Automating Academic Workload Management: A Centralized Solution for Students and Planners

Benjamin E. Rouse & Jeffrey D. Filiberto
Southern Adventist University,
PO Box 370, Collegedale,
TN 37315-0370, USA
Email: benjaminrouse@southern.edu, jeffreydf@southern.edu

***Abstract*—Having the ability to manage academic tasks and on campus planning are two critical aspects of a college campus. This process is often fragmented and painstaking. Students are often burdened with manually tracking assignments, while event planners lack access to clear and comprehensive workload data to effectively schedule events. Assignment Hub aims to create a better solution to this problem that is automated and easy to set up, starting with a centralized database and data tool, using OpenAI's API to automatically extract and organize assignment data from syllabi. We will provide students with a task management tool while also offering event planners a visual representation of the student body's academic workload across campus. Assignment Hub will significantly streamline task management and workload visualization, benefiting both students and planners for schools with SAU being our case study.**

I. Introduction

Even with syllabi being the center of nearly all university classes, the opportunity to harness the power of LLMs to ease the process of collecting data accurately from those syllabi (at a large scale) seems to have gone untouched. From our perspective, a twofold problem can be solved once this opportunity has been addressed and adequately developed. On one hand, students rely on inputting their assignments manually and/or resorting to their learning management system's built-in (and often lackluster) homework management tools. On the other hand, on-campus planners lack the resources to pull workload data from syllabi at a usable scale due to time and manpower deficits.

Our research and development aims to design, create, test, and deploy a system that provides the tools necessary to face these challenges head on. By doing so, Assignment Hub will significantly streamline the work of students and planners, creating the foundation for more campus tools to be implemented at scale.

II. Problem Statement

A. Overall problem this project is solving

After preliminary research was done, there was found to be no existing tools for students to easily access assignments in adjustable calendar form from only a syllabus without manually doing so. We also found that there was no tool for planners to effectively visualize the student workload using syllabi at a large scale. There are various resources that aid in assignment coordination, but there do not seem to be any that combine the utility of being able to adjust and delete assignments with the ease of merely selecting some classes or uploading a syllabus. We believe that the lack of such a tool presents challenges for universities that can be solved with the solution we will propose. For instance, a lack of organizational skills among students leads to forgotten assignments, lower grades, and reduced preparedness [5].

The main issue that stems from the preceding points is that there is currently no tool available at SAU, or external tools, that provides a simple solution for creating a comprehensive database containing general information about every class, every assignment, and the specific assignments associated with each student.

Once this problem is solved, we are able to proceed by working on the following problems:

1) A way for event planners to visualize the workload of parts or the entire student body to better plan events.
2) A way for students to have an easy and automatic way of tracking their assignments and manage due dates without manually recording them.

B. Problem with project solution

With there being no template or structure that professors' assignment schedules must follow, there is no simple or easy way to automate the extraction of the schedule information. With the scope of this project being limited by time, it is not realistic to write an extraction method for every schedule variation, and variations are too large to account for in a script. A solution would require a dynamic way of extraction that can handle different formats.

III. Limitations and Delimitations

The following list highlights the *limitations* that this project is bound by.

1) Time: With the required timeline of this project being confined to four months, there is potential that specific features may be left out for lack of time. For example: Extraction fine tuning, a custom extraction tool (as opposed to using openAI API) or a completely polished student frontend.
2) Limited Source Data: Only schedules that have been provided by professors can be automatically populated into the database.

3) Unstructured Data: Since the system relies on syllabi that do not follow a unified template, any inaccuracies or inconsistencies with the structure can potentially affect the accuracy of data.
4) Manual Adjustment: There may still be situations where students need to manually input assignments or adjust dates if the syllabi are not structured in a way that can be easily parsed.
5) Team Size: With one backend developer and a front end/business team member, having a small team has limitations

The following list shares the *delimitations* that this project is guided by.

1) This project will only focus on SAU's classes and student body.
2) This project will be using openAI's API platform: https://platform.openai.com/docs/overview
3) This project will focus only on undergraduate courses and not include graduate or doctoral-level courses.
4) The tool will initially be designed for web-based access only, excluding mobile app integration or offline functionality.
5) The workload visualization tool for event planners will first focus only on academic workload (assignments, exams, etc.) and will not account for co-curricular/extracurricular.
6) With limited time, we will focus on the design and implementation of the backend, with the front-end being our secondary focus once the extraction process and data management portion are complete.

