Ranking DBA Programs on a Rugged Landscape

T. Grandon Gill, University of South Florida

Abstract

The paper describes the use of the elaborated action design research (eADR) method to develop a prototype of a tool that generates user-customized rankings of U.S. DBA programs. It begins by reviewing literature highlighting the weaknesses of institutional and program rankings, the source of many being the underlying rugged landscape that characterizes the program selection problem. It then references two key design science research frameworks:

I. eADR (Mullarkey & Hevner, 2019), which views design as a set of cycles (diagnosis, design, implementation, and evolution), each of which involves planning, action, evaluation, reflection and learning activities.
II. Design fitness (Gill & Hevner, 2013), which draws upon evolutionary principles to assess an artifact's decomposability, malleability, openness, embeddedness in a design system, novelty, interestingness, elegance, and usefulness.

Using the design fitness framework, it identifies appropriate criteria for evaluating the three artifacts under development: i) the list of program criteria, ii) the analytical framework used to create the ranking, and iii) the prototype itself. These criteria are based upon the proposed design objective of building a tool that can be used to test an approach to ranking DBA programs.

The paper then steps through the design process that was used to construct the three artifacts. Key design cycles are enumerated and the rationale used to evaluate each cycle is described. The final version of each artifact is presented, with an evaluation of strengths and weaknesses from a design fitness perspective. It is argued that design objectives for decomposability, malleability and openness were well supported by the prototype. The objectives of novelty and usefulness had, at best, limited support in the prototype, while elegance and embeddedness in a design system were almost entirely lacking.

In identifying the weaknesses of the prototype artifact, the lack of embeddedness and its failure to incorporate programs outside the U.S. are judged to be the most serious. It proposes that institutional support might follow two possible paths: 1) support from an individual institution, likely to be motivated by the potential to increase the program's visibility and the ability to control the data used to develop rankings, and 2) support from an industry-wide institute or consortium of programs that would likely be motivated by the desire to keep less rigorous published rankings from getting traction.

The final topic discussed is the lessons learned from framing the artifact design process as an eADR project. It concluded that the eADR framework was readily adaptable to the actual process undertaken. It also notes that the characteristics of the project--with a single developer and three interacting artifacts--made it difficult to distinguish the diagnosis, design and implementation stages of the eADR. It proposes that an alternative way of framing the design process would be to characterize the actual process as design stage cycles involving interacting artifacts. More research on this alternative characterization would be needed to justify it.

The paper concludes by asserting that the growth of DBA programs will almost certainly lead to the publication of program rankings by sources widely viewed as legitimate. It repeats its earlier argument that widespread adoption of a tool such as that described in the paper could significantly reduce the impact of such published rankings.

Introduction

In December 2015, just a couple of days before the Muma College of Business was set to welcome its second cohort to the program, the following headline appeared in the Tampa Bay business press (Brown, 2015):

> "Ranking puts USF Muma doctorate program above Harvard, UF"

That we had come in first place in the 2016 national ranking by *Top Management Degrees* came as quite a surprise to us. Being just one year old, we had yet to graduate a single participant. Moreover, in attempting to identify attributes though which the Muma DBA program could have achieved such exalted status, the best we could come up with were: 1) the waistline of its academic director, or 2) the number of low production quality videos we had placed on YouTube…possibly, the sum of the two. On that basis, we decided not to draw attention to that ranking—lest our rigor (and integrity) be called into question.

As the popularity of U.S. and global DBA programs continues to grow, program rankings are likely to proliferate. Already, *CEO Magazine* publishes a listing of "the market's premier providers" (https://ceo-mag.com/wp-content/uploads/2024/04/2024-DBA-Listing-.pdf). The *Dubai Ranking*™ (https://dubairanking.net/#toprankings) similarly ranks international programs. Eventually, it seems inevitable that widely cited business ranking outlets, such as the *Financial Times* or *USNWR,* will begin publishing their own lists.

The present paper has three goals. The first is to summarize weaknesses intrinsic to standard ranking approaches. The second is to lay out, in design terms, the characteristics of a ranking system that might address some of these flaws. The third is to describe the development of a prototype—part of an ongoing design science research (DSR) effort—that incorporates a number of these design concepts. The paper concludes by considering possible paths for further developing and disseminating the prototype.

The Paradox of Rankings

Published rankings of institutions and programs present an intriguing paradox, with their impact being inconsistent with their limited rigor. A typical ranking system is constructed by collecting a wide range of institutional or program data attributes which are then assigned positive or negative weights. These weights are summed to create a composite score, can then be used to establish a rank order. On the impact side, the resulting listing provides one of the easiest shortcuts for guiding choices between institutions and/or programs. Their impact is indisputable. A widely cited study by the National Bureau of Economic Research (NBER) found that higher USNWR university rankings led to statistically significant reductions in admissions rate (selectivity), with accompanying increases in yield (percent of accepted students who attend) and SAT scores (Monks & Ehrenberg, 1999, p. 12), confirmed in a subsequent study (Meredith, 2004).

Regarding rigor, ranking systems suffer from three serious defects. The first is methodological. Leaving aside the valid concern that an institution or program might misreport their own date to achieve a higher ranking (the second defect), numerous concerns have been raised regarding the collection, transformation and analysis of ranking data. Bachrach et al. (2017) list the following general categories of methodological shortcomings: i) asking the wrong questions or not enough of the right ones, ii) question mis-/reclassification and the interconnections among factors, iii) fundamental problems with ordinal transformations, iv) fundamental problems with attempts at unidimensionality, and vi) overaggregation, de facto heuristics, and traps. Beyond these, there is the intrinsically subjective nature through which weights are established for different institutional or program characteristics.

The second defect is that prominent rankings invite manipulation. As an example (Gill, 2010, p. 64):

> it was recently suggested that Clemson University had made a significant number of decisions—impacting tuition, class size, and faculty salaries—with the specific goal of improving its national ranking (Van Der Werf, 2009), as opposed to basing the decisions strictly on educational merit. There were also accusations that its administrators gave very low peer assessments to other schools…while rating itself very highly so as to increase its relative position (McGurn, 2009).

The final category of defect is that rankings—when used to choose between institutions or programs—make little sense. For example, consider a hypothetical ranking of colleges. For that ranking to be a useful guide for a particular individual, that individual would need to have preferences consistent with the ranking weights. Somewhat tongue-in-cheek, these would imply an individual having the following characteristics (adapted from Gill, 2010, p. 66):

- SAT scores in the top range of the 99th percentile
- Straight A grades in both humanities and sciences
- At least one outstanding extracurricular activity
- No location preference
- No religious preference and no objections to any particular religion
- Indifference to public or private education
- Indifference to post graduate requirements for military service
- No financial constraints, but nevertheless desirous of a school with excellent financial aid
- Of indeterminant gender
- Very competitive in sports, but not in any particular sport
- Very interested in academics, but indifferent to field of study
- Engaged in all social and extracurricular activities

The first three address the question of the individual's likelihood of acceptance. The remainder include the types of criteria that typically lead to higher institutional rankings. Absent such characteristics, the weights used to create a composite score are unlikely to be tuned to the applicant's needs or desires.

The rankings paradox derives from the fact that the elements that contribute to a ranking score are not decomposable. Instead, they interact both with each other and with a user's preferences. For example, the combination of attributes that make for a good party school (e.g., strong Greek community, excellent sports teams, modest workload, lax enforcement of drinking laws) might be quite different from those that make for an outstanding undergraduate seminary. It is also quite possible for the same decision maker to be attracted to very different alternative possible combination of attributes.

In situations where desirability is heavily influenced by interactions, combinations of attributes dominate main effects. In consequence, local peaks (i.e., combinations where any incremental attribute change lowers the ranking value) abound. Evolutionary biologists refer to this as a rugged fitness landscape (Kauffman, 1993). On such landscapes, the ranking effect of an attribute change in one region of the landscape can be very different from the same attribute change in a different region. For example, an applicant planning to go away to school may be very sensitive to the characteristics of a school's size and location. An applicant seeking an online program, in contrast, may be totally indifferent to size and location attributes.

College rankings attempt to deal with the existence of local peaks by creating separate rankings for broad categories of institutions. That way, small liberal arts institutions are not ranked alongside large state research universities. At the graduate level, they rank programs separately for the same reason. While such separation reduces the problem somewhat, interactions between attributes still exist within categories. Thus, individuals using rankings as the primary criteria for deciding upon an institution or program will likely rely on attribute weightings inconsistent with their preferences.

