

The Academic Engine – The Heart of Student Management

Karen Chance, Business Applications Analyst, Arion Support Division, Auckland University of Technology

Dr Ineke Kranenburg, Academic Director, Auckland University of Technology

Arion is the established Student Management System for the Auckland University of Technology. The system has been re-aligned to put the new Academic Engine at its centre. The Academic Engine enables greater flexibility from a system perspective but also more comprehensive information on the student journey for both staff and students.

The Academic Engine components build upon each other. Paper lists define the papers for requirements. Completion Requirements use the Paper lists. The Courses of Study bind these requirements together to form templates that generate the individual Study Plan for students. The Study Plan is based on the components of the Academic Engine that allows the student to more easily combine majors from the programme in which they were admitted with other majors or minors external to the Programme.

This additional flexibility has the potential of adding disruption. Student choice is not so tightly constrained and this potentiality has an impact on the viability and availability of offerings.

Both staff and students are able to discern more clearly what options are available. Study Plan Modelling not only helps staff give flexible and accurate advice, but students are able to understand options. Reporting and communicating with cohorts of students is enhanced.

Introduction

Arion is the established Student Management System (SMS) for the Auckland University of Technology (AUT). It is a purpose-built system that has been used for more than 16 years. During that time, the University has also undergone significant changes in size and diversity of both student population and qualifications. Increasingly, qualifications have become more complex as students are able to complete a wider range of courses of study. These include joint degrees, areas of specialisation (majors and minors) from outside of their principal programme as well as elective courses from outside of their discipline or field of study. This complexity has led to increased demands on the SMS, especially in terms of study planning, reporting and determining progress from enrolment through to graduation.

In response to these issues, AUT has completed a multi-year project to realign Arion around the Academic Engine. This Academic Engine maintains academic regulations as components and uses these components to build Courses of Study that are used as templates to create individual Student Study Plans.

Historically, Arion was developed before AUT became a university. Auckland Institute of Technology (AIT), AUT's predecessor, made enrolment Arion's primary process (Figure 1). Essentially Arion was built to make enrolment as easy and as efficient as possible. All the component processes of the student management system - Admission, Fees and Finance, Assessment, Progress-to-Completion – were self-contained tasks linked to the student's application and paper enrolments, rather than the qualification outcome.

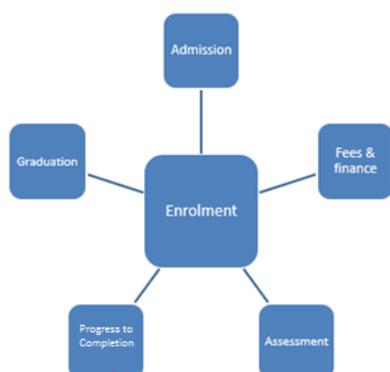


Figure 1 Previous State

AUT now focusses on the student's progress in the course of study in relation to the requirements of the prescribed path within the qualification. This is a qualitative change from emphasising numbers (of students) to a focus on results for the student, further driven by the introduction across the industry of Education Performance Indicators (EPI). Staff members need to consider how to help the students to the right outcomes. At the same time, it is important to know which outcomes the students are choosing, to understand the choices a cohort are making to better service it throughout the student journey.

Most of this redevelopment has concentrated on re-aligning functionality already available within Arion rather than changing its underlying architecture. The ultimate result is that the student is empowered to know where they stand as far as their current desired outcome, as well as knowing what other options are available and how to get there from where they currently are.

Specific Problems with previous Arion

Historically, Arion has struggled with many problems.

Structure inadequate: The structure for enrolment captures both enrolment and student participation. One implication of this focus is that every year the system must emulate returning students creating a "magic" application. Tasks in Enrolment, Assessment, Fees and Finance are tasks to be completed and then linked to this application.