IV. Justification

We believe that SAU (and potentially other colleges/schools) could benefit both their administration and their students by having a database that stores information about classes, assignments and respective due dates, and students tied to these assignments. In order to prove this, we aim to create this database in a way that is easy to maintain without requiring a forced syllabus format for professors or any on-campus planners, and then creating two tools that will use this database. One will be a tool that will send students summaries of assignments that are coming up.

The other will be a tool that will visualize the workload of students and allow planners on campus to adjust event timing for busier days and increase attendance. In a 2017 study, Irena Ljubisic notes that engagement in college events is closely tied to a sense of belonging and personal/professional development [4]. By adapting events to student workload, planners can maximize attendance and belonging among the university students.

V. Objectives

We have set a list of key objectives to be met within this project, in order to reduce scope to a feasible level.

1) Automate extraction of assignments and due dates from syllabi.
2) Design and populate a database storing information about classes, assignments and respective due dates, and students with those assignments.
3) Create a tool to visualize the workload of college students.
4) Begin developing a tool that notifies students about upcoming assignments based on their preferred time.

VI. Theoretical Framework

Every semester brings its new challenges, many of which revolve around the process of student assignment management as well as student time and load management. Keeping this in mind, we are viewing the task of developing Assignment Hub through two frameworks that revolve around students, professors, and university planners. By explaining these frameworks below, we intend to clearly lay out and identify the key concepts and relationships that guide our approach.

From our perspective, there are approximately three options that students take in managing their homework and assignment schedules. For example, students with a strong appreciation for both timeliness and organization may pause to write down or type in each of their assignments, which is a very time consuming and error prone procedure. For those interested in timeliness, but not so much in organization, a more haphazard approach is often used. They will depend on prebuilt tools (usually lacking in either usability, functionality, or applicability) to make efforts towards completing their assignments on time. The issue in this process is often due to the lack of cohesion and abundance of confusion between the various tools used. The last group, as you may have assumed, do not make much effort to mark down assignment due dates, commonly relying on memory or professors for reminders in class or via email. Each of these have negative side effects such as using up precious time, forgetting or mixing up assignments, and eventually poorer grade averages.

At the heart of each of these approaches is often a conflict between value and cost. Students rarely desire poor grades, but they almost always have a lack of time and many other important tasks to complete. It is from this mindset that we consider Assignment Hub to be a tool perfectly positioned to assist students with these challenges. With the ability to create a calendar populated with assignments, using only a few clicks, students can have the best of all of the approaches listed above.

In addition to supporting students, Assignment Hub aims to address a second critical gap on campus: the absence of tools for university planners to assess student workload. We come at this aspect of our project with the understanding that any major planner on campus takes on their job with few tools to plan for the most impact and attendance. For smaller events, sending out a survey to attendees may suffice, but it is time consuming and tedious. Beyond this, there are few tools specific enough to undertake the planning. Planners may use other information provided, but there are no resources (to our knowledge) that provide insights into the workload of students throughout a given semester. This leaves most planners with the option to hope for the best.

This second framework makes it clear that a pre-built solution to assess student workload would be of great value to those involved in event planning and management. The planning aspect of Assignment Hub forms an avenue for event organizers to visualize the workload of students, from an academic perspective, and to filter that data to their desired student subgroups. Our goal would be to increase attendance to vital university events by providing this tool to university planners.

VII. State of the Art

Within this section, we will discuss the current and most recent research, practices, and solutions within the area of automated assignment data extraction from course syllabi and student workload data visualization and utilization. We will start with existing materials that discuss the use of OpenAI's API (and other LLMs) to extract data from PDFs. For instance, Kumar et al. explored the potential of OpenAI's API in efficiently querying and extracting information from PDFs through chunking and vectorization methods for Q&A applications (. We will then discuss various research that has been done on using these tools to extract data specifically from course syllabi. Next, we will explore what tools exist to help students with their workload and assignment management, including any comparisons that we can draw to Assignment Hub. We will also review what tools are present to help university planners organize events around student workload. Finally, we will present the limitations of existing material and how Assignment Hub will contribute above what is already available.