Nevertheless, as stated at the outset, rankings—even imperfect ones—are too impactful to ignore. The research described in this paper is a design science research (DSR) project that constructed a prototype of a ranking tool that addresses at least some of the weaknesses inherent to published program rankings.

Design Objectives

The objectives of DSR differ from those of positivist and interpretive research in very fundamental ways. The latter forms of research generally involve one or more research questions. Answering these questions can be done using different approaches, e.g., proposing or modifying frameworks/theory, analyzing observations, and testing hypotheses. In contrast, the DSR process is generally better served by stating one or more broad research objectives that are, in turn, achieved through the design of one or more artifacts.

Objectives and Artifacts

In the current research, the key objective was as follows:

> To develop a tool that could be employed to rank DBA programs.

The artifacts developed through the process consisted of the following:

1. A list of attribute variables that could be employed to rank DBA programs according to their suitability, along with a series of DBA program test cases.
2. An analytical framework for creating a composite ranking index based on these variables.
3. A prototype that could be used to test the efficacy of the ranking process.

The Design Process

The elaborated action design research (eADR) framework, proposed by Mullarkey and Hevner (2019), was employed to guide the research process. As illustrated in Figure 1, this process consists of a series of four stages, each of which involves five activities: planning, artifact creation, evaluation, reflection and learning. The product of each stage is one or more artifacts.

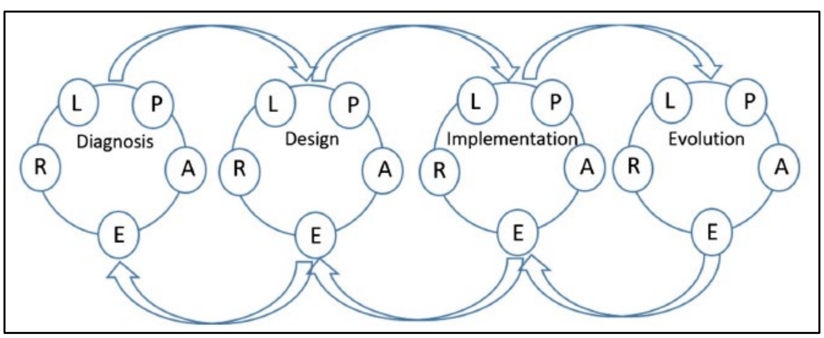


Figure 1: eADR process (Mullarkey & Hevner, 2019, p.9)

The three artifacts that were targets of the current research could be mapped to stages approximately as follows:

- *Diagnosis:* Establishing the list of variables and program test cases.
- *Design:* Developing an analytical framework suitable for performing the DBA program ranking task.
- *Implementation:* Constructing a prototype with an interface suitable for testing efficacy.

Entry to the eADR model can occur at any stage. As suggested by the outer arrows in Figure 1, it is also possible to move both forward and backward between stages, depending upon the evaluation of the artifact being considered. The potential for backward transitions proved to be particularly important during the project, since the identification of weaknesses in the implementation stage (i.e., the prototype) often necessitated changes to the algorithm and, in many cases, to the included variables and test cases.

Design Criteria for Artifacts

Guiding the evaluation process were a series of criteria used to assess the fitness of an artifact (Gill & Hevner, 2013). The criteria, derived from translating the concept of biological fitness to a design setting, are illustrated in Figure 2.

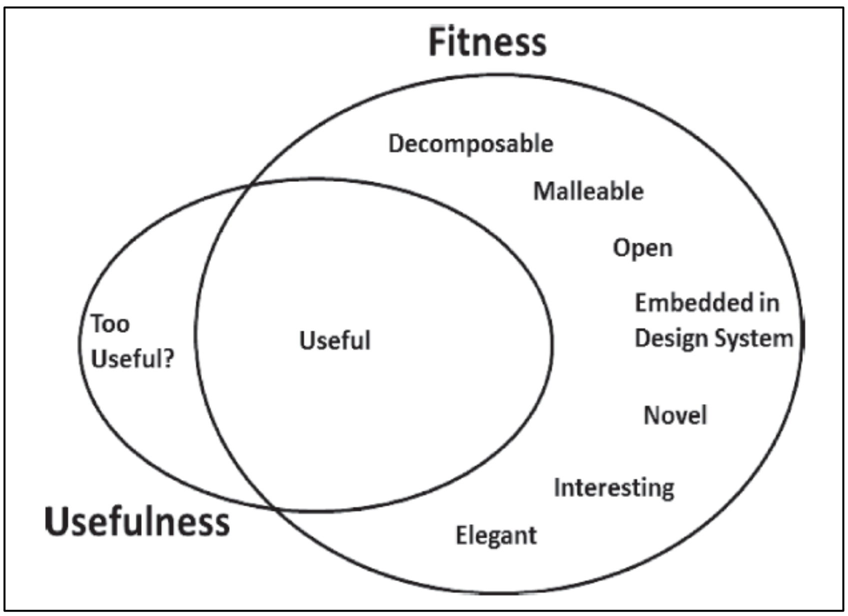


Figure 2: Criteria for assessing artifact fitness (Gill & Hevner, 2013, p. 5.15)

In the Figure 2 model, an artifact's fitness describes its ability to survive and reproduce (i.e., proliferate). A key aspect of this is the artifact's perceived *usefulness* (leaving open the possibility that an artifact could be *too useful* and consequently fail to evolve sufficiently to survive in the long run. Beyond direct usefulness, however, there are other characteristics that may contribute to fitness. These characteristics, shown in Figure 2, are as follows: *decomposable* is driven by the degree to which individual aspects of the artifact can be separated, *malleable* refers to the user's ability to adapt the artifact, *open* characterizes the transparency of the artifact's construction, *embedded in a design system* can be framed in terms of institutional support for the artifact, *novel* depends upon the lack of existing similar artifacts*, interesting* and *elegant* both address the artifact's ability to engage the user independent of the artifact's utility.

In assessing relative importance of these design characteristics, the research objectives were important. Because the third artifact—a prototype suitable for testing efficacy—would satisfy the principal goal of the research, the two attributes most associated with deploying a finished product (elegant, embedded in a design system) were deemed to be of lowest priority. Novel and interesting both seemed desirable but not necessarily critical given the early stages of research. Anticipating the likelihood that DBA programs would be resistant to rankings, as well as the need to engage users in the further development of the artifact, the characteristics of decomposability, openness and malleability were judged to be the most critical initially.

Method: The Design Process

Having established the design criteria in the diagnosis phase, the next step was to design the various artifacts. The approach chosen was iterative prototyping. This section describes the process through which the three main artifacts were created.

Initial Design Decisions

The fitness criteria identified in Figure 2 played a critical role in determining the system architecture. Specifically:

- *Decomposable:* The user should be able to identify all the program inputs and outputs and how they contribute to a ranking. Data should be separate from application logic.
- *Malleable*: Users should be able to modify and customize the application to their desired purposes.
- *Open*: Users should be able to fully inspect the underlying data and logic used to create rankings. In addition, the sources of data should be transparent.

As suggested by the figure, the design objective of being *useful* would also play a substantial role in determining the application's fitness.

Based on these priorities, a spreadsheet-based solution for the prototype implementation seemed the most logical choice:

- Data used in developing a ranking could be embedded in the application yet be displayed separately (supporting decomposability).
- Users were generally expected to be familiar with spreadsheets and likely had access to Excel (supporting malleability and usefulness). To avoid creating obstacles to non-technical users, macros and VBA code were not employed.
- Although worksheets and cells could be hidden and/or protected to prevent the user from making entry errors, experienced users could unhide/remove protections to inspect or modify functions (again supporting malleability and openness).

Simply stated, a spreadsheet deployment seemed to offer the greatest transparency—albeit at a considerable cost in terms of elegance. To ensure transparency, program test cases were assembled from data on the public websites of DBA programs. The source for each value was recorded in a separate workbook, consisting of an individual worksheet for each program.

Artifact 1: Variables and Program Test Cases

The first artifact developed for the project was a table of attributes to be used in establishing a customized ranking. Before beginning any data gathering, a list was developed. That list is presented in Appendix Table A1. Each attribute was assigned to a general category (i.e., location, meetings, institution, program, cost, feature). Such categories could be used to adjust the impact of groups of related attributes. This feature proved useful in both ranking and testing.

