An Innovative Analysis of the Crisis Decision Support Program's Emergency Response Procedures

Course: INF1341: Systems Analysis and Process Innovation

Instructor: Arik Senderovich

Group Members:
Abdulaziz Alsinafi
Benyamin Behzad
Adanna Chigbo
Yadi He
Faria Khandaker
A Mahfouz

Department: Faculty of Information

Program: Information Systems and Design

Word Count 4800

Executive summary:

This report discusses the emergency response focused business process of the crisis decision support program (CDSP) at the Kuwait Institute for Scientific Research. The current business process at the CDSP is as follows; Government entities in need of data call into the CDSP to request the data. This triggers a number of internal, manual processes: knowledge workers checking data availability and the access rights of said entities, CDSP personnel calling appropriate government offices to request data on behalf of the requesting entities, receiving the data via email and forwarding said data to the requesters, also via email. The CDSP currently does not store any of the datasets it receives from the other agencies which leads to multiple requests for the same files. The suggested automation aims to remove that by introducing ticket generating software which collects the requester information, crisis type and required data, having the CDSP only verify the information and forward the request to the data providing party who then directly works with the requester.

The suggested innovation approaches the issues of redundancy and manual work by introducing a central database that keeps a record of all the requests coming in via the ticketing system, cross-checks the ticket information with data tables containing access information and allows the requester to download the requested dataset once all the information about the requesting party is verified.

Table of Contents

1. Context for the study

The Kuwait Institute for Scientific Research is an independent national research institute that provides applied scientific research and technological consultations to both the governmental and private sectors in Kuwait. This report focuses on the Crisis Decision Support program (CDSP), a division of the Environment & Life Sciences research center. The division's mission is to provide the governmental branches of Kuwait with the critical data they require during a crisis so informed decisions can be made.

The current issues the Crisis Decision Support Program faces involve efficiency, timeliness, and redundancy. Optimizing and partially automating the program's data request processes is the program's current goal. To that end, this study examines their emergency response process, which involves obtaining the required data from any of Kuwait's government branches and then forwarding that data to the requesting government branch, to facilitate decision-making during a crisis.

The CDSP's involvement streamlines data exchange by eliminating the need for the requesting party to know which government branch to contact for data, as well as the need for the providing party to obtain authorization to distribute this data. There are currently eighty-seven governmental bodies in the program, with each having differing levels of access to other branches' data depending on the crisis. Datasets and access rights are kept track of using an index maintained at the Crisis Decision Support Program and must be manually checked by a crisis officer before dispatching the data request. Requests are usually made by phone, and the requested data is usually sent via email, often as links to the data.

The main issue of this process is its manual nature, which leaves the system prone to human error and often leads to prolonged response times. There are issues of redundancy where the same data must be requested repeatedly as no datasets are stored within the CDSP. This process and the required time it takes to perform is extremely critical. Time saved could result in improved public safety, and lives or costs saved (from the prevention of further damage).

2. Analysis using DFDs

A. Detailed Presentation of the As-Is situation

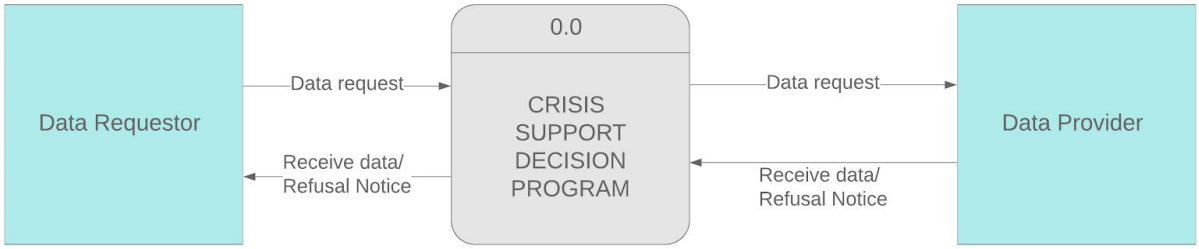


Figure 1: Context Diagram of As-Is Process

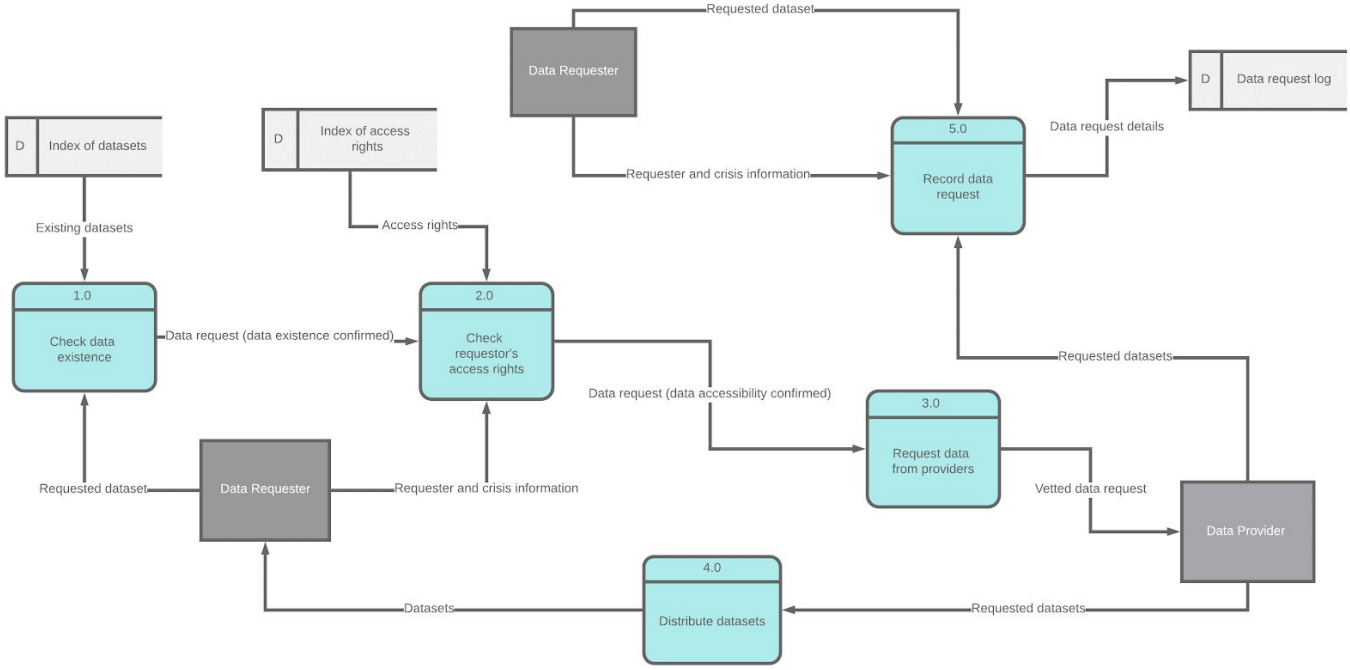


Figure 2: Level 0 Diagram of As-Is Process

As previously mentioned, the as-is process is a core process of the Crisis Decision Support Program. The data flow is initiated when a government agency requests data and is completed when the requesting agency receives the data or receives a notice of refusal (See Figure 1). The request can be decomposed into information about the crisis, the requested data, and information about the requester, such as what agency they are from. CDSP personnel must validate parts of the request separately. They identify the needed data and check to ensure it is available in an index of datasets. This index is currently a physical, analog document; checking it is time-consuming. Once the existence of the requested data has been established, CDSP personnel then have to make sure the requester has access rights to the data,

