course settings in these semesters were slightly different, the differences in student performance of the course as a whole were not investigated.

An analysis of the grades of the two assignments and the projects of the two groups was conducted. The assessments for all students and all treatments were graded by the same instructor using the same grading rubrics, but in the different semesters when the assessments were carried out. As the sample size of DSS projects and assignments in this preliminary study was extremely small and the collected data were not robust, the analysis of comparison of the two groups' performance was preparatory. Nevertheless, the analysis reveals that the difference in the quality of the projects of the two groups is not statistically significant, but the differences in the quality of the two assignments of the two groups are statistically significant at the 0.05 level, indicating that the no-code group performed better than the VBA group. The findings confirm the authors' natural observations in teaching that, in comparison with the no-code DSS implementation method, VBA seems to be less interesting and more time consuming to learn for the present business students population, which in turn adversely affects students' learning of the concepts of DSS and DSS implementation. As VBA and no-code app development are two different methods of DSS implementation, in terms of programming skills, the intersection of the two methods is

small. The two methods cannot be practically compared based on their products of DSS implementation projects, but can make differences in student learning of DSS concepts.

At the end of the DSS unit, students were asked to complete a short questionnaire anonymously. Table 2 summarizes the questions and the responses from seventeen students who have learned the no-code DSS implementation method. The questions used a 5-point Likert scale. Table 2 indicates the favorable level of overall student satisfaction with the no-code DSS implementation method for the DSS unit.

	3. Excellent	2. Good	1. Poor
Decision-making problem identification and formulation	° The ill-structured decision-making problem to be solved is clearly defined ° Decomposition of the ill-structured problem into structured decision-making problems is meaningful	° The ill-structured decision-making problem to be solved is fairly clear ° Decomposition of the ill-structured problem into structured decision-making problems can be improved	° The ill-structured decision-making problem to be solved is not defined ° Decomposition of the ill-structured problem into structured decision-making problems is missing
DSS scope	° Excellent database support ° Excellent model base support ° Excellent user-interface design/ ° Excellent documentation	° Good database support ° Good model base support ° Good user-interface design ° Fairly good documentation	° Missing database support ° Missing model base support ° Poor user-interface design ° Poor documentation
Self-regulation (Self-monitoring)	° Conscious learning strategies ° Quick learning new concepts and techniques ° Counting on minimum assistance ° Thorough self-testing	° Fairly good learning strategies ° Good learner of new concepts and techniques ° Occasional needs for assistance ° Good self-testing	° No or poor learning strategy ° Poor ability to learn new concepts and techniques ° Poor utilization of assistance ° No self-testing

Table 1. Rubrics for Assessment of No-Code DSS Implementation

Question	Percentage of Students Who Agree or Strongly Agree
1) Knowledge of no-code DSS implementation is useful for business students	91%
2) The DSS unit with no-code DSS implementation enhances business students' knowledge set and skill set	89%
3) The techniques introduced in this DSS unit are not difficult to learn	78%
4) The delivered DSS unit with the no-code DSS implementation method meets your expectation of business study	82%

Table 2. Summary of Students' Feedback

The observations of evidence were preliminary. The assessment of student performance conducted by the authors might involve biases. As no pedagogical report of teaching methods of no-code DSS implementation can be found at this point, any further comparative assessment of the DSS implementation method has not been conducted. Potential subjects for future investigations include more analyses of the design of the materials in this subject and additional assessments of learning outcomes.

5. DISCUSSION

When a DSS unit is taught by using the traditional DSS implementation methods of computer programming languages, the unit is usually hosted in an information systems course. The new DSS unit with no-code app development can be integrated in many business courses. Potential host courses of the DSS unit with no-code app development could be any business courses related to decision-making in production and operation management, marketing, finance, or human resource management.

There is little doubt that the use of computer programming languages would discourage many business students to learn the DSS unit. Although the concepts of DSS and implementation of DSS have different emphases, the learning processes of the two parts are not disconnected. In fact, the design of DSS and the implementation of DSS are two steps of an iterative process of DSS development. When students are unable to fully understand implementation due to the difficulties with computer languages, they would lose interest in the subject of DSS. On the other hand, once students can visualize the implementation of the example DSS, they would be engaged to learn more about the concepts of DSS for their own examples of DSS.