From our research, we failed to identify any existing platform, service, or tool that focused on extracting data from a syllabus specifically. There are, however, tools which specialize in PDF data extraction. We examined an assortment of these tools such as PDF Chat and ChatPDF but their primary use is within their user interface. They are not geared toward API integration and automation. We also struggled to get a uniform response that could be used to transform the extracted data into a form we could use, such as a JSON format. We also looked at Azure AI Document Intelligence, which is a tool built by Microsoft for extracting structure and unstructured information from PDFs. Our main concerns with Azure's tool were the cost being too high for our initial testing and that it would struggle with the format of our data source. The variation between each syllabi format is substantial and tools that are marketed to pull data from receipts, invoices, and other similar forms (generally following a similar structure) would likely fail to extract properly. Irugalbandara et al. conducted a comparative analysis showing that open-source SLMs can extract PDF data with effectiveness comparable to OpenAI's GPT-4; however, Assignment Hub builds on OpenAI's proven reliability to handle unstructured syllabi content, where tools like SLMs have not yet been tested extensively for parsing highly variable formats [1]. For our purposes, OpenAI's API system is robust and adaptable, having been shown to have strength in natural language processing and its ability to analyze unstructured text accurately (C. M. Gallardo). The Smart PDF Inquiry Hub project provides a strong example of how OpenAI is capable to query and extract relevant data from PDF documents using similarity search and advanced embeddings (U. Kumar).

While there are numerous tools for extracting PDF data on the market, we could not find any that were designed solely for assignment calendar extraction. There are various examples of research done with the goal of analyzing syllabi using semantic analysis; however, their aim was to compare course content rather than retrieve a structured response [3][7]. In regards to extracting course content, our process and aim is unique.

Taking a comparative and comprehensive view of our project, it is clear from our research that there are no current market solutions whose processes and goals match those of Assignment Hub's. There are countless tools for managing student assignments but none are designed to be automated and streamlined straight from the syllabus. These tools require the user to input each assignment manually, which can be a considerable time draw. Additionally, we could not find a tool that allowed the user to monitor and analyze student workload from data extracted from course syllabi. Of all the tools that exist to help planners within a university, none utilized student assignment and workload data to examine preferable event times. In searching for syllabus workload visualization tools, the only result was for a tool that examined the differences between syllabi [8].

VIII. Software Design and Implementation/Experiments

For this section, we will discuss the overall design of the Assignment Hub tool and all of the components that are being utilized to create a unified ecosystem to perform syllabus extraction, assignment management, and workload visualization. After we discuss our design and tech stack, we will explain our system in its current state and the process/tools that are being implemented to achieve our final objectives.

A. System Model

For the sake of clarity, we have split the sequential diagram into two parts: the assignment management portion and the workload visualization portion. It is important to note that the process, until the data is stored in the Azure SQL Server, is virtually the same for both tools. These diagrams explain the various tools and users within the system and how they will relate to each other. *Figure 1* is covering the assignment management side of Assignment Hub. *Figure 2* covers the various components of the workload visualization side.

As you can see from these diagrams, the process starts with a student or administrator uploading syllabus to the Management Interface. This is the same interface that students will use to access and manage their assignments. Once the assignment is uploaded, it will be sent to a custom Python server utilizing Flask and hosted with Azure to allow the OpenAI API to have access to the syllabus for data extraction. The API will examine the syllabi and return the pertinent data in a JSON format. This will be returned back to the Flask server where it is sent to be stored in the SQL Server being