A second problem was that changing student outcomes was an onerous process. When a student wants to change a major, for example, a member of staff must unenrol the student from all previously enrolled classes related to the original major and then re-enrol them in new classes for the new major, generating invoices for classes that the student had already paid. These invoices must be suppressed.

Difficulty tracking academic regulations for each student: The regulations that determine the student's path to completion are those applying on the day the student first enrolled. The staff must determine which regulations apply for each student. This involves comparing the student enrolment information with external regulations.

Student choice unnecessarily constrained: As mentioned above, changing one's outcome is a time-consuming process. This constrains unnecessarily the choices a student may have at a time in which there are more choices than ever before. Further, the old SMS did not fully or easily present the entirety of the student's course of study options.

Difficulty in discerning progress to completion for both the individual student and the programme as a whole: There is a need to enable enhanced management of student progress through statuses on the study plan at both the individual student level but also cohort groups. This enables members of staff to offer better and more timely course advice. In addition, they need to be able to alert students (where necessary) to contact them to discuss their study options.

Students are also able to track their progress visually and number of credits left to complete in relation to the requirements of their study plan.

The Solution

There was a desire to build a robust system for a 21st century university. The internal Arion Solution Division (ASD) was responsible for effecting this change. AUT contracted a third party to begin the requirement elicitation process. This stage began in 2012, consisting of a preliminary nine-month process. ASD recognized the importance of an iterative and incremental requirements gathering process (Larman and Basili, 2003) and built a feedback loop within the process. ASD also recognised the importance of both the end user and the customer (Agile Alliance, 2001). During this initial process the requirements elicitation team interviewed a wide variety of end users and customers such as Academics, Programme Administrators, and members of the Academic Quality Office (AQO). ASD collated, analysed, summarised the results of these interviews. Through a series of feedback sessions, the team presented the results back to the interviewees to elicit more information.

Based on these early requirements, AUT developed an ambitious solution:

- A new, enhanced way of entering academic regulations.
- Individualised "Study Plans" for each student

This new way of entering academic regulations is the Academic Engine. The Academic Engine builds the components that it turn create Student Study Plans. These Study Plans are the prescribed path by which a student can achieve their desired outcome. While the Academic Engine effectively outputs the Study Plan based upon the Course of Study chosen, there is also a need to allow for personalisation of the study plan itself, in order to cope with whatever individual circumstances a student may have.

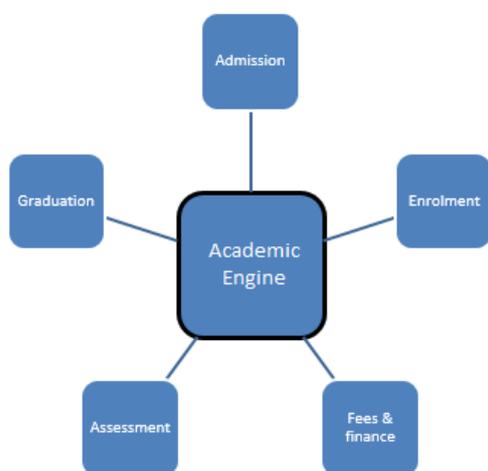


Figure 2 Current State

The current state of the Arion SMS revolves around the Academic Engine instead of Enrolment. This change simplifies many problems. The Study Plans produced by the Academic Engine enhance a focus on student outcomes. No longer are a series of Enrolment, Assessment, Fees and Finance tasks seen as ends in themselves but as supporting the Study Plan and student outcomes.

Along with the functional requirements, those consulted suggested many non-functional requirements, according to which the solution should

- Enhance reuse of components to promote both efficiency of entering information and ease of finding information
- Allow the information from the Academic Engine to be available for other uses
- Be independent of roles or specific organisational configurations for these efficiency gains
- Have appropriate inbuilt quality assurance checks
- Provide improved and more accurate reporting on major subjects as well as student progress in general
- Provide enhanced measuring/monitoring of student progress
- Provide an improved method for communicating with specific groups of students
- Support student expectations throughout all student views
- Provide a platform for outlining student options, improved consistency of course information with regards to course planning for students, administrative and academic staff
- Enable the impact of a change to papers or completion requirements to be modelled prior to implementation.
- Provide an effective and easy to use user interface aligned to what currently works in academic advising.