given the agency and the crisis. For example, an agency may have access rights in the case of extreme weather, but not in the case of a civil disturbance. This check also uses an analog index, physically kept with the dataset index, though CDSP staff conceptualize the two as separate data stores.

After verifying the existence of the requested data and the appropriate access rights, the request is then forwarded to the appropriate data provider, typically a government agency that maintains the data. The data provider sends the required data back to CDSP, who then distribute it to the requester. Distribution usually just entails forwarding the email containing the data to the requester.

This As-Is Data Flow Diagram depicts the happy-path flow of information. It is possible that no data is available or that the requester does not have the appropriate access rights to receive it. In this case, the result of the respective confirmation steps will be an empty list of datasets. The output of the data distribution step will be a rejection message.

Regardless of the outcome of the data availability and accessibility checks, CDSP creates a record of the request in a request log. This record includes the date of the request, the requester, what data was requested, what data was returned, and who supplied the data. The request log is currently also a physical, analog document. At this point the data flow is considered complete.

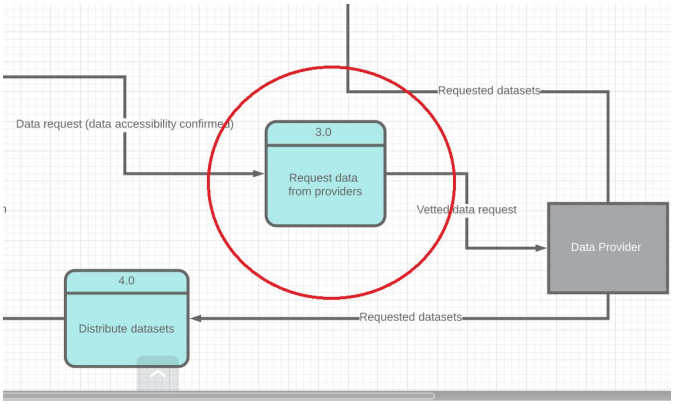


Figure 2.1: Process of redundancy in As-Is situation

Issues with the as-is process

The data flow diagram oversimplifies the current request process in some ways. The request transmission is often messier and more informal than the diagram presents, and is usually via a phone call. Furthermore, the physical nature of the data stores is an issue for information retrieval and management. Confirming data availability and access rights takes longer than it would if the indices were digitized.

There is also redundancy in the process: data that may have been sent via the CDSP to a different government entity previously would have to go through the whole cycle again before being dispatched to another requesting party, as the CDSP does nott store datasets within the organization (See figure 2.1).

The data distribution step is another point of redundancy. CDSP personnel typically forward the data and links that the providing agency sends. It would be more efficient if the data provider sent this information directly to the requesting agency.

B. A Summary of To-Be Alternatives Considered

Type	Description	Pros	Cons
Automation	Automating the entire request validation where the ticketing software itself checks if the data is available and the requester has the appropriate access rights. If both criteria are met, the request ticket is sent directly to the data provider who can then provide access to the data requester.	Faster and more smooth process with the software taking care of the request validation's finer details	Doesn't verify crisis, data privacy not guaranteed
Automation *alternative modeled	Keeping the ticket system from the previous plan but keeping the crisis confirmation process as a manual practice.	More accurate information, less complexities	Timeliness, perhaps the phone line is busy
Automation	Automating the reply to the data requester	Resolves data request faster	A mistake or loophole in the automated system may cause delays in the business process
Innovation	Data expiry. Instead of manually updating the database with new data at the time of the request, data would expire in a given time frame, such as a month or a week, depending on the nature of the data	This would eliminate the need to check if data available in the database is up to date, Enhanced data security and protection	After data expires, it will be challenging to retain it should the need ever arise

Innovation *alternative modeled	A centralized database. If the request has been previously made, then CDSP checks with the providing agency	Ensuring that data is up to date before sending it off to the requester certifies data accuracy	Other segments of CDSP may not have immediate access to the database
Innovation	Storing crisis information in a database that would be maintained by a separate group within the CDSP.	Manual confirmation takes time and a quick reference to a crisis database could speed the process	Establishing an on-call liaison between the CDSP and database maintenance group may be difficult; introduces new security and IT requirements