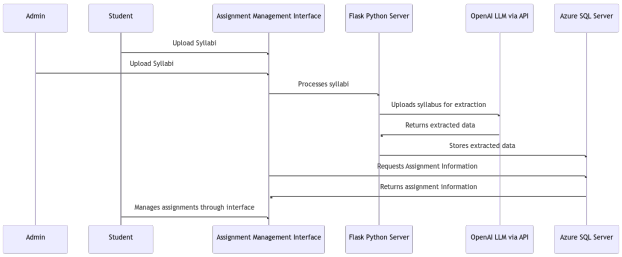

Figure 1. Student Assignment Management Sequential Diagram

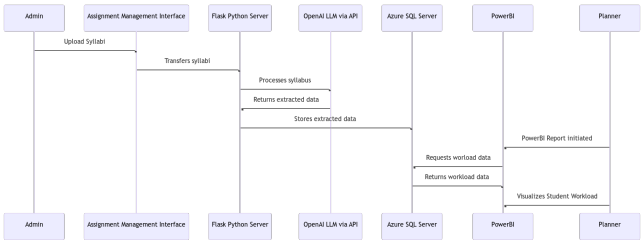

Figure 2. Student Workload Visualization Tool Sequential Diagram

hosted in Azure. With the Assignment Management Interface developed in React, the Flask server acts as a necessary safety measure to avoid hard coding queries where they would be accessible to the end user. Once this data is stored in the Azure database, it is accessible for both the assignment management system and the workload visualization. We will discuss what data is extracted when we discuss the design of our database.

For the assignment interface, the student will log in and be able to visualize and manage the assignments corresponding to the courses they are in. For the workload visualization, we will have a pre-built PowerBI report that allows planners to filter the semester workload by department and level. When there is a change in the syllabus data, a refresh will take place and the data will be updated by requesting it from the SQL Server via an API already built into PowerBI.

To further improve clarity, *Figure 3* shows a use-case diagram that explains the various actors within the system and their corresponding roles.

In order to access the necessary data for all Assignment Hub's functionality, we thoughtfully prepared the design for the database tables. The structure had to be planned ahead to provide the required class, student, professor, assignment, and enrollment data. Below in *Figure 4* we have the database schema that we used to build our SQL Server database.

B. Current Project State

University admins will have access to and be able to manage the folder that stores all of the syllabi and the csv files that store the student enrollment. The required data is processed through the Python script which utilizes the OpenAI API to extract and validate the data, populating the database with the processed output. This validation process has seen accuracy ranging from 50-100%, with the average being on the upper portion of that scale. This data is being stored in our Azure SQL Server. A student or other user can log in and input their