After development of the new Academic Engine began, a user reference group (including many users from the faculties along with members of the Academic Quality Office) has provided ongoing guidance to ASD.

Individualised “Study Plans” for each student

A Study Plan is the path by which a student can achieve his or her outcome. Study plans look much like the structure delineated in the university calendar, the official publication for all the qualifications that includes the requirements for completing the different prescribed paths or courses of study available within the qualification, as in Figure 3, which is an excerpt from the 2016 AUT Calendar (Auckland University of Technology, 2016, Page 380).

A programme has a variety of courses of study, based on its majors and minors. The student applies to enter a Course of Study in a Programme. AUT may offer a place in a Course of Study for a programme. The student accepts this offer. At that point, the system can prepopulate the students study plan (prescribed path) or alternatively the student can select majors, minors, papers and electives (in essence the components available within the Course of Study). Doing so populates the Student's Study Plan.

The Course of Study consists of components based on the academic calendar. In the example in Figure 3 for the Bachelor of Communication Studies, the Core is the requirement that every student in a Programme must satisfy, regardless of Course of Study. The student may combine a major, a major and a minor, or possibly a double major to the Core. Any shortfall in points can be made of other papers in the Communication Studies table or papers from other programmes.

Structure

Core Papers

90 points	Level 5:	145701, 145708, 145717, 145718, PHIL580
and 30 points	Level 6:	146719, COMM601
and 15 points	Level 7:	147717

Majors

Advertising Creativity

30 points	Level 6:	146735, 146740
and 45 points	Level 7:	147764, 147772, 147835
and 15 points from	Level 7:	147774, 147834, ADVT780

Creative Industries

30 points	Level 6:	146756, CIND600
and 45 points	Level 7:	147780, 147819, 147820
and 15 points from	Level 7:	147779 or an alternative Creative Industries paper with the approval of the programme leader

Digital Media

15 points	Level 6:	146735
and 15 points	Level 7:	147800
and 15 points from	Level 6:	146747, 146750
and 45 points from	Level 7:	147724, 147750, 147767, 147768, 147775, 147812, 147821-147823

Journalism

15 points	Level 5:	145727
and 30 points	Level 6:	JOUR600, 601
and 60 points	Level 7:	147830, 147831, JOUR700, 701

Figure 3 Core and Specialisation Requirements from 2016 Calendar (Auckland University of Technology, 2016, Page 380)

In Figure 3, the Core has three conditions, which the students must also satisfy as part of the Core requirement. In addition the student typically chooses a major and satisfies any points lacking with Electives.

As can be seen from the physical Calendar depicted above, a Bachelor of Communication Studies involves a "Core" that all students take, and one of many specialties. These majors are Advertising Creativity, Creative Industries, Digital Media, and Journalism. What is not shown in Figure 3 are the Electives. These may be papers listed in the Bachelor of Communication Studies table, or "any other bachelor's degree with the approval of the programme leader." (Auckland University of Technology, 2016, p. 379).

Core, Major, and Electives correspond to different kinds of completion requirements. As can be seen above, the completion requirements are broken down into conditions, in which a certain number of points to complete is set before a condition is satisfied.

In the above example, for the Core, each student must complete every paper for every condition to satisfy the core requirement. However, alternative papers may be listed on conditions.

The Study Plan records the student choices. The Study Plan captures the academic regulations as captured in the Course of Study current as of the moment the student first enrolls. Over time, the student can have her Study Plan personalised.

The student can also change majors and minors without requiring the overhead currently required.