C. Detailed Presentation of Two To-Be Alternatives.

Automation

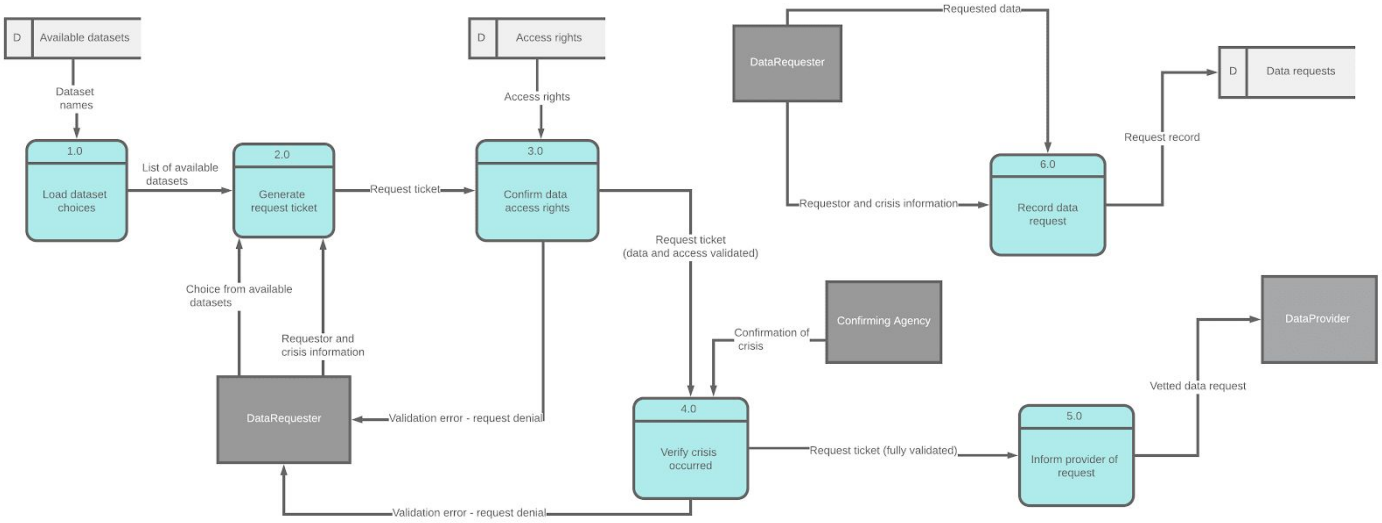


Figure 3: Level 0 Automation

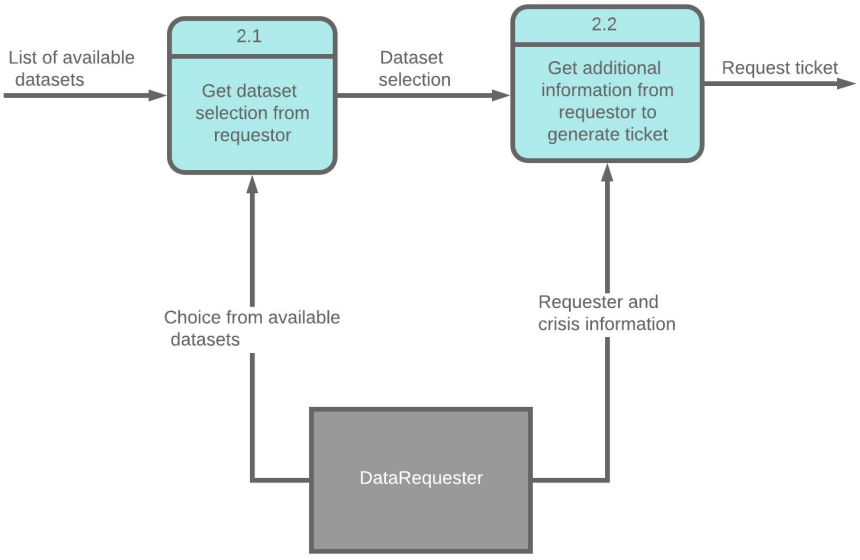


Figure 4: Level 1 diagram of Process 2

For Automation, we went with implementing a semi-automated ticketing system but decided to keep the crisis confirmation as a manual practice. Other alternatives were not considered because they leave the system more prone to security breaches.

The proposed automation process tackles the biggest problem with the as-is process: the manual nature of request validation. While the initial data flows look similar, there is a new step. Information about the crisis, the data requester, and the data needed are folded into a computer-generated service ticket that acts as a temporary data store for the duration of the process.

This computer program automates two previously manual steps, confirming the availability of the data requested and the access rights of the data requester. Data availability is guaranteed at the input stage: the requestor makes their request using a preset “drop down” menu whose options are pulled from a database of extant datasets. Access rights are validated programmatically against a data store of rules concerning what information each agency can request, and under what circumstances they can get it.

There is the risk that a malicious actor could game the request system to fraudulently access data. To guard against this, we separated out a process for crisis verification. CDSP staff would take the validated request ticket and verify that the crisis actually occurred by means like contacting other government agencies or checking internal news sites to confirm the crisis scope.

Once the ticket has undergone this final verification, it is forwarded to the data provider. The data provider then sends data directly to the requester, without going through the CDSP; that flow has been accordingly removed from the diagram. This affects the request log. Without data flowing to the CDSP, the request record consists only of information from the original ticket; details about the dataset provided are lost. This is not a deal-breaking issue. Since the purpose of the ticket validation is to make sure only extant, accessible datasets are requested, we can assume that what was requested of the data provider is what was sent to the data requester.