During the project's design cycles (Appendix Table A2), significant changes were implemented to the attribute variables and weights over time. These included:

- Adding new variables as more DBA programs were added to the prototype
- Normalizing variables across their range to make their effect more transparent to the user.
- Assigning default values to missing variables.
- Setting optional target values for variables whose optimal values were likely to be somewhere between their minimum or maximum values.

Using the initial list (Table A1) of program characteristics as a guide, the program data used to test the model evolved as follows:

- 2 (then 3) program test cases were constructed for the purpose of testing the interface and formulas.
- Data was later acquired for 30 programs by visiting each program's website. The choice of programs was driven by EDBAC membership and was limited to U.S. programs. The limitation to U.S. programs was driven by the researcher's unfamiliarity with the design of non-U.S. programs, discussed later under research limitations.
- Through exploring program websites, new program attributes were identified and added to the attribute list. Eventually, the list expanded from its original 24 attributes to 60.
- The list of programs was further expanded to 32 and a second pass of data gathering was initiated to acquire missing data and confirm existing data.

Artifact 2: Analytical Framework

The initial analytical framework employed was based on a commercial expert system developed by the researcher in the late 1980s. That system provided users (typically, high school students) with a ranked list of college choices filtered by their likelihood of admission. The source data used by the system was provided by a company that also supplied the data used by the *Arco College Guide*. Within that earlier application, nearly all college attributes were coded as binary or ordinal variables on a 1 to 5 scale. Each was assigned to a broader category (e.g., academics, extracurriculars, social, sports, location, etc.) A college's score was the sum of the weighted values of the attributes for each college:

$$\text{Score}(i) = \sum_j C_k(j) * W_j * V_{ij}$$

where C_k was the category weight associated with attribute j, W_j was the weight for the particular attribute (j) and V_{ij} was the actual value of the attribute j for college i.

Initially, the DBA prototype employed the same approach. Later, it became apparent that the approach was very cumbersome when dealing with real values, such as tuition and hours of meetings per semester. For this reason, a new analytical approach was selected. The user would enter values in their raw form, with each variable having a specified minimum and maximum value. Using this information, each variable could be normalized to a value between 0 and 1 across its range, with values falling outside the range being treated as equivalent to their minimum or maximum value, as appropriate.

The resulting formula for the program's score was as follows:

$$\text{Score}(i) = \sum_j C_k(j) * W_j * N(V_{ij}, \text{Min}_j, \text{Max}_j, \text{Target}_j)$$

where Min_j, Max_j, Target_j (if provided) were the minimum, maximum, and target values for attribute j and N() was the norming function that took the raw value (V_{ij}) for attribute j, program i, and transformed it to a value between 0 and 1 . The three norming behaviors, based on the sign of the criteria (W_j) and whether a target value was provided, are illustrated in Figure 3.

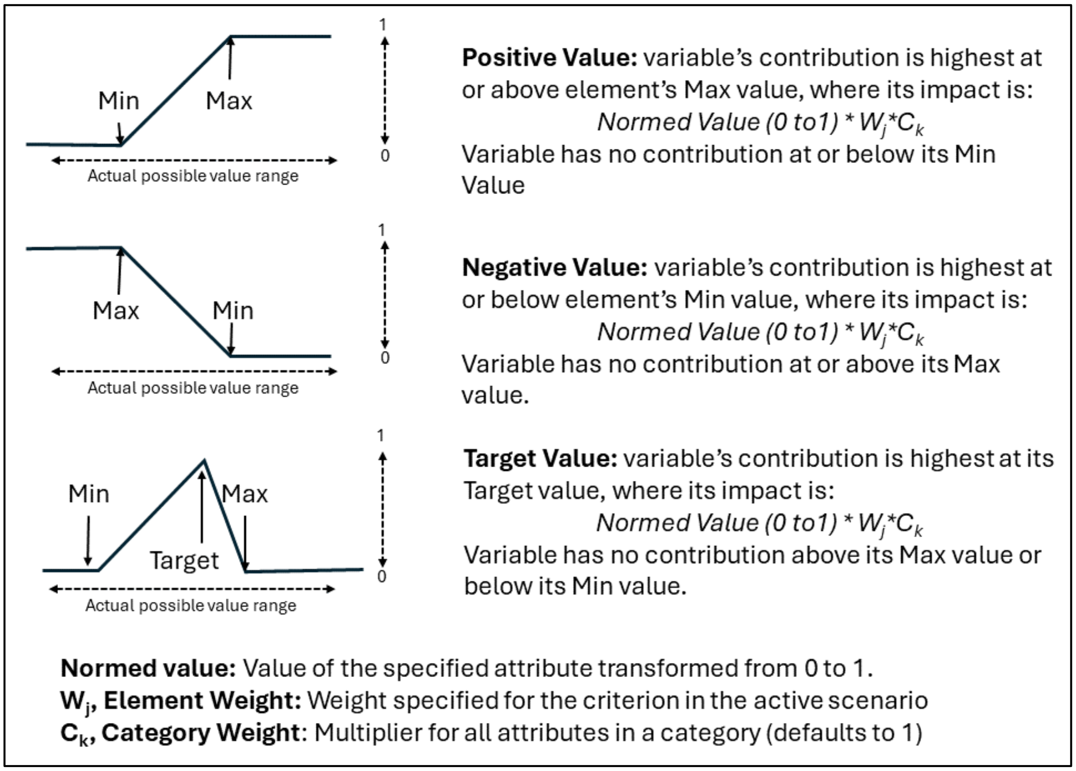


Figure 3: Impact of norming program attributes depending on positive, negative and target values for criteria

The need for different contribution behaviors was a consequence of the ruggedness of the fitness landscape for different users. For example:

- Some users might view a program's lack of a conference requirement positively, whereas others could view it negatively. This type of reversal is common in rugged landscapes. For example, if we mapped ingredients to the fitness of different recipes, adding garlic might benefit the fitness of some recipes and detract from the fitness of others.
- Certain attributes, such as the expected hours of work per week specified by the program, could be viewed as both too high (i.e., too much work) or too low (i.e., evidence of lack of rigor). For these attributes, it would be better to have the user specify a target value, with the contribution to a program's score falling off as the program's variable value differed from the user's target.

As suggested by Figure 3, regardless of an element's weight (positive, negative, targeted), in computing a program's score the variable's contribution to a program's overall score always ranged from 0 to some positive value (the absolute value of the element's weight). By avoiding subtractive criteria, it became easy to calculate each element's incremental contribution to the program's overall score.

While gathering data from actual program websites, it became clear that missing values abounded. For this reason, a default value was introduced for each variable. Use of a default was signaled by a -1 value in the program's data. To assess data integrity, the percentage impact of default values was then computed for each program.

Artifact 3: The Prototype

Figure 4 (which is likely to be readable only by zooming in) contains the final list of criteria for the version of the prototype at the time of writing. The weights and targets are those of a test case for ranking online programs and do not represent values in a likely scenario.