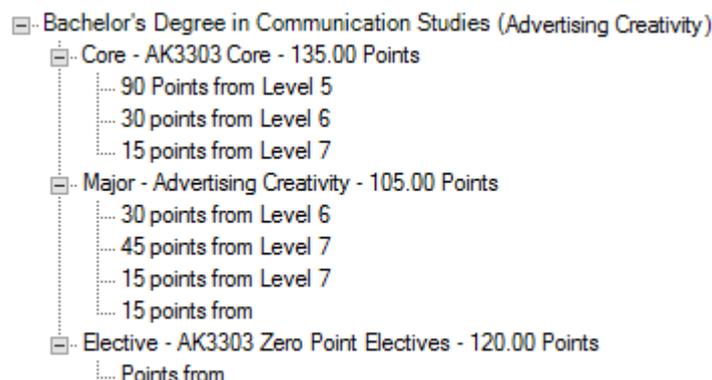


Figure 4 Course of Study Structure of a Study Plan in Arion

Figure 4 above displays the structure of a Study Plan from the development version of Arion SMS. Comparing it to the excerpt in Figure 3, one can see that the Study Plan structure is essentially the same as that in the Calendar. The Study Plan has a Course of Study structure that combines completion requirements to provide a prescribed path by which the student will achieve the desired outcome, in this case a Bachelor's Degree in Communication Studies (majoring in Advertising Creativity). The completion requirements are further broken into conditions. The conditions may focus on a level or may be multi-level.

The user can drill down and see what the papers are on each completion requirement and the conditions on the requirements, as shown below for the Advertising Creativity major.

Component	Condition	Min Pts	Max Pts	Paper	Level	Points	Status	Personalised	Allocated	Mandatory	Exempted	Waive Req	Fulfilled
Major	30 points from Level 6	30.00	45.00	146735	6	15.00	Available	<input type="checkbox"/>					
Major	30 points from Level 6	30.00	45.00	146740	6	15.00	Available	<input type="checkbox"/>					
Major	30 points from Level 6	30.00	45.00	146752	6	15.00	Available	<input type="checkbox"/>					
Major	45 points from Level 7	45.00	45.00	147772	7	15.00	Available	<input type="checkbox"/>					
Major	45 points from Level 7	45.00	45.00	147835	7	15.00	Available	<input type="checkbox"/>					
Major	45 points from Level 7	45.00	45.00	147764	7	15.00	Available	<input type="checkbox"/>					
Major	15 points from Level 7	15.00	15.00	147774	7	15.00	Available	<input type="checkbox"/>					
Major	15 points from Level 7	15.00	15.00	147834	7	15.00	Available	<input type="checkbox"/>					
Major	15 points from	15.00	15.00	405701	5	15.00	Available	<input type="checkbox"/>					

Figure 5 Papers for the Advertising Creativity Major in Arion

Unlike the Calendar, Arion's Academic Engine specifies the Electives suitable for the Course of study from which the student can choose.

The Academic Engine:

The solution re-aligns Arion to put the new Academic Engine at the centre of the system. The Academic Engine enables greater flexibility from a system administrative perspective but also, more comprehensive information on the student journey. The challenge is to automate the academic regulations held in the yearly calendar into an Academic Engine.

The Academic Engine consists of components

- Paper Lists
- Component Lists
- Completion Requirements
- Courses of Study

The Academic Engine components build upon each other. Paper lists define the papers for a specialisation, and allow for easy updating. Completion Requirements “consume” these paper lists. Instead of adding individual papers to a completion requirement condition, we “add by list.” Completion requirements are the central component of the Academic Engine. They provide flexibility amongst the various specialisations such as majors and minors. The staff member combines these requirements as delineated in the Calendar to form Courses of Study. A Course of Study forms a template that generates the individual Study Plan for students.