Innovation

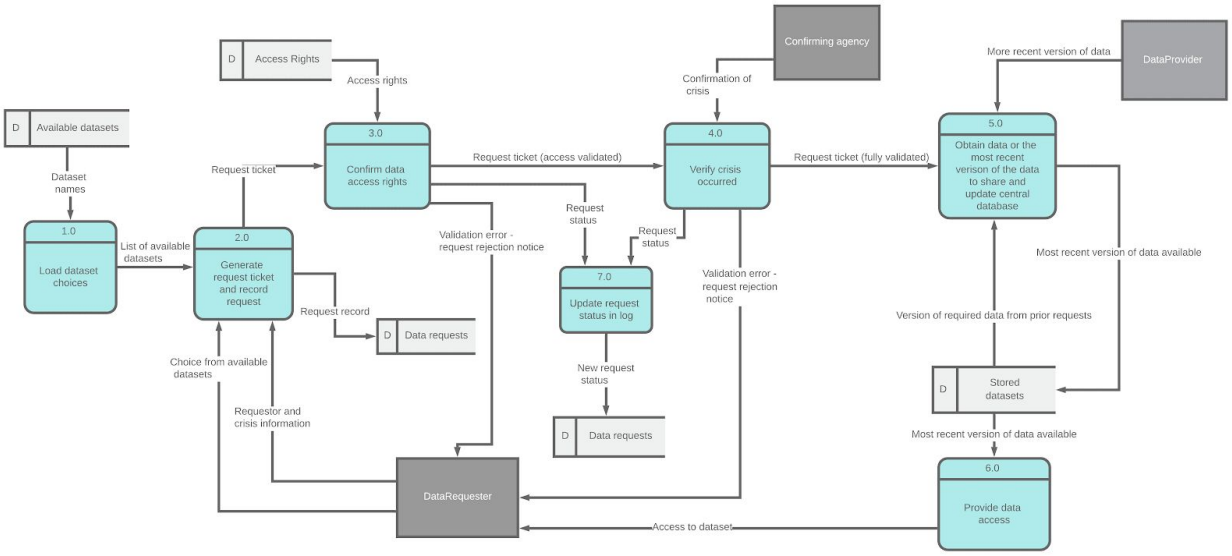


Figure 5: Level 0 Innovation Diagram

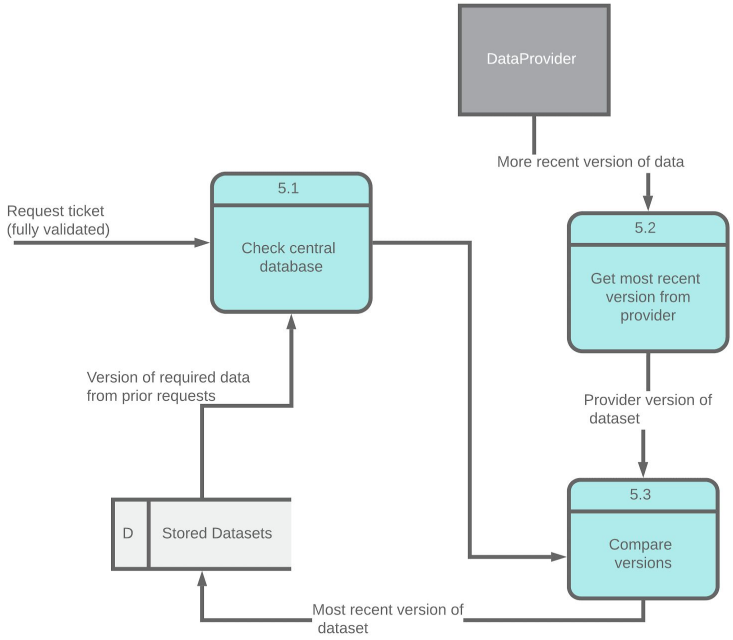


Figure 6: Level 1 diagram of Innovation Process 5

For Innovation, we propose building a central database to reduce the data request redundancy of the as-is process. Other alternatives were not considered because they take the issues of information accuracy and dataset availability from external organizations and bring it within the CDSP itself.

This process extends the changes proposed by the automation alternative (See figure 5). Dataset availability and access rights are still automatically checked against their respective digital data stores during ticket generation, but the requested datasets are now collected and stored in a centralized database within the CDSP.

Notably, the verified request ticket is not automatically forwarded to the data provider. Instead, the CDSP checks the central database to see if the same dataset has been requested and stored and asks the provider for a more recent version of the dataset, if they have one. If the dataset does not already exist in the central database, it is added. If the dataset does exist in the central database but the data provider has a newer version of it, the database is updated accordingly (See Figure 6).

Under some circumstances, CDSP personnel may know that the dataset in the central database is the newest one and simply provide that link to the requester, without asking the provider. For example, if the request includes a dataset that was provided by the agency the day before, and staff know that this data is typically only updated every few months, they may just provide that dataset without asking the provider for a most recent version.

This alternative also allows for the quick provision of ‘good-enough’ data in time-critical situations. However, redundancy remains in checking for newer data. A better solution would involve creating a pipeline to update the CDSP database when new data versions are available.

UML Class Diagram for To-Be Innovation

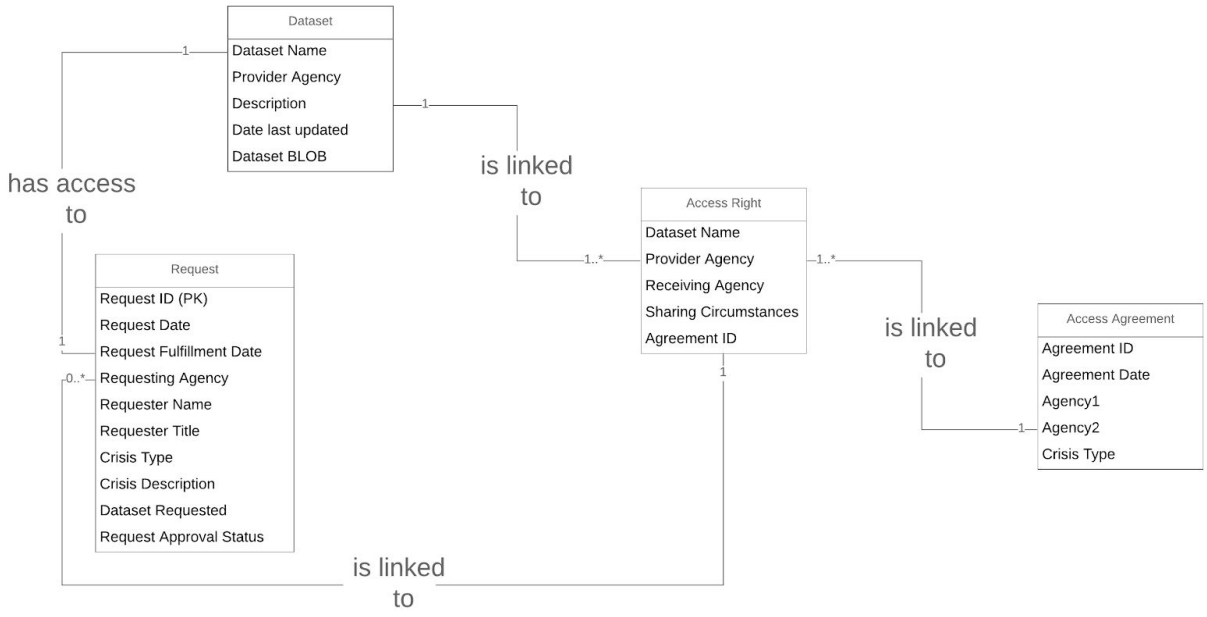


Figure 7: UML Class Diagram of To Be Innovation

The UML Class Diagram (Figure 7) displays the relationship between major data entities in the process and reflects possible schema for a CDSP Central Database. The request entity contains information collected during the ticket creation, which is then stored in the request log. A ticket represents a request for a single dataset or information product. An agency can submit multiple data requests as a result of a crisis, generating multiple tickets. Each ticket is linked, based on the requesting agency, to a single access right status. This status is linked to the Access Agreements made between organizations on data sharing. The Access rights class determines which datasets can be released and to whom It is an association class that was promoted to full class to break up the many to many relationship linking Access Agreements to Datasets. It also streamlines the cross-referencing for data access that is required for the request ticket, instead of checking both access agreements and dataset classes individually.