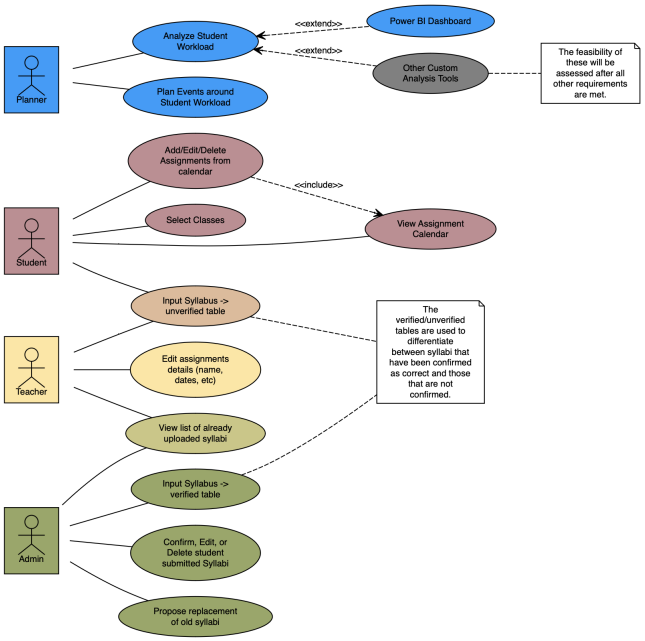

Figure 3. Project Use Case Diagram

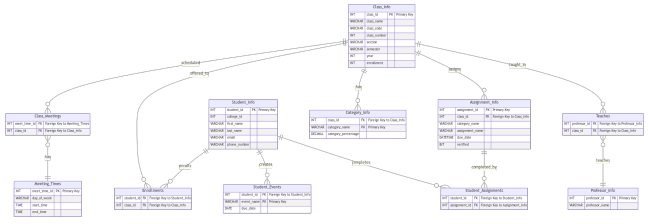

Figure 4. SQL Server Database Schema

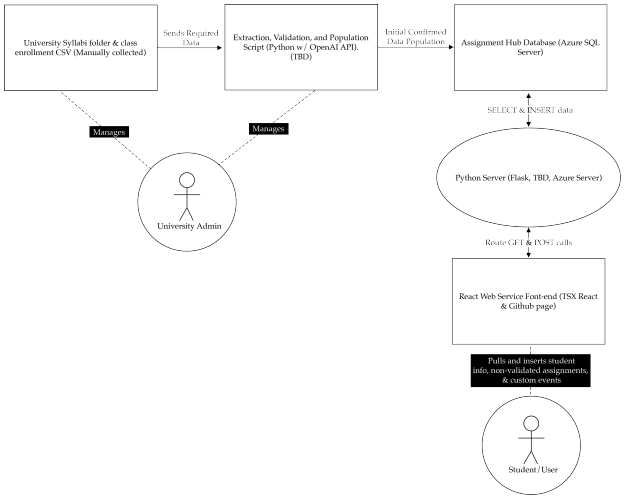

Figure 5. Process Diagram

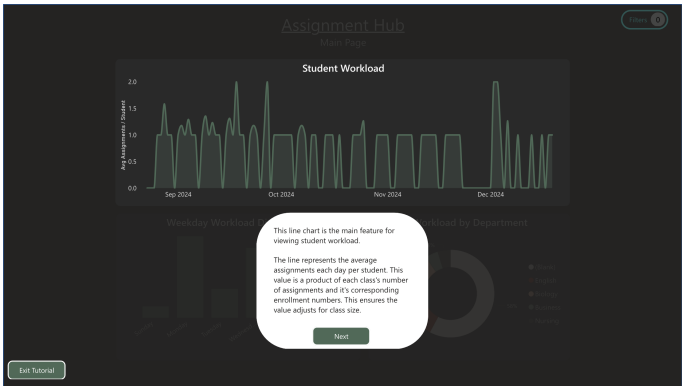

Figure 6. Example Power BI Report Tutorial Page

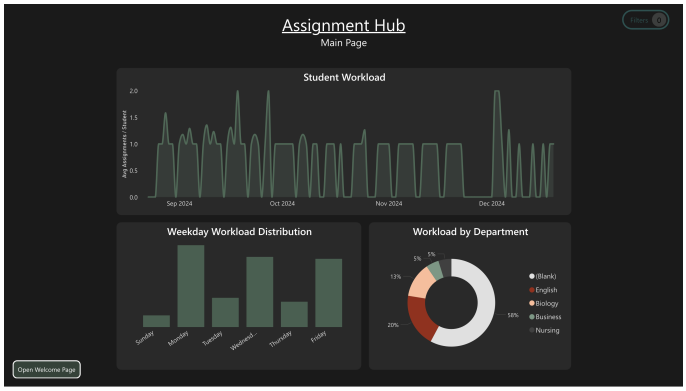

Figure 7. Main Power BI Page

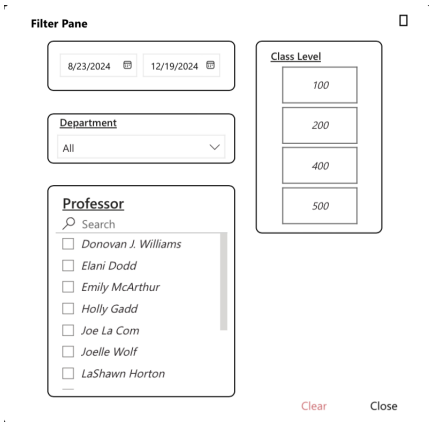

Figure 8. Filter Pane

student information, non-validated assignments, and custom events. The front-end is built with React and hosted through Github pages. The Python Server (running with Flask, TBD, and Azure Server) acts as the middleman to upload the student information to the SQL Server and to retrieve validated assignment data to display on the student's front-end.

We achieved our goal with the front-end for both the React assignment interface and the Power BI dashboard, allowing the end users to access the data that has been processed from the syllabi. The React interface for students has the basic functionality to allow users to create or log in with a google account, enroll in specific courses, view enrolled courses, see existing assignment due dates for those courses, and create new events.

With a small test set of syllabi uploaded to SQL Server for a start, we have been able to make a connection to the database from Power BI using the built-in API and create a report dashboard to visualize the student workload. The dashboard has a simple home page that allows for navigation to the main planning page or to enter the tutorial.

When user's select the tutorial, they are taken through a step-by-step guide that explains the elements on the main dashboard. Figure 6 is an example of one of the eight tutorial steps.