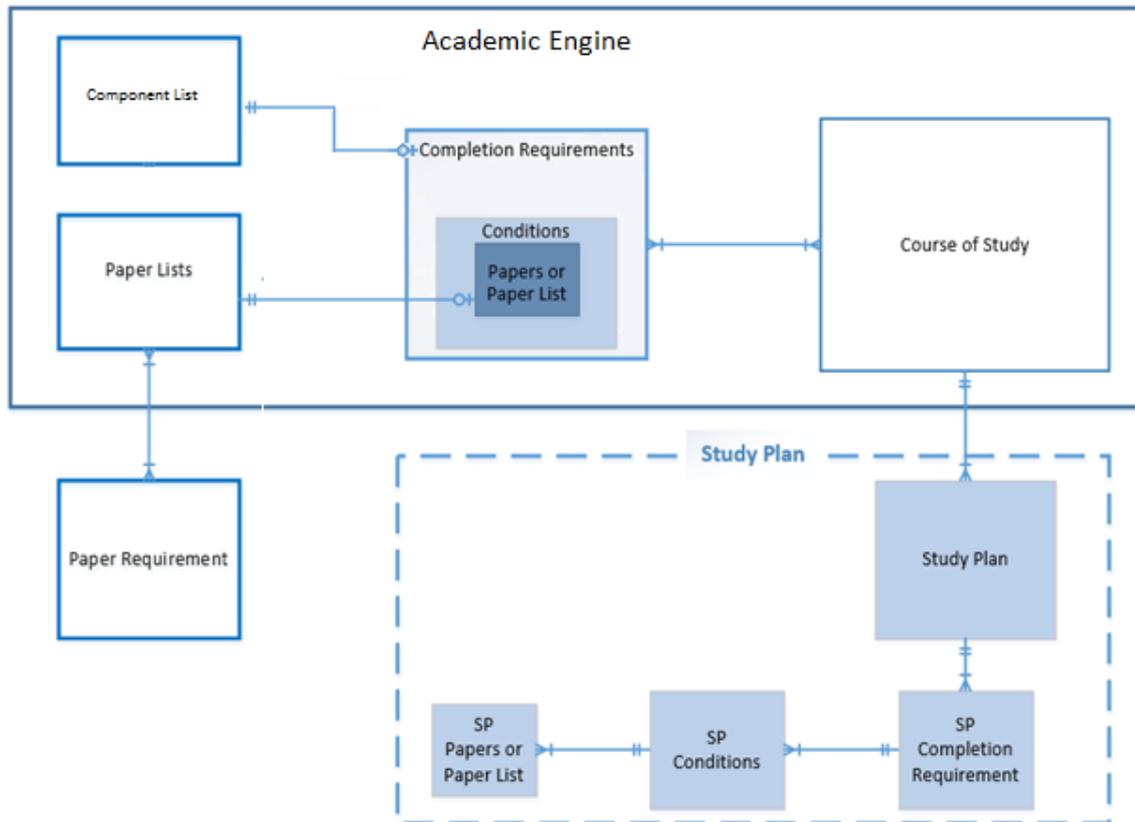


Figure 6 The Components of the Academic Engine and their relationship to Study Plans

Paper Lists

In order to ease the entry of information into the system, and to provide the required quality assurance as required by the high-level constraints, the base component of the academic engine is the paper list.

Paper Lists define the papers used in different completion requirements (core, majors, minors, etc.). If we have a major in Electronics, we also have an “Electronics” paper list.

There are three types of paper lists

- Master List
- Other Paper Lists including Core and Specialisations
- Elective Lists

The types of paper lists correspond with the different categories of completion requirements. The Other Paper Lists is either the Core paper list or the “specialisations” such as the various majors and minors.

The Elective lists contain the elective papers.

The Master List however is composed of all the papers of the Core and all the Specialisations. The Master List has two functions

- It acts to help validate and keep consistent the paper lists

- If a paper is not on the Master List, a user cannot add it to an Other Paper List.
- It helps drive change through the system

Changes such as deleting or substituting papers on the Master List drive that change through any Other Paper List that has those papers. It also drives changes through to the Completion Requirements.