3. Analysis using BPMN or UML Activity Diagrams

A. Detailed Presentation of the As-Is situation

The process has three actors: A data-requesting government agency, a data-providing government agency, and the CDSP. (See figure 8)

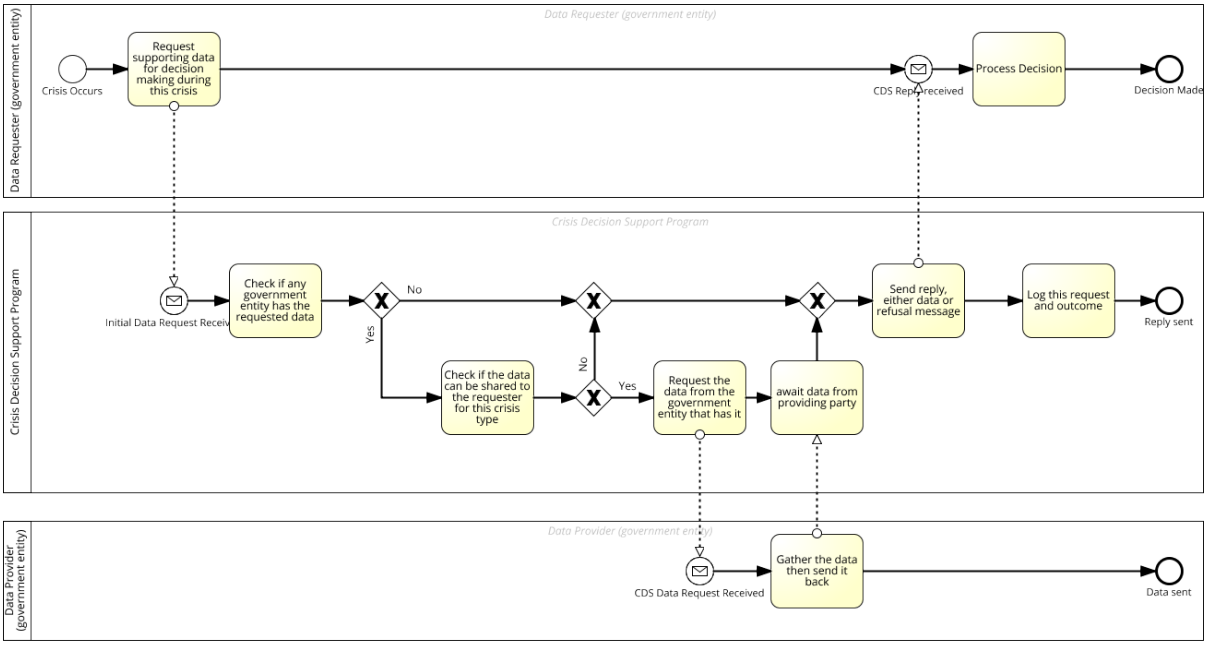


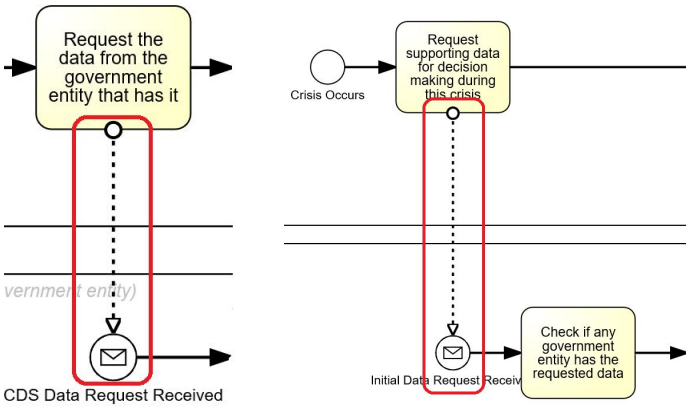
Figure 8: As-Is BPMN Model

Currently, the process begins when a crisis occurs and a government agency sends the CDSP a data request stating the current crisis and data required. The CDSP checks whether any government agency has the required data by looking through a physical index. This index has the name of every shareable dataset and the name of the government agency holding it. If the data exists, the CDSP checks a physical index containing all the data request rights and accompanying crisis types to verify whether the data requester has access rights to this data for the current crisis type. If the data requester has the needed access rights, then the CDSP requests the data from the agency that has it. The data provider complies with this request because of the established access rights agreements with the CDSP. The CDSP forwards the data they receive to the requester.

If the data does not exist or the requester does not have the rights to access it, the requesting party is sent a request denial notice, informing them of the reasons. In every case, the data request and its outcome are logged.

Unfortunately, the current process has multiple issues. The issues can be categorized into three main types: issues with communicating requests, issues with data and access rights lookup indices, and issues with data delivery.

The issues with communicating requests can be seen in the highlighted areas of figures 9 and 10.



Figures 9 and 10: Highlighted areas depicting communication issues

The main problem for this category is that almost all request communication is done via phone, which leads to problems such as busy lines or missed calls. While this is rarely an issue when trying to contact the CDSP as they have trained personnel ready to receive calls at any time of the day, it is an issue when trying to contact the data provider. It is difficult and costly to ensure that there is always staff available at every possible data-providing agency to handle emergency data request calls, considering that the majority of government agencies in Kuwait only operate during day shift time. Finally, there is the issue of having to spend time verifying the legitimacy of the caller.

The issues with data and access rights lookup indices can be seen in the highlighted areas of figure 11.

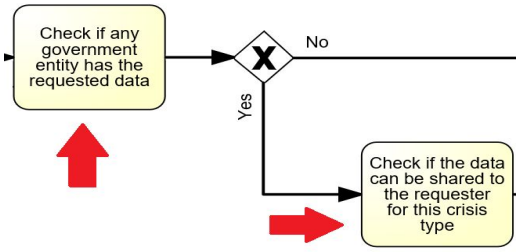
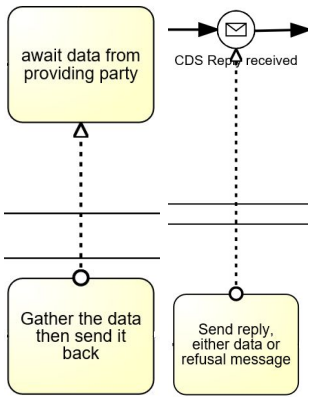


Figure 11: Highlighted areas depicting lookup issues

The main issue with the lookup of data and access rights is that searching physical indices is time-consuming. There is also the issue of potentially missing an element of the index because of human error and reporting it as data that does not exist. Manually updating the indices is also tedious, time-consuming, and error-prone.

The issues with data delivery can be seen in figures 12 and 13.



Figures 12 and 13: Sections depicting data delivery issues

The main issue here is redundancy. The data is uploaded and downloaded twice before it reaches its final destination. This issue becomes apparent when dealing with large, slow-to-transfer datasets such as satellite images. Another issue is that the entire process depends on how fast the data provider uploads the dataset. Again, this issue becomes apparent when large datasets are involved.