Measurement	Category	Low Value	High Value	Default Value	Rubric	Negative Allowed	Weight (0-100	Target	Operator (EQ,NE,LT,GT)	Value
Facilities	Location	0	5	3	0=Unacceptable, 1=Poor, 2=Sub-Par, 3=Okay,4=Good, 5=Outstanding	No	1			
Location Assessment	Location	0	5	3	0=Unacceptable, 1=Poor, 2=Disliked, 3=Okay,4=Good, 5=Outstanding	No	10			
Work Experience Required	Admissions	0	20	10	Years of work experience required	No	1	10		
Executive Experience Required	Admissions	0	20	0	Years of executive experience required	No	1	5		
Master's Degree Required	Admissions	0	1	1	0=No,1=Yes	Yes	0			
International Student Visas	Admissions	0	1	0	0=No,1=Yes (International student visas supported)	No	1			
GRE or GMAT Required	Admissions	0	1	0	0=No,1=Yes	Yes	0			
Website Comprehensiveness	Admissions	0	5	3	0=Unacceptable, 1=Poor, 2=Weak, 3=Okay,4=Good, 5=Outstanding	No	5			
Program Impression	Admissions	0	5	3	0=Unacceptable, 1=Poor, 2=Weak, 3=Okay,4=Good, 5=Outstanding	No	5			
Focus on Research and Research Methods	Curriculum	0	5	4	0=No,1=Minimally,2=Somewhat, 3=About Half,4=Mainly,5=Entirely	Yes	5			
Course Descriptions Available	Curriculum	0	1	0	0=No,1=Yes (Course descriptions are posted on website or through links to ca	No	5			
Disciplinary Specializations Required	Curriculum	0	1	0	0=No,1=Yes (Degree is specialized, such as a DBA in finance)	Yes	0			
PhD or Follow-On PhD Supported	Curriculum	0	1	0	0=No,1=Yes (A PhD option is offered, either during or after degree; N.B. Parttin	Yes	0			
Individual Electives Available	Curriculum	0	10	0	Number of elective courses available	Yes	0			
Trimester Schedule	Attendance	0	1	0	Is the program organized into trimesters (0=No,1=Yes; if so, values for meetin	Yes	0			
Face-to-Face Hours Per Semester	Attendance	0	200	60	Hours of face-to-face meetings per semester	Yes	2			
Campus Visits per Semester	Attendance	0	20	3	Number of visits required each semester	Yes	-1			
Dedicated Online Program	Attendance	0	1	0	Is the program specified to be an online program (0=No,1=Yes)	Yes	0		EQ	0
Hyflex Delivery	Attendance	0	1	0	Is the program simulcast for online and face-to-face attendance (0=No,1=Yes	Yes	0			
Synchronous Online Hours Per Semester	Attendance	0	200	0	Hours of synchronous online meetings per semester	Yes	2			
Synchronous Online Meetings Per Semester	Attendance	0	50	0	Number of synchronous online sessions per semester	Yes	-1			
AACSB Accredited	Institution	0	1	0	0=No, 1=Yes (AACSB accredition is generally considered the premier standard	No	10			
EDBAC Member	Institution	0	1	0	0=No, 1=Yes	No	2			
AAU Member	Institution	0	1	0	0=No, 1=Yes (An invitation-only organization of North American research insti	No	2			
National Ranking	Institution	0	500	500	National ranking of institution or business school (should use same ranking	Yes	-4			
HBCU	Institution	0	1	0	0=No, 1=Yes (Historically Black College or University)	Yes	0			
PhD Program	Institution	0	1	0	0=No, 1=Yes (Does the institution have a PhD program in Business or related	No	2			
Research 1	Institution	0	1	0	0=No, 1=Yes (A Carnegie classification signifying very high research activity)	No	2			
Profile Data Verified	Program	0	1	0	0=No, 1=Yes (Has the program's profile been verified by the program itself?)	Yes	0			
Executive PhD Degree	Program	0	1	0	0=No, 1=Yes (Is the degree awarded an Executive PhD vs. a DBA?)	Yes	0			
Cohort Based	Program	0	1	1	0=No, 1=Yes (Do participants proceed through the program as part of a cohort	Yes	5			
Faculty Profiles Posted	Program	0	1	1	0=No, 1=Yes (Are the faculty teaching in the program featured on the website	No	1			
Expected Years to Complete	Program	0	7	3	Number of years the program is designed for (3 is typical)	Yes	1	3		
Years Offered	Program	0	10	5	The number of years the program has been offered (caps out at 10)	No	3			
Transfer Credits Accepted	Program	0	1	0	0=No,1=Yes (Credits can be brought in from other programs)	No	0			
Estimated Workload Per Week	Program	0	50	15	Estimated hours of work per week	Yes	2	12.5		
Credit Hours	Program	0	100	60	Total credit hours of the program	Yes	0			
Number of students in typical cohort	Student	0	50	20	Typical cohort size in recent years	Yes	1	20		
Student Profiles Provided	Student	0	1	0	0=No,1=Yes	No	1			
Student Publications Featured	Student	0	1	0	0=No,1=Yes	No	1			
Percent of students who live locally	Student	0	100	50	Estimated percentage of students who live local to the program	Yes	0			
Percent of women	Student	0	100	50	Percent of women in recent cohorts	No	1	50		
Percent of underrepresented minorties	Student	0	100	20	Percent of underrepresented minorities in recent cohorts	No	1	20		
Percent of international students	Student	0	100	20	Percent of international students in recent cohorts (not necessarily on studer	No	1	20		
Number of DBA alumni hired by college fulltime	Student	0	20	0	Count of DBA alumni been hired by the college as fulltime instructors or for a	Yes	1			
Do alumni regulary teach/advise in the program?	Student	0	1	0	0=No,1=Yes (Do alumni teach DBA courses or serve on DBA dissertation comm	Yes	1			
Alumni Engagement with Program	Student	0	5	3	0=None,1=Very limited,2=Moderate,3=Typical of most programs,4=High,5=Extre	No	1			
Program Tuition	Cost	0	250000	100000	Total program tuition	Yes	-10			
Books Included	Cost	0	1	1	0=No,1=Yes (Does program cost include books)	No	1			
Meals Included	Cost	0	1	1	0=No,1=Yes (Does program cost include meals)	No	1			
Computer Provided	Cost	0	1	0	0=No,1=Yes (Does program cost include providing each participant with their	No	1			
Research Budget Included	Cost	0	25000	0	Total research and travel budget provided by the program	No	2			
Conference Requirement	Feature	0	5	0	How many academic conferences must participants attend?	Yes	1			
Dissertation Type Options	Feature	0	1	0	0=No,1=Yes (Does the program offer options besides the standard dissertatic	Yes	1			
Dissertation Groups	Feature	0	1	0	0=No,1=Yes (Does the program assign a group of students to a dissertation cc	Yes	1			
Mentor Program	Feature	0	1	0	0=No,1=Yes (Does the program have a formal process for assigning mentors?)	No	1			
In-House Journal	Feature	0	1	0	0=No,1=Yes (Does the program publish one or more journals suitable for pub	No	1			
In-House Conferences	Feature	0	1	0	0=No,1=Yes (Does the program host one or more conferences each year suital	No	1			
Cohort Selected Courses	Feature	0	5	0	0=None,1,2,3,4,5=5 and over (Number of courses selected by the cohort)	No	1			
	User									
	User									
	User									
	User									
	User									

Figure 4: Final list of criteria for the Alpha version of the prototype

Some features shown in Figure 4 were incorporated specifically to support the design objective of malleability:

1. *Two criteria were designed to be driven by user experience*: i) location assessment, and ii) program impression. These were set to the default flag (-1) for all programs. Users could replace the default values in the program table with their own assessments of perceived desirability (for location) and a rating of their experiences interacting with the program (program impression). Weighted heavily enough, these values could become the main driver of the rankings. Placing the user's thumb in the scale in this manner would not necessarily be a bad thing. For example, a case can be made that it makes sense to prefer a local program; similarly, a positive experience interacting with a program in the initial stages might auger well for the future.
2. *The final five criteria are left open for users to establish their own ranking criteria*. The process of adding attributes is relatively straightforward since a variable's impact was fully determined by settings for the columns in Figure 4. Unfortunately, the benefits of this customizability were, perhaps, less than they might have appeared. Their drawback was that, to be meaningful, users

would need to research the actual values for each program (currently 32) included in the application.

Key changes made during the design process are now described.

Incremental Changes

The target interface for the prototype evolved over many design cycles, as summarized in Table A2. The main motivator of these changes was the growing complexity of the application as more criteria and programs were added. Among these changes:

- Program data was gathered in a separate spreadsheet so that the source of each value could be logged. In some cases—such as establishing the number of hours of synchronous class time per semester—significant calculations needed to be performed to achieve relative consistency across programs. Documenting data sourcing supported the design objective of openness.
- To reduce user errors and misinterpretation of results, conditional formatting was added to highlight cells outside of established ranges (red, bold) and scores driven by default program values (blue, bold). This highlighting supported the design objective of malleability.
- To further support the design objective of openness, when presenting rankings the application displayed:
    - Program scores on the sorted list of ranked programs
    - The contribution of each category to the program's score. These scores could be compared to those of other programs.
    - The percentage of each program's score that could be attributed to default values, an indicator of trustworthiness. As default percentages grew higher, the fidelity of a program's ranking score became increasingly questionable. Indeed, on some runs a hypothetical "program" consisting entirely of default values could end up somewhere in the middle of the ranking.

Transformational Change: Implementing Scenarios

The biggest challenge presented by the interface involved the substantial number of criteria that a user needed to enter to get a suitable ranking. This was identified as a problem by program personnel and college administrators midway through the design process (see Table A2, cycle 5). To address this concern, a major departure from the original design was undertaken, the introduction of scenarios.

In the original design, users entered their own weights, targets and filtering conditions directly into the criteria page (shown earlier, in Figure 4). That approach had significant implications:

- If users wanted to test out different rankings (e.g., a ranking for online programs and one for regular programs), they had to save separate copies of the entire spreadsheet workbook.
- To get a quick ranking, they had to understand how most, or all, of the criteria impacted rankings—a significant investment in learning.
- The application could only be distributed with a single "base case".