The Microsoft Power Apps environment has good features of user-interface development. In comparison with computer languages, Microsoft Power Apps is an ideal tool for data and model presentations without coding, and any business students who have basic skills in Excel are able to learn and use this no-code software development approach. However, a DSS must have a core component of "what-if" trial function which must contain more or less programming features such as action

instructions for the computer, the syntax of functions of the user-interface elements, and if-logics. The design of "what-if" trials highly depends on the process of specific decision-making, and demands the DSS developer to have a deep understanding of the decision-making problem. Upon the completion of this no-code DSS implementation method, students understand that no-code DSS implementation does not mean "no-programming DSS implementation." In fact, the cut-line between "no-code" and "low-code" can never be clear. During the DSS implementation process, the DSS developer is actually performing a programming task for "what-if" trials that demand more thinking skills than technical skills.

6. CONCLUSIONS

No-code/low-code system development is transforming information systems development paradigms. This paper presents an example of teaching no-code method for business students to expand their technical skill set and knowledge set in DSS implementation because decision-making and DSS are important subjects in business programs. Traditionally, DSS implementation tools are computer languages which are not business students' major interest. In our experience, the difficulties in using computer languages discourage business students to learn DSS. To address this problem, a teaching method of use of no-code business app development for DSS implementation has been designed and introduced in our business programs. The DSS unit provides a tutorial of the no-code DSS implementation method with Microsoft Power Apps, and includes an illustrative DSS app example. The DSS unit expects about one-third-credit-hour workload, and can be embedded in a three-credit-hour business course.

The paper presents the original pedagogical design and contents of no-code DSS implementation for business programs. The preliminary study of student performance evaluations has indicated students' positive learning experiences and overall satisfaction with the no-code method. By exercising the no-code DSS implementation examples, students had no difficulty in learning the basics of no-code DSS implementation within a short timeframe. The progressive nature of the no-code app development accommodates differing levels of preparation for learning no-code and sets the stage for students to progress to advanced levels on their own. The design and delivery of the no-code DSS implementation method have demonstrated that knowledge about no-code DSS implementation is practicable and very useful for business students to study DSS. The paper can be valuable for instructors who wish to incorporate a practical no-code method in their business courses.

7. ACKNOWLEDGEMENTS

The comments of the Associate Editor and four anonymous reviewers have contributed significantly to the revisions of the paper.

8. REFERENCES

Amabile1, S., Laghzaoui, S., Peignot, J., Peneranda, A., & Boudrandi, S. (2013). Business Intelligence Practices for Exporting SMEs. *International Business Research*, 6(2), 101-111.

Apigian, C. H., & Gambill, S. E. (2019). Are We Teaching the IS 2009 Model Curriculum? *Journal of Information Systems Education*, 21(4), 411-420.

Arnott, D. (2004). Decision Support Systems Evolution: Framework, Case Study and Research Agenda. *European Journal of Information Systems*, 13(4), 247-259.

Arnott, D., & Pervan, G. (2014). A Critical Analysis of Decision Support Systems Research Revisited: The Rise of Design Science. *Journal of Information Technology*, 29(4), 269-293.

Benson, J., & Dresdow, S. (2015). Design for Thinking: Engagement in an Innovation Project. *Decision Sciences Journal of Innovative Education*, 13(3), 377-410.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language User Guide* (2nd ed.). Addison-Wesley Professional, Hoboken, NJ.

Chaudhry, S. S., Salchenberger, L., & Beheshtian, M. (1996). A Small Business Inventory DSS: Design, Development, and Implementation Issues. *Computers and Operations Research*, 23(1), 63-72.

Dekhane, S., Xu, X., & Tsoi, M. Y. (2013). Mobile App Development to Increase Student Engagement and Problem Solving Skills. *Journal of Information Systems Education*, 24(4), 299-308.

Domínguez-Escrig, E., Broch, F. F. M., Lapiedra, R., & Chiva, R. (2018). Promoting Radical Innovation Through End-User Computing Satisfaction. *Industrial Management & Data Systems*, 118(8), 1629-1646.