B. A Summary of To-Be Alternatives Considered

Type	Description	Pros	Cons
Automation	Add an automated system to check whether the data can be shared (access rights).	It replaces part of the CDSP's work, increases work and time efficiency.	There is an additional actor in the process. The system only replaces part of the job.
Automation	Add an automatic check for if the data has been requested/provided before. If it has, the system will skip checking the access rights.	The system is more automated, and it reduces redundant processes.	Access rights may change from time to time, so they need to be checked as a new request comes in.
Automation	Completely remove the CDSP process, let the system respond to requests solely.	Requesters can get a faster response, and thus respond to the crisis faster.	The system cannot confirm crisis authenticity. It has to go through the CDSP so that it can check if the crisis actually exists.
Innovation	Establish a daily updated central database at the CDSP to hold the data regardless if it has been requested or not.	Ensures the currency of the data while eliminating the need for manual confirmation of newer data versions, minimizing workload and saving time.	Resource intensive in regards to IT resources and data throughput. Not all government agencies might have the bandwidth to spare to be continuously uploading data daily.
Innovation	Implementing a priority queue for identified	Eliminates the need to run through the	Identifying highly-requested or

	highly-requested data and extremely critical data	entire process for frequently requested or critical data, reducing turn-around time for data requests of frequent or critical nature.	critical data might be difficult because of the nature and infrequent occurrences of crises, which might result in wasted effort that leads to incorrect conclusions.
Innovation	Establishing a central database at the CDSP that holds all previously provided data. This data would be forwarded to the requesting party if it is the newest version.	Eliminates the redundancies of re-requesting and re-uploading previously requested and provided data, saving time and resources.	Introduces the risk of a destructive data breach because of the central nature of the database and the sensitive nature of the data.

C. Detailed Presentation of the To-Be Alternatives

Automation

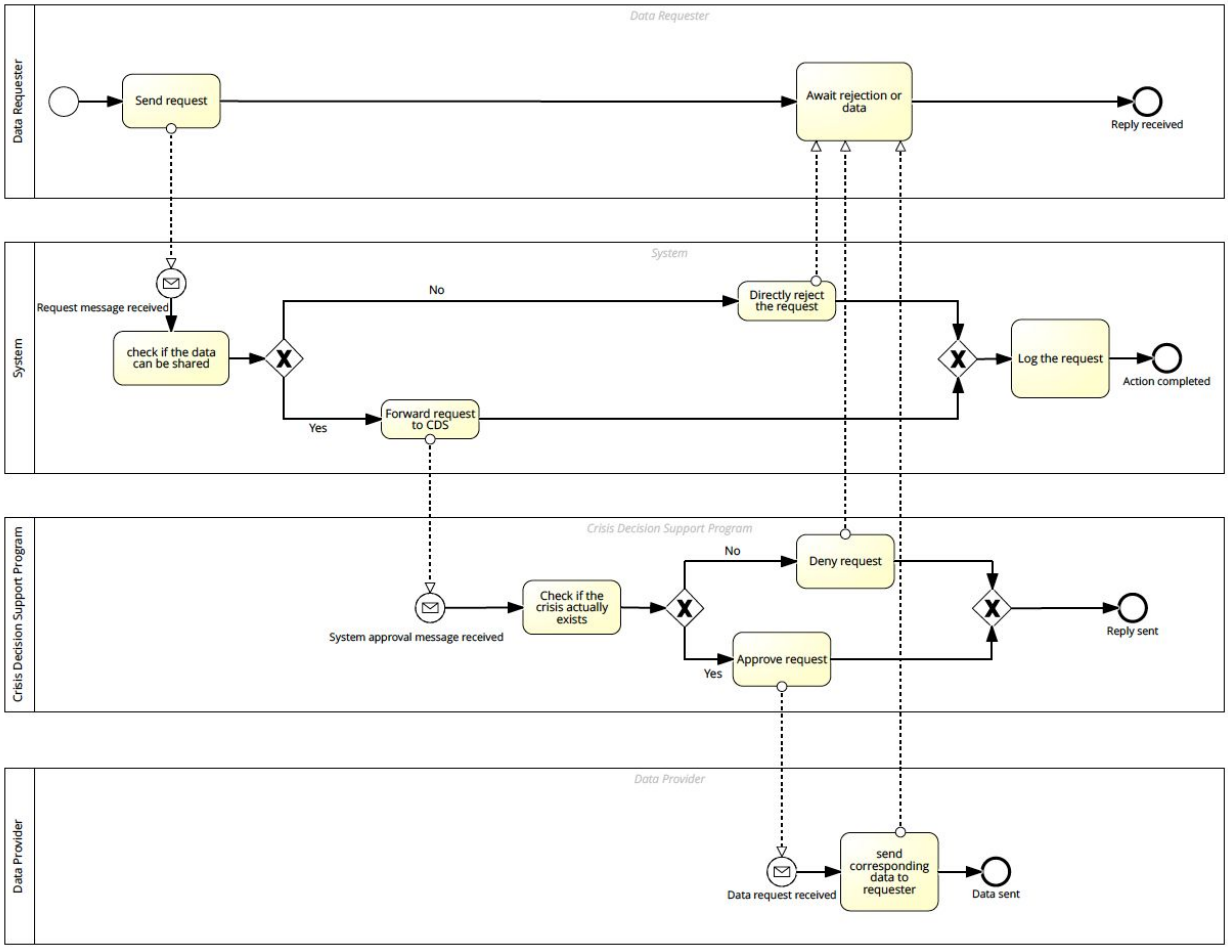


Figure 14. Automation To-Be BPMN Model

The automation to-be alternative that we chose to pursue is the first one listed in section 3B. We did not choose to go with the other two because their cons undermine important steps in the CDSP's responding process that can not be eliminated. Since the current CDSP process is completely manual, we think implementing a system would significantly improve time efficiency and reduce repetitive work for the CDSP. This automation alternative would also help satisfy the CDSP's goal to provide the governmental branches of Kuwait with the critical data they require during a crisis in a timely manner.

As shown in the BPMN model (figure 14), we proposed adding an automated system to the process. When a request is sent, the system checks the access rights first before forwarding it to the CDSP. If the requestor does not have access rights to the data, the system automatically rejects the request without forwarding to the CDSP. A drawback of the current manual process is slow response time: people cannot react faster than an automated system. With this system implemented, the requester would receive a response in less time, which is important in a crisis.

Since time is crucial, we think this system is the best automation method to achieve the CDSP's goal. However, the automation proposal is not perfect. For example, the system does not store rejected requests. With the as-is process, CDSP staff notice if the data that cannot be shared is repeatedly requested. They can note that and see if the access rights can be modified. Also, as access rights may change, information stored in the system needs to be constantly maintained by the CDSP. If the updates were not made in a timely manner, then the system may wrongly reject requests. This means that even with automation in place, manual checks are still required.