The introduction of scenarios addressed many of these problems. Scenario worksheets (ultimately 10 in total) were added, each of which contained its own set of criteria weights. Two types of scenarios were provided:

1. *Pre-configured scenarios*, with criteria weights and filters adjusted towards a particular ranking objective (e.g., ranking online programs, ranking programs according to their similarity to PhD programs, etc.)

2. *Unspecified scenarios*, available for users to modify, as desired, to achieve their personalized ranking objectives.

When the user selected a scenario (by number, 1 through 10), the values from that scenario populated the Criteria worksheet (Figure 4), which automatically led to a recalculation of the ranking. In the final version of the prototype, the scenario number was entered in a worksheet labeled "Main", consisting of:

- A list of available Scenario worksheets
- A cell to for the user to enter a scenario number
- The resulting ranked list of programs with a breakdown by category and the default percentage contribution for each program's score.

A description of the scenarios is provided in Appendix A, Table A3.

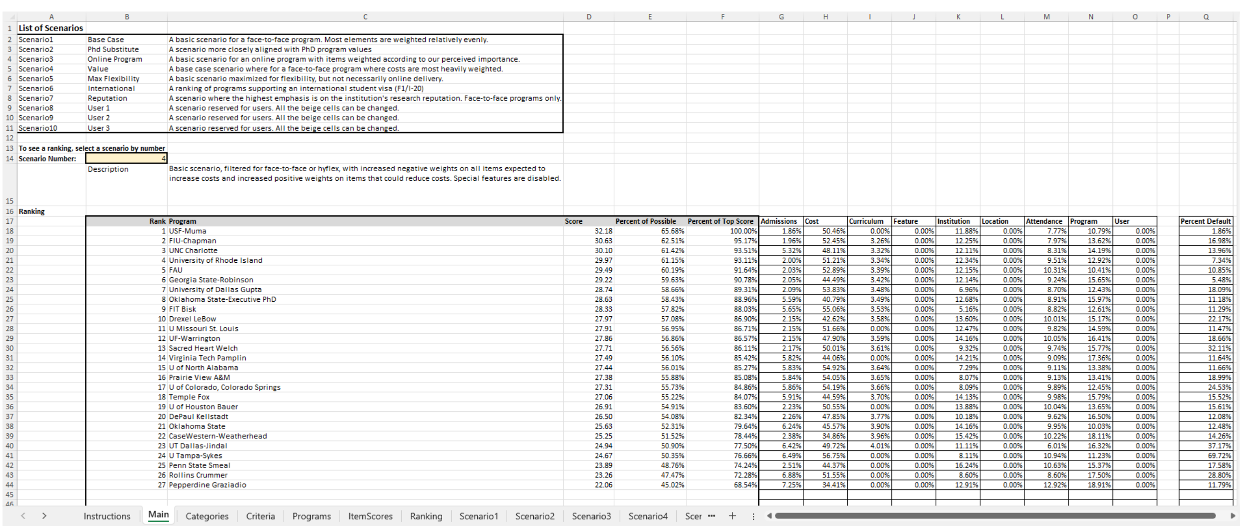
Results: The Implementation Cycles

A particular challenge of trying to fit DSR into a traditional paper format is the iterative nature of the process. Specifically, "results" are continuously being acquired as the method (i.e., the design process) proceeds. In the current paper, the results section consists of:

- A brief look at the initial Alpha prototype, and
- A self-assessment of the fitness of the initial Alpha prototype.

Artifact 3: The Alpha Prototype

After the completion of the 10th major cycle (Table A2), the initial test version of the prototype was complete. A screen capture of the Main worksheet for the Alpha prototype is shown in Figure 5.



List of Scenarios

Scenario1	Base Case	A basic scenario for a face-to-face program. Most elements are weighted relatively evenly.
Scenario2	Phd Substitute	A scenario more closely aligned with PhD program values
Scenario3	Online Program	A basic scenario for an online program with items weighted according to our perceived importance.
Scenario4	Value	A base case scenario where for a face-to-face program where costs are most heavily weighted.
Scenario5	Max Flexibility	A basic scenario maximized for flexibility, but not necessarily online delivery.
Scenario6	International	A ranking of programs supporting an international student visa (F1/I-20)
Scenario7	Reputation	A scenario where the highest emphasis is on the institution's research reputation. Face-to-face programs only.
Scenario8	User 1	A scenario reserved for users. All the beige cells can be changed.
Scenario9	User 2	A scenario reserved for users. All the beige cells can be changed.
Scenario10	User 3	A scenario reserved for users. All the beige cells can be changed.

To see a ranking, select a scenario by number

Scenario Number: 4

Description: Basic scenario, filtered for face-to-face or hyflex, with increased negative weights on all items expected to increase costs and increased positive weights on items that could reduce costs. Special features are disabled.

Ranking

Rank	Program	Score	Percent of Possible	Percent of Top Score	Admissions	Cost	Curriculum	Feature	Institution	Location	Attendance	Program	User	Percent Default
1	USF-Muma	32.18	65.68%	100.00%	1.86%	50.46%	0.00%	0.00%	11.88%	0.00%	7.77%	10.79%	0.00%	1.86%
2	FIU-Chapman	30.63	62.51%	95.17%	1.96%	52.45%	3.26%	0.00%	12.25%	0.00%	7.97%	13.62%	0.00%	16.98%
3	UNC Charlotte	30.10	61.42%	93.51%	5.32%	48.11%	3.32%	0.00%	12.11%	0.00%	8.31%	14.19%	0.00%	13.96%
4	University of Rhode Island	29.97	61.15%	93.11%	2.00%	51.21%	3.34%	0.00%	12.34%	0.00%	9.51%	12.92%	0.00%	7.34%
5	FAU	29.49	60.19%	91.64%	2.03%	52.89%	3.39%	0.00%	12.15%	0.00%	10.31%	10.41%	0.00%	10.85%
6	Georgia State-Robinson	29.22	59.63%	90.78%	2.05%	44.49%	3.42%	0.00%	12.14%	0.00%	9.24%	15.65%	0.00%	5.48%
7	University of Dallas Gupta	28.74	58.66%	89.31%	2.09%	53.83%	3.48%	0.00%	6.96%	0.00%	8.70%	12.43%	0.00%	18.09%
8	Oklahoma State-Executive PhD	28.63	58.43%	88.96%	5.59%	40.79%	3.49%	0.00%	12.68%	0.00%	8.91%	15.97%	0.00%	11.18%
9	FIT Bisk	28.33	57.82%	88.03%	5.65%	55.06%	3.53%	0.00%	5.16%	0.00%	8.82%	12.61%	0.00%	11.29%
10	Drexel LeBow	27.97	57.08%	86.90%	2.15%	42.62%	3.58%	0.00%	13.60%	0.00%	10.01%	15.17%	0.00%	22.17%
11	U Missouri St. Louis	27.91	56.95%	86.71%	2.15%	51.66%	0.00%	0.00%	12.47%	0.00%	9.82%	14.59%	0.00%	11.47%
12	UF-Warrington	27.86	56.86%	86.57%	2.15%	47.90%	3.59%	0.00%	14.16%	0.00%	10.05%	16.41%	0.00%	18.66%
13	Sacred Heart Welch	27.71	56.56%	86.11%	2.17%	50.01%	3.61%	0.00%	9.32%	0.00%	9.74%	15.77%	0.00%	32.11%
14	Virginia Tech Pamplin	27.49	56.10%	85.42%	5.82%	44.06%	0.00%	0.00%	14.21%	0.00%	9.09%	17.36%	0.00%	11.64%
15	U of North Alabama	27.44	56.01%	85.27%	5.83%	54.92%	3.64%	0.00%	7.29%	0.00%	9.11%	13.38%	0.00%	11.66%
16	Prairie View A&M	27.38	55.88%	85.08%	5.84%	54.05%	3.65%	0.00%	8.07%	0.00%	9.13%	13.41%	0.00%	18.99%
17	U of Colorado, Colorado Springs	27.31	55.73%	84.86%	5.86%	54.19%	3.66%	0.00%	8.09%	0.00%	9.89%	12.45%	0.00%	24.53%
18	Temple Fox	27.06	55.22%	84.07%	5.91%	44.59%	3.70%	0.00%	14.13%	0.00%	9.98%	15.79%	0.00%	15.52%
19	U of Houston Bauer	26.91	54.91%	83.60%	2.23%	50.55%	0.00%	0.00%	13.88%	0.00%	10.04%	13.65%	0.00%	15.61%
20	DePaul Kellstadt	26.50	54.08%	82.34%	2.26%	47.85%	3.77%	0.00%	10.18%	0.00%	9.62%	16.50%	0.00%	12.08%
21	Oklahoma State	25.63	52.31%	79.64%	6.24%	45.57%	3.90%	0.00%	14.16%	0.00%	9.95%	10.03%	0.00%	12.48%
22	CaseWestern-Weatherhead	25.25	51.52%	78.44%	2.38%	34.86%	3.96%	0.00%	15.42%	0.00%	10.22%	18.11%	0.00%	14.26%
23	UT Dallas-Jindal	24.94	50.90%	77.50%	6.42%	49.72%	4.01%	0.00%	11.11%	0.00%	6.01%	16.32%	0.00%	37.17%
24	U Tampa-Sykes	24.67	50.35%	76.66%	6.49%	56.75%	0.00%	0.00%	8.11%	0.00%	10.94%	11.23%	0.00%	69.72%
25	Penn State Smeal	23.89	48.76%	74.24%	2.51%	44.37%	0.00%	0.00%	16.24%	0.00%	10.63%	15.37%	0.00%	17.58%
26	Rollins Crummer	23.26	47.47%	72.28%	6.88%	51.55%	0.00%	0.00%	8.60%	0.00%	8.60%	17.50%	0.00%	28.80%
27	Pepperdine Graziadio	22.06	45.02%	68.54%	7.25%	34.41%	0.00%	0.00%	12.91%	0.00%	12.92%	18.91%	0.00%	11.79%