Fryling, M. (2019). Low Code App Development. *Journal of Computing Sciences in Colleges*, 34(6), 119.

Guarda, T., Santos, M., Pinto, F., Augusto, M., & Silva, C. (2013). Business Intelligence as a Competitive Advantage for SMEs. *International Journal of Trade, Economics and Finance*, 4(4), 187-190.

Hyun, C. Y. (2019). Design and Implementation of a Low-Code/No-Code System. *International Journal of Advanced Smart Convergence*, 8(4), 188-193.

IBISWorld. (2019). *Smartphone App Developers Industry in the US: Market Research Report*. https://www.ibisworld.com/united-states/market-research-reports/smartphone-app-developers-industry/

Jamaluddin, A., & Dickie, C. (2011). Decision-Making Related to Business Growth: Malay Small Businesses in Selangor. *International Journal of Business and Management*, 6(10), 284-296.

Katzman, B. E., Verhoeven, P., & Baker, H. M. (2009). Decision Analysis and the Principal-Agent Problem. *Decision Sciences Journal of Innovative Education*, 7(1), 51-57.

Keen, P. (1980). *Decision Support Systems: A Research Perspective*. Center for Information Systems Research (No. 54), Alfred P. Sloan School of Management (Working paper No. 1117-80), Cambridge, MA. https://dspace.mit.edu/bitstream/handle/1721.1/47172/decisionsupports1980keen.pdf?sequence=1&isAllowed=y

Lee, E., Ross, J., & Kramer, J. (2020). Teaching on the Front End: Gathering All Educators Interested in Web and Mobile Design and Development. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (*SIGCSE'20*). https://doi.org/10.1145/3328778.3372512

Lieberman, H., Paternò, F., Klann, M., & Wulf, V. (2006). End-User Development: An Emerging Paradigm. In Lieberman, H., Paternò, F., & Wulf, V. (Eds.), *End User Development* (pp. 1-8). Springer, Dordrecht, Netherlands.

Microsoft Power Apps. (2020). Microsoft Power Apps. https://powerapps.microsoft.com/en-us/

Montazemi, A., & Wang, S. (1988). The Impact of Information Presentation Modes on Decision Making: A Meta-Analysis. *Journal of Management Information Systems*, 5(3), 101-127.

Newell, S., & Marabelli, M. (2015). Strategic Opportunities (And Challenges) of Algorithmic Decision-Making: A Call for Action on the Long-Term Societal Effects of "Datification". *Journal of Strategic Information Systems*, 24(1), 3-14.

Nygård, H., van Beest, F. M., Bergqvist, L., Carstensen, J., Gustafsson, B. G., Hasler, B., Schumacher, J., Schernewski, G., Sokolov, A., Zandersen, M., & Fleming, V. (2020). Decision-Support Tools Used in the Baltic Sea Area: Performance and End-User Preferences. *Environmental Management*, 66(6), 1024-1038.

Olavsrud, T. (2020, May 29). Decision Support Systems: Sifting Data for Better Business Decisions. *CIO*. https://www.cio.com/article/3545813/decision-support-systems-sifting-data-for-better-business-decisions.html

Ortiz, R. F., & Lombardo, G. F. (2009). Influence of the Capacities of Top Management on the Internationalization of SMEs. *Entrepreneurship and Regional Studies*, 21(2), 131-154.

Power, D. J., & Heavin, C. (2017). *Decision Support, Analytics, and Business Intelligence*, 3/e. Business Expert Press, New York, NY.

Ragsdale, C. T. (2001). Teaching Management Science With Spreadsheets: From Decision Models to Decision Support. *INFORMS Transactions on Education*, 1(2), 68-74.

Sharda, R., Delen, D., & Turban, E. (2020). *Analytics, Data Science, & Artificial Intelligence*, 11/e. Pearson, London, UK.

Simon, H. A. (1969). *The Sciences of the Artificial*, 1st ed. MIT Press, Cambridge, MA.

Sprague, R. (1980). A Framework for the Development of Decision Support Systems. *MIS Quarterly*, 4(4), 1-25.