Innovation

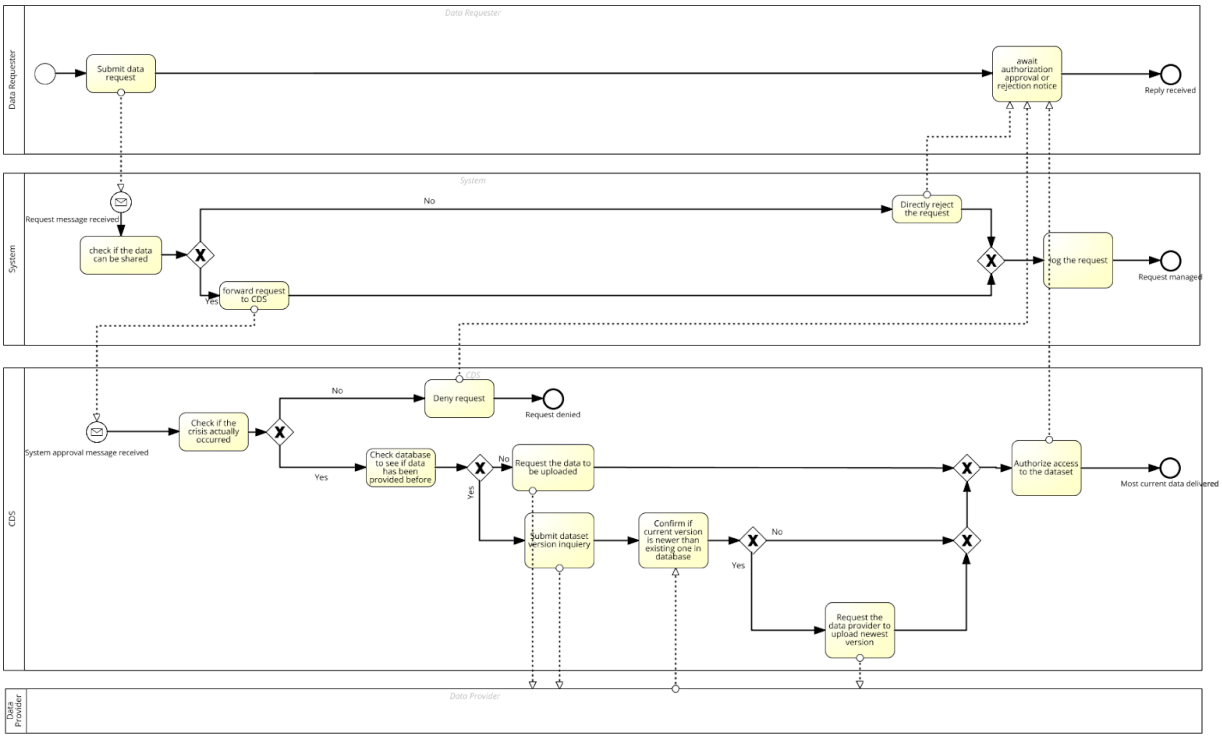


Figure 15: Innovation To-Be BPMN Model

The changes proposed in the innovation To-Be alternative include all the changes in the previously mentioned Automation To-Be process (figure 15). Additionally, a database is introduced at the CDSP where all previously provided data is stored. This changes the process by reducing the need for the data-providing party to upload the data again, unless in the case of there being a newer version of the dataset that has not been uploaded yet. This solves the case of data request bottlenecks occurring because of technical and bandwidth related issues from the data-providers side. This change can help reduce the time it takes to deliver the data to the requesting party. This change also reduces workload and frees up

resources by reducing redundancy, as a large amount of data requests received at the CDSP are repeat requests for the same datasets.

The main drawback of this proposal is that centralization introduces database administration costs and the risk of a data breach. This breach would be devastating as the database contains highly sensitive datasets and documents. Additional resources must be spent to ensure the security of this database.

Entity relationship diagram for To-Be Innovation

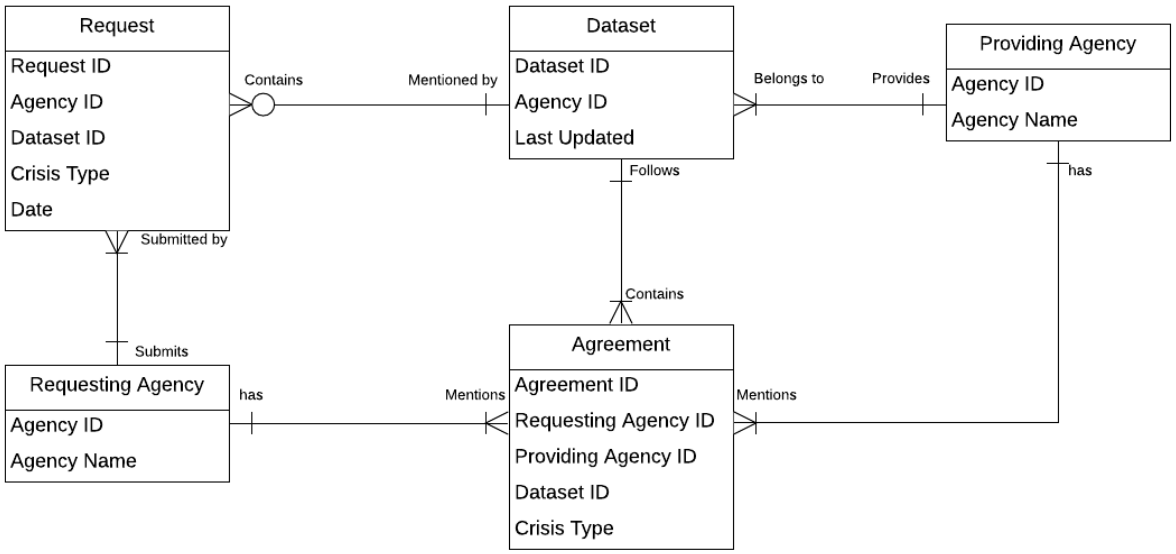


Figure 16: ER Diagram of To-Be Innovation

The Entity Relationship Diagram (see figure 16) shows the relationship that the requesting and providing agency entities have with the dataset, access agreement, and data request entities in the proposed innovation To-Be model. Requests are an associative entity representing instances of data requests that an agency might submit. The same agency can submit multiple requests. Each request is for a single dataset, but the same dataset can appear in multiple requests. The dataset entity contains the identifier of the providing agency. A single agency provides any given dataset. A dataset can belong to multiple access agreements, with each agreement mentioning only one dataset. The access agreement entity is an associative entity linking the requesting agency, the providing agency, and the dataset to be shared. It contains the crisis type as the condition the dataset can be shared in.

4. Comparison of the Two Modeling Techniques

DFD Pros:

Modeling the CDSP process using Data Flow Diagrams (DFD) helped paint a more detailed picture of the business process highlighted by BPMN models. It filled in the details that the BPMN models abstracted and allowed us to see the boundaries of the whole system with regards to user interaction. It was also easier to see where the inefficiencies lie within the system as information processing is what moves a task or process forward.