Figure 5: Main worksheet of the Alpha Prototype showing a scenario placing the heaviest emphasis on program costs

The artifact in its final version—albeit still an alpha prototype—proved to be considerably more ambitious than the researcher anticipated. With respect to the original design criteria, considerable progress was made. Specifically:

- *Decomposable*: The prototype had various components that could each potentially be used independently. For example, the Criteria worksheet could be used as a guide to the information an "ideal" program website might contain. The program data spreadsheet could be loaded into a database to facilitate considerably more sophisticated searches than provided by the application's rudimentary filtering capability.

- *Malleable*: Users with a very limited knowledge of spreadsheets could, nevertheless, build relatively sophisticated rankings using the existing scenario architecture. More sophisticated spreadsheet users could adapt the artifact to special needs in a variety of ways, adding and removing criteria, creating new scenarios, etc.
- *Open*: Everything in the spreadsheet and all DBA program data could be inspected. The only protections were those implemented to reduce the risk of accidental overwrites. These were easily disabled. Additionally, the researcher's plan is to release the application under a Creative Commons license that would allow for unlimited copying and modification (with attribution).

The *usefulness* of the application has yet to be fully established. The researcher sees four potential ways that the application might prove useful:

1. To individuals interested in a doctorate seeking to identify DBA programs that are a good fit.
2. To program directors seeking a better understanding of the design of other programs.
3. To program web managers deciding what criteria might be incorporated in program websites.
4. As a means of forestalling or reducing the influence of general published program rankings by providing a customizable alternative.

Self-Assessment of Prototype Fitness

In assessing the completed Alpha Prototype, the researcher mapped the characteristics of the artifact to the design characteristics earlier presented in Figure 2. A summary of the assessment is presented in Table 1.

Table 1: Developer's self-rating of prototype artifact against Figure 2 design criteria

Fitness Attribute	Rating (Weak, Limited, Adequate, Strong)	Justification
Decomposability	Adequate	Plus: Three artifacts—data, analytical framework, prototype—each could offer design value independently. Minus: in current implementation, changes to data typically required time consuming changes to the prototype.
Malleability	Strong	Plus: Prototype logic and data are fully exposed to user modification and all protections can be disabled. Users could adapt the tool to very specific situations that were not originally envisioned.
Openness	Strong	Plus: Prototype is fully transparent, and the intent is to license a release version using Creative Commons license.
Embedded in Design System	Weak	Minus: Minimal institutional support for the prototype currently exists and no long-term plan for release and modification exists. Even if a plan did exist, a tool developed by a particular program used to rank programs will likely be greeted with justifiable suspicion.
Novel	Limited	Plus: Different approach to program ranking problem than normally taken by publishers. Minus: Not that new. All the elements of the solution have been applied in other contexts.
Interesting	Adequate	Plus: Everyone who has seen the prototype has expressed an interest in seeing rankings. Minus: Large amount of default data makes many of its rankings questionable.
Elegant	Weak	Minus: Spreadsheet implementation makes advanced use and modification quite clunky. For most users, it requires more data that they would care to supply and generates unnecessarily detailed results.
Useful	Limited	Plus: Substantial range of users (e.g., potential applicants, program directors, program web managers). Provides an easy way to compare program data. Minus: Requires substantial time investment to take advantage of its full functionality.

Discussion

Because the Alpha Prototype was intended to be an intermediate stage in a continuing development process, three topics are the focus of this section. The first is the limitations of the existing prototype from a design standpoint. The second is possible future directions for the research project. Finally, the discussion concludes by reflecting on DSR lessons learned during the process.

Limitations of Existing Prototype

As earlier documented in Table 1, the existing prototype falls well-short of optimal fitness. The most significant weaknesses appear to be in the following areas:

- Embedded in design system
- Elegance
- Novelty
- Usefulness

Each of these limitations is now discussed.

Embedded in Design System

The embeddedness criterion reflects the degree to which the artifact is likely to acquire institutional support. The current prototype suffers from serious weaknesses in this regard. As of this writing, the prototype artifact is entirely the product of a single individual. Although the researcher's institution was informed of the project, and expressed mild support, it is unclear if resources would be made available for its long-term evolution and maintenance. Arguably, maintaining control of a ranking artifact could benefit an institution's visibility—the University of Texas, Dallas' UTD 24 list of journals and UTD research rankings (https://jsom.utdallas.edu/the-utd-top-100-business-school-research-rankings/) come to mind. But such support might also lead to suspicions regarding bias in the rankings.

Concerns regarding bias in how program criteria were selected are well justified. By determining what criteria are gathered and used to create program rankings, a program would be hard-pressed not to highlight its own unique features. For example, the criteria listed in the Features category of Figure 4 describe aspects of the researcher's program that were unique at the time they were introduced. Included in the score, that program's ranking necessarily rose. While disabling their impact (e.g., by setting the "Features" category weight to 0) was straightforward, doing so would not be the default. Nor should such innovations necessarily be ignored. Perhaps they should boost a program's score?

A particularly critical expect of the embeddedness property involves long term system maintenance. For the artifact to retain any credibility, program data would need to be updated regularly. That task would be time consuming (e.g., by the end of the Alpha Prototype design process, the researcher had invested over 50 hours in gathering test data from program websites). Realistically, without resources and the long-term cooperation of other DBA programs, the potential value of the application would diminish rapidly.

Elegance

The elegance property of a design is elusive. We may not be able to articulate why one artifact seems more "right" than another, but that does not make the distinction unimportant. Indeed, some designs are so elegant that they survive even though they were not perceived as useful when they were created. Boolean algebra is an example of such an artifact (Gill & Hevner, 2013).

The Alpha Prototype seems about as far from elegant as an application can get. Key characteristics, such as the number of DBA programs supported, the number of available criteria, and the number of scenarios provided, are hard coded into many of the spreadsheet functions, making modifications an unnecessarily painful process. Many of the formulas consist of deeply nested if-statements that are nearly impossible to decipher. In some places, complex indirect reference formulas are utilized to avoid using VBA code to copy data from one place to another. The list goes on…

These issues, along with the clunky interface, could be addressed more cleanly if the artifact were implemented as a well-designed application employing a database for programs and criteria. Such an implementation for the initial prototype, however, would have undermined the artifact's openness and malleability (particularly for non-programmers). Nevertheless, the artifact's long-term fitness would likely benefit from a more elegant approach.

Novelty

Ranking methodologies are not particularly novel. Basic sorting and filtering functionality—such as available in most online shopping applications—accomplish the same task. Advocating the artifact as a substitute for published rankings, however, is a somewhat novel approach to the DBA program choice problem. Should the other design weaknesses of the artifact be addressed, its perceived novelty might be amplified.