Statista. (2019). Mobile App Usage: Statistics & Facts. https://www.statista.com/topics/1002/mobile-app-usage/

Topi, H. (2019). Reflections on the Current State and Future of Information Systems Education. *Journal of Information Systems Education*, 30(1), 1-9.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F. Jr., Sipior, J. C., & de Vreede, G. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*, 26, Article 18.

Wang, S., & Wang, H. (2021). A Teaching Module of No-Code Business App Development. *Journal of Information Systems Education,* 32(1), 1-8.

Watkins, A., & Hufnagel, E. M. (2007). Video Vignettes: Teaching Computer Programming to the MTV Generation. *Decision Sciences Journal of Innovative Education*, 5(2), 391-395.

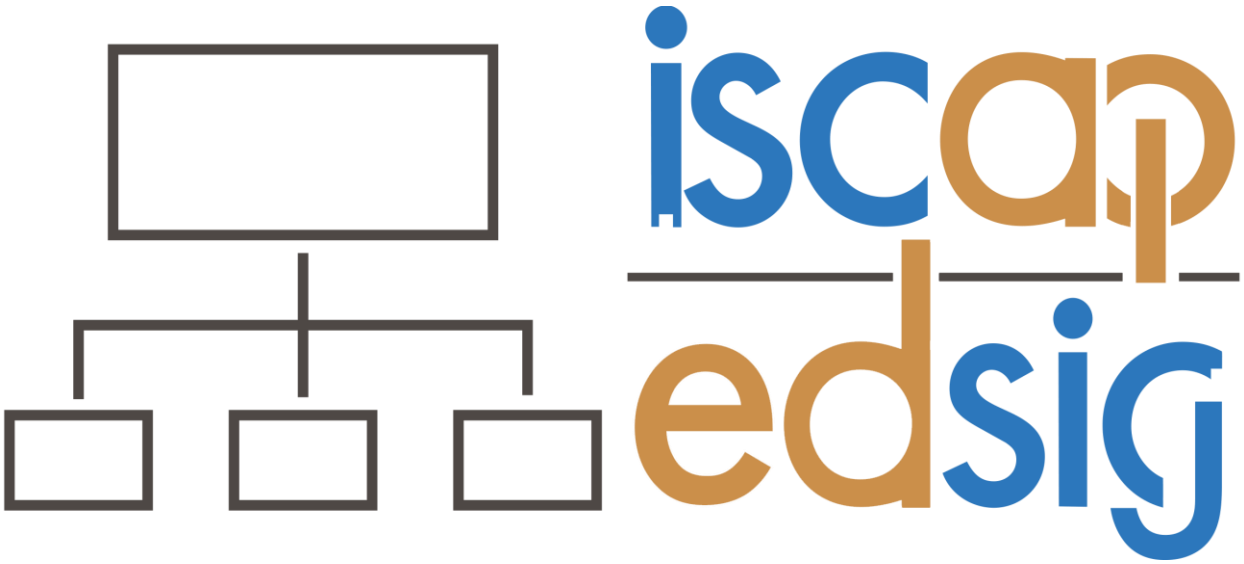
Watson, H. J. (2018). Revisiting Ralph Sprague's Framework for Developing Decision Support Systems. *Communications of the Association for Information Systems*, 42, Article 13.

Yan, G., Rawat, D. B., Shi, H., & Alnusair, A. (2014). Developing and Applying Smartphone App in Online Courses. *Journal of Information Systems Education*, 25(2), 149-159.

AUTHOR BIOGRAPHIES

Hai Wang is a professor in the Department of Finance, Information Systems, and Management Science at Sobey School of Business of Saint Mary's University. He received his B.Sc. (1995) in Computer Science from the University of New Brunswick, and his M.Sc. (1997) and Ph.D. (2004) in Computer Science from the University of Toronto. His research interests are in the areas of software engineering, database management, knowledge management, and e-commerce. Dr. Wang has published more than 90 research articles.

Shouhong Wang is a commonwealth professor of management information systems at University of Massachusetts Dartmouth. He received his PhD in Information Systems from McMaster University. His teaching and research interests include innovative teaching, business intelligence, semantic networks, and knowledge management. He has published over 140 papers in academic journals and several books in the MIS area.



Information Systems & Computing Academic Professionals

Education Special Interest Group

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 2574-3872