Ease of Use: Paper lists make creating Completion Requirements and even other paper lists easier. The system allows an “Add from List” functionality that keeps the user from adding individual papers to paper lists. The Completion Requirement is linked to a Paper List (or possibly a Component List in the case of a Modular Completion Requirement) that

- Constrains and Validates: The Master List constrains what papers can be placed on Other Paper Lists.
- Is Reusable: Paper Lists can be used to make other paper lists in other programmes.
- Automatically Update: The paper list identifies those other paper lists that should be updated.

Component Lists

One concern early in the development process was over how much time will be necessary to create and maintain Courses of Study for every possible combination of specialty. At this point, we developed component lists to allow the user to choose from a list of components, or completion requirements. A modular completion requirement is linked to a component list. A Configurable Course of Study in turn consumes it. This Configurable Course of Study allows the staff user to create a study plan from a combination of Completion Requirements from component lists.

Ease maintenance of Courses of Study: Component Lists allow the staff user to develop different combinations of Courses of Study as needed, without having to prepare many different permutations of major/minor Completion Requirements.

Completion Requirements

In parallel to paper lists, there are completion requirements for Core, Specialisation, and Electives.

Completion Requirements are the specific components of a Course of Study.

One requirement for Completion Requirements is that they must be flexible. There are “unspecified” completion requirements, allowing students to explore within a programme without committing to a major. Users can designate a completion requirement as “non-academic,” involving a requirement that does not equate to classroom time. Lastly, there are modular completion requirements that hold a component list. This kind of Completion Requirement allows the user to choose a major/minor combination when offering a place. This makes the need to create and maintain large numbers of Courses of Study for many different combinations obsolete.

Courses of Study

Courses of Study are the templates for Study Plans. Many students may have study plans created from the same template. At the same time, a student may have a Study Plan based on a unique Course of Study.

Courses of Study may be components or prerequisites of other Courses of Study. There are also Configurable Courses of Study that allow a member of staff to assemble a Course of Study at Offer. This is especially valuable for those programmes that have many possible permutations of specialisations.

The Course of Study functions as a template by which to build Study Plans for students. The version of Course of Study is that current on the start of teaching date when the student first enrolls.

The staff user can build different Courses of Study using different components. For a single Programme, we may assemble different Courses of Study that reflect different prescribed paths.

Easy to Construct: The user can assemble (within rules) Courses of Study out of Completion Requirements, Courses of Study as Components and Courses of Study as Requirements.

Configurable Course of Study: The user can build a Course of Study based on a Structure such as Single Major, Double Major, Major and Minor. Each of these components requires a modular completion requirement. During the admission process, the user can choose the desired specialisation(s) for the offered applicant.

Maintaining the Components - Versioning, Impacts and Automatic Updating

As academic regulations change, the Academic Components must also change. This is where the real power of the Academic Engine resides. The Academic Engine drives changes “forward” from each academic component to the next component that consumes it.

Versioning of the Academic Components enables this kind of change. The versioning of each of the Academic Components allows the university to change academic regulations

When a staff member changes a component, it is to a new version of that component. Each new version must be both verified and published. Verification is a two-step process – it validates the component and then does an impact analysis to see what the next downstream components are that will be impacted by this change.

Between verification and publication, the staff member can examine the components that will be changed. If the user wants the changes to cascade forward, the user can indicate which if any of the impacts they “accept.” If none are accepted, the academic component becomes available.

Publication both makes the component available for use and updates the next downstream components if the user chooses it to be updated. Changes cascade forward. Changes to Paper Lists cascade forward and automatically accepted changes to Completion Requirements (and Other Paper Lists in the case of the Master List). Changes to Completion Requirements cascade forward to update accepted Courses of Study. And lastly, changes to Courses of Study may also push forward changes to the Study plans. This step is tightly constrained with rules to make sure these changes do not disadvantage the student.