Usefulness

Usefulness makes a significant contribution to the fitness of most artifacts. As noted earlier in Table 1, three categories of potential user were identified for the ranking prototype:

- *DBA program applicants*, who could use the tool to compare possible programs in making their decision.
- *Program directors*, who could use the tool's program data to compare their program to other programs and to assess the potential impact of changes.
- *Program web managers,* who could use the tool's criteria page to provide insights into the content that should be included on their site.

The usefulness of the existing prototype is limited by factors that include:

- For many applicants, the universe of suitable choices is limited by geography. There are good reasons—such as convenience, time efficiency, and access to an active alumni network—why a local program may be the obvious choice for an applicant. Although the prototype easily accommodates location preference (i.e., the user can rate each program's location then weight the criterion heavily), adjusting the analysis parameters to ensure a favored location receives the top ranking makes the usefulness of the entire analysis moot. Naturally, the importance of geography (time zone aside) should be substantially lower for online programs, increasing prototype usefulness for such searches.
- The descriptions of the criteria included in the prototype are generally terse, leaving them open to misinterpretation. Without greater documentation than is currently available, it is unclear how helpful they would be in making decisions about program content. The same drawback applies to managers of a program's website.

Probably the greatest deficiency of the prototype was its failure to include international programs. That choice was motivated by time constraints and the researcher's limited knowledge of the criteria that could best be used to characterize such programs. Ironically, this very ignorance meant that incorporating global programs into the prototype would have significantly added to its potential usefulness.

Future Directions

The limitations of the existing prototype provide a roadmap for the future of the prototype and the associated DSR process. The dominant obstacle seems to be embeddedness. Acquiring institutional support for further development would be critical for the tool's survival and evolution.

There appear to be two alternative paths for acquiring institutional support available (ignoring the ever-present "abandon the project" alternative):

1. *Embed in an institution offering a DBA program*. Applying the previously mentioned UTD model, a single institution could support ongoing development and data collection for the artifact. That institution could benefit in various ways, including controlling the criteria incorporated in the model (potentially improving the institution's ranking), attracting traffic to the program's website, and increasing the national visibility of the program. Funding for the application's evolution could be supported indirectly through increased enrollments. The drawbacks of the path would be concerns relating to built-in bias and the potential unwillingness of other programs to provide and verify data required by the system.
2. *Embed in a neutral industry-wide institution or a consortium of programs*. A path in which no single institution exerts control over the application should greatly reduce concerns of bias and increase willingness to provide data. Unfortunately, this path also comes with drawbacks. Because no single institution benefits from the application, acquiring resources to support further development is likely to be problematic. In addition, key design decisions—such as what DBA programs to include in the database, what criteria need to be incorporated, and the appropriate settings for default scenarios—would demand a process for achieving consensus. Assuming the process involves forming a committee consisting mainly of faculty, one can anticipate interminable discussions with minimal resolution.

At the time of writing, both potential paths are being considered as part of the ongoing design process. The artifact consisting of the current paper is the focus of a design cycle whose objective is exploring the second path.

Reflections on DSR Process

Beyond the creation of a prototype ranking artifact, a significant personal goal for the researcher was to acquire experience applying the eADR framework (Mullarkey & Hevner, 2019) to a non-trivial development task. As previously noted, the initial expectation was that the task would be neatly broken up into three stages, each with its own artifact: diagnosis (developing the list of input data for each program), design (developing an analytical framework that could be used to rank programs) and implementation (constructing a spreadsheet-based implementation of the framework). The framework's model of moving forwards and backwards between stages modeled the actual process well; modifications to each of the associated artifacts happened continuously throughout the process. This is also consistent with observations made in other agile projects (Mullarkey & Hevner, 2019, p. 14).

In developing Table A2—which was constructed by examining intermediate prototypes saved at key stages of the development process—it became clear that the cycles were not associated with the individual artifacts as cleanly as Figure 1 suggested. Indeed, most cycles involved simultaneous modifications to at least two of the three artifacts, which proved to be highly interrelated. Moreover, the distinction of what constituted diagnosis, design and implementation throughout the process was fuzzy, at best.

The nature of the project probably accounts for much of the fuzziness. First, the project was constructed by a single developer. In consequence, the need for a formal transfer of information between individuals—as is normally required between stages in a complex project—was eliminated. Second, the broad objective of the project was to create a prototype. Since a major reason to create any prototype is to test out a design, such an objective necessarily blurs the distinction between design and implementation.

Given the context, the entire project might alternatively have been framed as iterations of a design cycle. Based on the actual design experience, however, each cycle tended to involve multiple artifacts. The eADR process description provided by Mullarkey and Hevner (2019) does not preclude multiple artifacts being refined within a single cycle. The paper's examples, however, normally involve a single artifact per cycle (although different cycles within the same ADR stage may involve different artifacts). For projects like the ranking prototype, it might be useful to explicitly model multiple "interacting artifact" cycles, as illustrated in Figure 6.

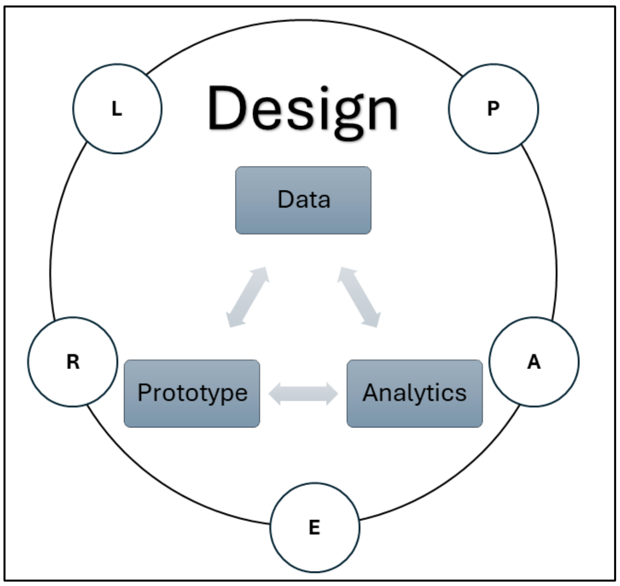


Figure 6: Design cycle involving multiple artifacts

Conclusions

If DBA programs continue to grow in popularity, the dissemination of rankings by credible publishing organizations is inevitable. The assumption motivating the current research is that by offering a tool that approaches the ranking problem in a (somewhat) sensible manner, we may be able to forestall the adverse impacts of such published rankings on the DBA program community. Otherwise, the goals of acquiring great applicants and the prestige associated with high rankings could lead to more intense, zero-sum competition between programs.

The creation of the ranking tool prototype described here demonstrates the technical feasibility of allowing users to co-create their own rankings for DBA programs. Its attractiveness to actual users remains untested, as does its long-term development path. Exposing the broader DBA community to the tool in its early stages and acquiring feedback is an important step in its evolution.

References

Bachrach, D.G. et al. (2017). On academic rankings, unacceptable methods, and the social obligations of business schools. *Decision Sciences, 48(3)*. 561-585.

Brown, J.L. (2015, 8 December). Ranking puts USF Muma doctorate program above Harvard, UF. *Tampa Bay Business Journal*. https://www.bizjournals.com/tampabay/news/2015/12/08/ranking-puts-usf-muma-doctorate-program-above.html

Gill, T.G. (2010). *Informing business: Research and education on a rugged landscape*. Informing Science Press.

Kauffman, S. A. (1993). *The origins of order*. Oxford University Press.

McGurn, W. (2009, September 15). Let's grade Wall Street like colleges. *The Wall Street Journal*, p. A19.

Meredith, M. (2004). Why do universities compete in the ratings game? An empirical analysis of the effects of the US News and World Report college rankings. *Research in Higher Education, 45,* 443-461.