The main page on the report has three main components that allow users to visualize the aspects of student workload. The main component is a dynamic line chart that reflects the average number of assignments per student, based on the filtered time frame and classes. In order to calculate the average number of assignments per student on a specific day, we used a DAX Measure within Power BI to get the classes on that day and multiply the number of enrolled students in each class with the number of assignments due in that class. The measure then takes the total number of enrolled students within those classes and divides the original value by the total. This gives the user a rough estimate of the workload a student might have on that specific day. This information can then be used to plan to hold events on days that have a lighter workload.

Within this report, users can also filter by various values (shown in the figure below). When one filter is selected, all other filters adjust accordingly. A user can select a specific time period to view, departments, professors, and class level (100 to 500).

We also added two other visuals into the report that show a distribution of the assignments by department and the average workload on each of the week days. Within the example, you can see the heavier assignment days within our current data are Monday, Wednesday, and Friday. All of these dashboard elements can be used by planners to ensure that they are selecting the best day in order to maximize attendance.

C. Future Improvements and Additions

The current state of the project is right where we had aimed to be, but there is always room for further implementation and improvements. If we chose to continue pursuing this project further, we would have more work to do in improving the front-end and increase the testing, especially gaining outside insight. Additionally, we would incorporate the Power BI report as an embedded element into a web platform to combine the planning and student management aspects of our project into one. Overall, we believe that this project has potential to be refined and then utilized by universities across the country.

IX. Conclusion

Assignment Hub's proposed aim is to provide a practical solution to simplify assignment management and foster data informed planning. By automating the process from syllabus

to interface, our tool addresses common challenges such as the unstructured nature of syllabus schedules and the time consumption to retrieve this data at scale. Assignment Hub is designed to assist students with organizing their assignments and to allow event planners the ability to schedule events that take student workload into account.

Looking forward from here, Assignment Hub has great potential to become a versatile resource for universities and other educational institutions. With more time in refining its functionalities and adding more capability, we believe that Assignment Hub could be implemented in co-curricular planning and mobile integration, further increasing the value. With the system being designed with scalability in mind, there is no limit to the impact that Assignment Hub can make in this realm.

REFERENCES

[1] C. Irugalbandara et al., "Scaling Down to Scale Up: A Cost-Benefit Analysis of Replacing OpenAI's LLM with Open Source SLMs in Production," 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Indianapolis, IN, USA, 2024, pp. 280-291, doi: 10.1109/ISPASS61541.2024.00034.

[2] C. M. Gallardo Paredes, C. Machuca, and Y. M. Semblantes Claudio, "ChatGPT API: Brief overview and integration in Software Development", Int. Jour. Eng. Ins., vol. 1, no. 1, pp. 25–29, Nov. 2023.

[3] G. Orellana, M. Orellana, V. Saquicela, F. Baculima and N. Piedra, "A Text Mining Methodology to Discover Syllabi Similarities among Higher Education Institutions," 2018 International Conference on Information Systems and Computer Science (INCISCOS), Quito, Ecuador, 2018, pp. 261-268, doi: 10.1109/INCISCOS.2018.00045.

[4] I. Ljubisic, Student Engagement in College-Related Events. M.S. thesis, Dept. of Service Systems, Coll. of Applied Science and Technology, Rochester Inst. of Technology, Zagreb, Croatia, 2017. [Online]. Available: https://repository.rit.edu/theses

[5] J. M. Gambill, L. A. Moss, and C. D. Vescogni, The Impact of Study Skills and Organizational Methods on Student Achievement. Action Research Project, Saint Xavier University, Chicago, IL, USA, May 2008.

[6] U. Kumar, S. R. S, K. P and G. Sivakamasundari, "Smart PDF Inquiry Hub: A Comprehensive Solution for Efficient PDF Document Querying and Information Extraction," 2024 International Conference on Expert Clouds and Applications (ICOECA), Bengaluru, India, 2024, pp. 192-198, doi: 10.1109/ICOECA62351.2024.00045.

[7] T. Sekiya, Y. Matsuda and K. Yamaguchi, "Development of a curriculum analysis tool," 2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET), Cappadocia, Turkey, 2010, pp. 413-418, doi: 10.1109/ITHET.2010.5480101.

[8] Y. Yaginuma, "Syllabus visualization tool based on standard curriculum," 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 2017, pp. 1-2, doi: 10.1109/GCCE.2017.8229215.