Personalising the Study Plan for the Individual Student: Each student’s study plan can be personalised. The kinds of personalisation available allow the student to complete their study plans in slightly different ways. Some of these personalisations make sense from the viewpoint of the member of staff. For example, making a particular paper mandatory is a way to redress a deficiency in the student’s background or skills. However, the student may request many of these personalisations.

Personalisations include

- Allocate/Unallocate
- Allow a Paper
- Make a Paper Mandatory
- Waive Requirements for a Paper
- Allocate Credit
- Vary Points
- Exempt

Allocate/Unallocate: Passing a paper updates the Study Plan. As a paper may exist on different requirements, the Academic Engine “allocates” the points completed to the condition with the highest priority (effectively, the occurrence of the paper “highest” on the Course of Study tree (Figure 4)). However, it is possible that the student wants to take another paper on that requirement and would like to apply that completed points to a different, lower-priority requirement. The member of staff uses Study Plan Personalisation to accomplish this change.

Allow a Paper: A member of staff can add any paper to a student’s Study Plan.

Make a Paper Mandatory – If there are a number of options by which a student can satisfy a condition, but a member of staff wants a student to take a specific paper, the staff member

Waive Requirements for a Paper: A student may have already demonstrated proficiency and does not need to take a prerequisite. In this case, a member of staff waives the requirements for that paper, specifically prerequisites, corequisites, and pre- corequisites.

Allocate Credit: Within rules, AUT grants Recognition for Prior Learning (RPL). This credit may be either internal or external, specified or unspecified. In the case in which the student has received a tranche of unspecified credit, a staff member allocates the credit to a specific condition, and it counts towards the satisfaction of that condition.

Vary Points and Exempt: Vary Points and Exempt are of most interest to the staff member. These personalisations are rarely used. Vary points allows the higher-access user to change downward the points of a completion requirement. Exempt allows the user to exempt the student from the need to satisfy a completion requirement. These personalisations are most often used for Advanced Standing students who enter a programme with a body of work behind them, or in cases of Bereavement.

Changing One's Course of Study

Study Plan Management allows the user to change the underlying Course of Study on a Student's Study Plan. The staff user chooses a Study Plan to change and can choose the kind of Course of Study structure within the Programme – Single Major, Double Major, Major and Minor. Having chosen a structure, the user can choose from the relevant options.

Upon choosing alternatives, the function creates a new study plan. Any personalisations on the old study plan are lost. However, it updates the status of already completed papers on the new Study Plan with the status on the old, along with automatic allocation of these completed papers. The staff user can also allow any papers that were completed on the old Study Plan but do not exist on the new.

Modelling Study Plan Scenarios

One of the advantages of using study plans is to allow members of staff and eventually the students themselves to model a proposed change. Study Plan Modelling allows different scenarios based on different qualification outcomes.

On entry, Study Plan Modelling displays a student study plan. This display shows the current state of the student study plan. The user can change the Course of Study to model the impact of changing the Course of Study would have. The function models a new Study Plan based on the chosen Course of Study. This display includes progress-to-completion and any unusable completed papers in the modelled Study Plan.

The value is that the student can literally see how far along they are towards their current qualification. It also allows what-if modelling to see how the student's current progress could be applied to another, different qualification. In this way, Study Plans empower the student to choose whatever outcomes they desire and know the implications of such a change.

Discussion

Benefits

The Study Plan is based on a chosen prescribed path within the academic regulations (requirements) of a qualification. These regulations are in turn, reflected in the components of the Academic Engine. This allows a student at any point of enrolment to see the various prescribed paths (joint degree, major, minors or combinations of these) available to them than is currently the case. In particular, it allows a student to more easily combine a major from the programme in which they were admitted with other majors or minors external to the Programme as well as elective papers across other disciplines that are available to all students.