Monks, J., & Ehrenberg, R. G. (1999). The impact of US News and World Report college rankings on admission outcomes and pricing decisions at selective private institutions. *NBER Working Paper Series, Working Paper 7227*. 1-13. Retrieved from https://www.nber.org/system/files/working_papers/w7227/w7227.pdf

Mullarkey, M.T. & Hevner, A.R. (2019). An elaborated action design research process model, European Journal of Information Systems, 28:1, 6-20. https://doi.org/10.1080/0960085X.2018.1451811

Van Der Werf, M. (2009, June 3). Researcher offers unusually candid description of university's effort to rise in rankings. *Chronicle of Higher Education*. Retrieved from http://chronicle.com/daily/2009/06/19270n.htm

Appendix

Table A1: Initial Criteria for Data Gathering

Category	Measurement	Rubric	Negative Allowed	Weight (1-10)	Status
Location	Geography	0=Unacceptable, 1=Poor, 2=Disliked, 3=Okay,4=Good, 5=Outstanding	No	3	Ok
Location	Campus Setting	0=Unacceptable, 1=Poor, 2=Disliked, 3=Okay,4=Good, 5=Outstanding	No	3	Ok
Location	Convenience	0=Unacceptable, 1=Poor, 2=Disliked, 3=Okay,4=Good, 5=Outstanding	No	3	Ok
Meetings	Face-to-Face Hours Per Semester	0=None,1=1-10,2=11-25,3=26-50,4=51-100,5=101 and over (Hours of face-to-face meetings per semester)	Yes	5	Ok
Meetings	Campus Visits per Semester	0=None,1,2,3,4,5=5 and over (Number of visits required each semester)	Yes	-8	Ok
Meetings	Synchronous Online Hours Per Semester	0=None,1=1-10,2=11-25,3=26-50,4=51-100,5=101 and over (Hours of synchronous online meetings per semester)	Yes	5	Ok
Meetings	Synchronous Online Meetings Per Semester	0=None,1=1-2,2=3-4,3=5-6,4=7-10,5=11 and over (Number of synchronous online sessions per semester)	Yes	5	Ok
Institution	AACSB Accredited	0=No, 1=Yes (AACSB accreditation is generally considered the premier standard in the U.S.)	No	8	Ok
Institution	AAU Member	0=No, 1=Yes (An invitation-only organization of North American research institutions)	No	2	Ok
Institution	Research 1	0=No, 1=Yes (A Carnegie classification signifying very high research activity)	No	3	Ok
Program	Cohort Based	0=No, 1=Yes (Do participants proceed through the program as part of a cohort?)	Yes	8	Ok
Program	Class Size	0=None,1=1-4,2=5-9,3=10-15,4=15-25,5=26 or more (Estimated number of students in each class)	Yes	2	Ok
Program	Estimated Workload Per Week	0=None,1=1-5 hours,2=6-10 Hours,3=11-15 Hours,4=16-25 Hours,5=26 or more hours	Yes	4	Ok
Program	Credit Hours	0=None,1=1-30,2=31-40,3=41-50,4=51-60,5=61 or more (Total credit hours of the program)	Yes	4	Ok
Cost	Program Tuition	0=None,1=$1-29999,2=$30000-59999,3=$60000-89999,4=$90000-119999,5=$120000 and up (Total program tuition)	No	0	Ok
Cost	Books Included	0=No,1=Yes (Does program cost include books)	No	3	Ok
Cost	Meals Included	0=No,1=Yes (Does program cost include meals)	No	3	Ok
Cost	Research Budget Included	0=None,1=$1-1999,2=$2000-3999,3=$4000-5999,4=$6000-7999,5=$8000 and up (Total research and travel budget provided by the program)	No	5	Ok
Feature	Conference Requirement	0=None,1,2,3,4,5=5 and over (How many academic conferences must participants attend?)	Yes	5	Ok
Feature	Dissertation Type Options	0=No,1=Yes (Does the program offer options besides the standard dissertation and multi-paper dissertation?)	No	5	Ok
Feature	Dissertation Groups	0=No,1=Yes (Does the program assign a group of students to a dissertation committee?)	Yes	5	Ok
Feature	Mentor Program	0=No,1=Yes (Does the program have a formal process for assigning mentors?)	No	5	Ok
Feature	In-House Journal	0=No,1=Yes (Does the program publish one or more journals suitable for publishing DBA research?)	No	5	Ok
Feature	Other Features	0=None,1,2,3,4,5=5 and over (Provided to account for features not included in the spreadsheet)	No	0	Ok

Table A2: Summary of Major Development Cycles

Cycle	Substantive Changes	# Crite-ria	# Test / Total Cases	Evaluation
1	Base case (see Table A1)	24	2/20	Criteria need to be normalized for their impact to be understood by the user. Comparing program scores across columns was cumbersome.
2	Criteria normalized in hidden column, min and max values for data added. This allowed raw variable values to be used, rather than ordinal categories. Used Transpose() and Sort() functions to generate ranked list based on scores.	26	2/20	Certain variables, such as hours work per week, better viewed as having optimal values rather than being either negative or positive across their range. Recognized need to remove obviously unapplicable programs from sorted listing.
3	Added variable targets. Implemented a rudimentary filter capability	26	2/20	Many more variables seemed potentially relevant to ranking.
4	Added data elements, including 5 user-definable attributes	46	2/20	With more growth in the number of attributes, missing data was expected to become a problem
5	Added default value for each attribute and allowed a -1 to signal unknown values. Added conditional formatting logic to some cells, indicating out-of-range and default-driven values.	46	3/20	2 test cases proved insufficient for further testing. Demonstrated application to program leadership, who indicated that the interface was likely to be too cumbersome for general users.
6	Expanded maximum programs to 50. Gathered data on 30 programs using EDBAC membership list. Implemented numbered scenarios to make rapid ranking simpler.	46	30/50	Based on review of existing program websites, incorporating additional criteria seemed warranted. Number of intermediate spreadsheets was increasing the degree to which the interface was confusing.
7	Added 14 new criteria and began hiding intermediate worksheets to streamline the interface. Verified data for existing programs using program websites.	60	30/50	Prototype was starting to resemble a usable tool, but there were concerns that some U.S. programs might have been missed based on CEO listing. Also, the scenario interface implementation was easy to miss.
8	Gathered data for 2 additional programs. Created Main worksheet that allowed user to see key rankings for different scenarios in one place.	60	32/50	Additional scenarios and explanations were needed.
9	Added additional scenarios and cleaned up elements of the interface.	60	32/50	Running dangerously close to submission deadline for EDBAC.
10	Finalized Alpha version	60	32/50	Need to complete DSR paper artifact.

Table A3: Scenario Descriptions (copied from the Alpha Prototype)

Name (Scenario)	Title	Description
Base Case (Scenario1)	A basic scenario for a face-to-face program. Most elements are weighted relatively evenly.	Basic scenario. Can be used to copy to user-created scenarios. Unit weights on most items except those that have no clear dominant directionality.
PhD Substitute (Scenario2)	A scenario more closely aligned with PhD program values.	Basic scenario adjusted to make it as close to a PhD as possible. For example, increased weight on requiring a Master's degree, requiring GMAT/GRE, curriculum flexibility criteria, executive PhD designation, bumped targeted workload to 25, reduced cohort size, made hiring own graduates negative, bumped up research budget value, made program tuition more negative, removed or negated weights on unusual DBA features.
Online Program (Scenario3)	A basic scenario for an online program with items weighted according to our perceived importance.	Base scenario with a filter that only captures online programs. Made hours of simulcast negative to distinguish hyflex programs from pure online programs. Zeroed out location criteria.
Value (Scenario4)	A base case scenario where for a face-to-face program where costs are most heavily weighted.	Basic scenario, filtered for face-to-face or hyflex, with increased negative weights on all items expected to increase costs and increased positive weights on items that could reduce costs. Special features are disabled.
Max Flexibility (Scenario5)	A basic scenario maximized for flexibility, but not necessarily online delivery.	Basic scenario with high positive weights on hyflex (simultaneous online and in-class) delivery and negative weights on synchronous class hours and visits. Cohort structure is not weighted.
International (Scenario6)	A ranking of programs supporting an international student visa (F1/I-20).	Basic scenario filtered for international students. Since students can't work outside of the university, frequency of visits and availability of other activities are positively weighted, as are attributes that build connections with alumni and other students.
Reputation (Scenario7)	A scenario where the highest emphasis is on the institution's research reputation. Face-to-face programs only.	Basic scenario with much greater weight placed on indicators of reputation (i.e., AACSB, AAU, R1, national research ranking). Special features are zeroed out. Only face-to-face programs are considered.
User 1 (Scenario8)	A scenario reserved for users. All the beige cells can be changed.	Users: Edit this block to describe the purpose of the scenario and how it has been modified.
User 2 (Scenario9)	A scenario reserved for users. All the beige cells can be changed.	Users: Edit this block to describe the purpose of the scenario and how it has been modified.
User 3 (Scenario10)	A scenario reserved for users. All the beige cells can be changed.	Users: Edit this block to describe the purpose of the scenario and how it has been modified.