DFD Cons:

The time-independent aspect of DFDs made it difficult to evaluate efficiency gains when changing business processes. Similarly, showing multiple data flows sometimes made a process look more complicated than it appeared in the corresponding BPMN model. Initially, we assumed automating portions of the As-Is process would simplify the DFDs, but due to more information passing through different processes, the diagram looks more complex, even though the system is more efficient in processing the data requests.

BPMN Pros:

Visually showing the CDSP process using BPMN makes it easier to understand and follow. BPMN modeling made it easy to picture the business process through the perspective of the different actors. Inefficiencies and areas of potential hang ups were made apparent.

BPMN Cons:

Processes are visually represented the same way with the same size, eliminating the sense of scale and therefore do not indicate how long processes might take. There are limitations that are quickly reached to how detailed the models can be without simplifying processes or using sub-processes.

Business Process Modeling and Notation and Data Flow Diagrams complement each other; individually, one simplifies the business process, the other simplifies the flow of information. But together they paint a logical and detailed picture of the tasks and processes that occur within organizations. For example, in a BPMN model, we may see a message flow from one pool to the other, but not understand its significance until a DFD is examined.

For the DFD team, it was difficult to switch from a BPMN mindset to a DFD mindset as a BPMN mindset is about abstracting the business process and making it as concise as possible. With DFDs, however, the concept is already simplified to focusing only on dataflows; but you have to show more detail on the data flows and what processes need to happen for that flow to occur. Both modeling techniques are needed for a complete picture of an organization's business processes.

5. Methods, Activities, and Tools Used

Phone interviews were conducted with two CDSP personnel: Dr.Abdullah Al-daboos, CDSP research coordinator, and Dr. Abdullah Al-Enizi, CDSP project leader. The interviews included questions regarding how the CDSP currently operates, how crises are confirmed, where and how are the indices maintained, and issues the CDSP currently faces. A document with multiple upgrade proposal drafts for the CDSP was examined. These methods helped in collecting the information necessary for a formal examination of the CDSP's Emergency Response Process.

BPMN and DFDs were used to model the As-Is, Automation To-Be, and Innovation To-Be processes. An Entity Relationship Diagram was used to analyze the BPMN innovation data models. A UML Class diagram was used to analyze the DFD innovation model. The BPMN models were created in Signavio; the ERD, UML Class Diagram and the DFDs were created using LucidChart. Sometimes, the final model looked completely different from our initial ideas, often because we overlooked a process during our brainstorming session. These analysis tools allowed us to not only highlight issues with As-Is process, but they also allowed us to visualize both positive and negative impacts of process changes.

There were communication limitations given the CDSP's operations in another timezone and language. There was also only one point of contact to the CDSP, via a group member employed there. This made modelling DFDs especially difficult as we only had our groupmate's assignment one as a reference and had to ask him to clarify processes that would be best described by the personnel carrying them out daily. As a result, some of the terms in our DFD diagrams are vague.

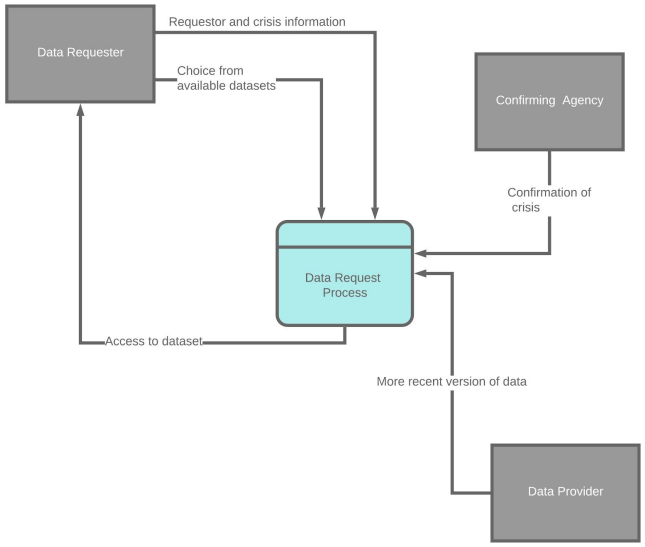
Both UML Class Diagrams and ERDs were used to model the data within the CDSP. UML Class Diagrams were used for the higher level abstraction it offers for modeling the relationship of data between the classes involved. ERDs tend to focus on relationships between different database tables. For the sake of dataflow, it is much simpler to show the relationship between each class without specifying which keys and attributes are common between them. But an ERD was also created to show the relationships between the different entities that would exist within the CDSP Central Database. Both diagrams modeled the innovation to-be alternative.

6. References

Senderovich, A.(2019) INF1341: Systems Analysis and Process Innovation, Lecture 2, Process Modelling and Analysis [PowerPoint Slides]
Senderovich, A.(2019) INF1341: Systems Analysis and Process Innovation, Lecture 6, Data Flow Diagrams [PowerPoint Slides]
Senderovich, A.(2019) INF1341: Systems Analysis and Process Innovation, Lecture 7, ERD and UML Class Diagrams [PowerPoint Slides]
Jacob, Jane. (2018, March 19).What are the advantages and disadvantages of data flow diagrams. Retrieved from www.quora.com

Appendices

Innovation To Be Context Diagram:



Statement of Individual Contributions:

BPMN Team: Abdulaziz Alsinafi, Adanna Chigbo, Yadi He

Abdulaziz Alsinafi: Created the ERD model and write up. Wrote the detailed presentation of the As-Is BPMN process, BPMN pros and cons. Altered the innovation model.

Adanna Chigbo: Created the innovation BPMN model.Wrote half of section 3.C.

Yadi He: Created the automation BPMN model and the To-Be alternatives table. Wrote half of section 3.C. Helped with editing and formatting the overall report.

DFD Team: Faria Khandaker, A Mahfouz, Ben Behzad

Faria Khandaker: Creating As-Is Context DFD and Innovation Level 0 DFD, working with A to edit and give input to rest of the DFD Diagrams and the UML Class Diagram, making edits to DFD write-up, DFD pros and cons write up, executive summary, contributed to methods, activities and tools write up and formatting of the report.

A Mahfouz: Created As-Is Level 0 DFD, Automation Level 0 DFD, Innovation Level 1 DFD, UML Class Diagram derived from data flows; co-created Innovation Level 0 DFD with Faria; drafted DFD analysis writeup; helped articulate some alternative solutions.

Ben Behzad: creating the DFD alternatives table, assisted with writing and editing the DFD write-up.