The staff are able to discern more clearly what the student options are and are able to offer flexible and accurate advice. It helps staff give student advice with modelling possibilities. Reporting and communicating with cohorts of students will be significantly enhanced. This also enables targeted communication to example students who have chosen a major to inform them of a special event or proposed changes in course offerings.

Reporting on students enrolled and completing majors has been difficult in qualifications that do not require students to specify their chosen major on enrolment in the broader qualification. This type of information is often needed for both internal planning purposes and for external accreditation requirements.

While these benefits for staff alone are invaluable, it is in regard to the student where the biggest benefit lies. The Study Plan enables the student to have a 360 degree view of all the possibilities available to them. The student has not only a clear map to their current qualification outcome, but a map to every outcome available at AUT. With the Academic Engine, the student is empowered with knowledge of not only where they are but also how to get to where they want to go.

Disruption

This additional flexibility has the potential of adding disruption. Student choice is not so tightly constrained and this potentiality has an impact on the viability and availability of offerings. Staff within a School often pre enrol students into their course of study ensuring that students are enrolled in the appropriate papers and to ensure that from a resourcing perspective the class is viable or that there are sufficient offerings of the course to accommodate both staff and students. Students are often slow to enrol and a greater level of choice and flexibility might result in more bottlenecks or requests for more assistance thus slowing the enrolment processes down.

Challenges

Implementing such a comprehensive change to a working system. As Arion is a successful SMS that is currently supporting the university, it is imperative that the Academic Engine is implemented without disrupting these services. To accomplish that end, the project has a multi-year rollout in which Arion is tested and adapted before the changeover.

Buy-in from members of staff: As noted earlier, the ASD team has recognized the importance of gaining acceptance of the change from members of staff. ASD continues to work with user reference groups, not only for the Academic Engine, but also other user groups for other component functions of the Arion SMS such as Enrolment, Graduation, and Admissions.

The need to standardize naming conventions: As Academic Engine effectively automates the maintenance of the academic regulations, there is a need to develop robust naming conventions for the Academic Components. The names of the components appear not only on Student Study Plans and on reports, but they also can appear as part of the qualification title on the parchment.

Need for more across university planning and consistency of processes and further alignment of timing for administrative activities: One stated problem with the current version of Arion is that it allows a great degree of flexibility in business processes. Members of staff from different schools and faculties have developed different ways to use it. With the future introduction of the Academic Engine, we need to standardize many of these approaches.

Summary

The Academic Engine is an ambitious project AUT has undertaken to provide the university with a future-oriented Student Management System. The project has changed the focus of the Arion SMS from enrolment to the student outcome. This enables AUT to more agilely cater for different offering the student may desire.

Perhaps it is best said by Kouassi, Martins, and Molar

The redevelopment of ARION around the student study plan facilitates a more student centred focus by providing enhanced personalisation and support in their progress to completion. This use or focus of an information system often has a broader objective and is designed to enhance the student experience and their overall satisfaction (2016)

The Arion's Academic Engine and Study Plans gives students a map that allows them to understand how they can achieve any outcome available at AUT and make informed choices based on that knowledge. The Academic Engine focuses on the outcome the student desires.

References

Agile Alliance. (2001). Manifesto for Agile Software Development. Online at <http://agilemanifesto.org/>.

Auckland University of Technology. (2016) 2016 Calendar. Auckland, New Zealand: Auckland University.

Kouassi, A., Martins, J. T., & Molnar, A. (2016). Customer Experience Management System at a University's Student Support Services: An Organizational Ambidexterity Perspective. In J. Martins, & A. Molnar (Eds.), Handbook of Research on Innovations in Information Retrieval, Analysis, and Management. pp. 431-450. <http://dx.doi.org/10.4018/978-1-4666-8833-9.ch016>.

Larman, Craig, & Basili, Victor R. (2003). Iterative and Incremental Development: A Brief History. Computer 36(6) pp. 47-56. <http://dx.doi.org/10.1109/MC.2003.1204375>.