MOOCS ARE NP-COMPLETE: IMPLICATIONS FOR PRICING POLICY, DIGITAL LEARNING TRANSACTION STREAMS AND BUSINESS MODELS

Brian Subirana

Auto-ID Laboratory, Massachusetts Institute of Technology (UNITED STATES)

Abstract

Why is the educational industry so slow in adopting crowd content opportunities? This contrasts with transaction streams observed in other industries such as advertising, music, encyclopedias and travel. To answer this question, and informed by a literature review, we develop a model of transaction streams in the MOOCs industry and prove that MOOCs pricing, optimal design and best-price search are NP-complete problems. The proof suggests computational complexity is what prevents aggregation services and provides additional insights into what should be pricing policies that foster transaction streams in digital learning. In particular, it suggests three business models to develop the full value creation potential of transaction streams in education: first, one open to aggregation at no cost (very open or open source software model); second a model centrally controlled by one academic institution (flat fee or Spotify model); finally, we propose a class of simplified pricing models (and introduce one of them, MOOCIX) to enable a third route (legal programming or blockchain model) that would enable business models with financial sustainability for the MOOC industry by creating a blockchain-based “Internet of Value” around the learning industry. We contend existing models such as Open Courseware and CCX are timid approaches that do not fall in any of the three categories above and, thus cannot explode because of computational complexity constraints. To overcome these hurdles we recommend Open Courseware, for example, could change its licensing to allow for-profit revenue from improvements (i.e. a-la-GNU-GPLx), and CCX could evolve into a Spotify-type “flat” model. We conclude reviewing briefly possible avenues of future research.

Moving schools to free software is more than a way to make education a little “better”: it is a matter of doing good education instead of bad education.


Keywords: Transaction Streams, Pricing, MOOCS, Value of Education, Computational Models, Complexity Theory, Open Source.

1 INTRODUCTION: HOW DO WE FINANCE “FREE MOOCS”??

Funneled by brave University officials, donors and public officials, and pioneered by MIT Open Courseware, both the urge to give to the world a free education in the form of MOOCs and the corresponding market response are unprecedented. Figures are staggering, to name but a few: 60% CAGR 2012-2016, over 200m learners, improved academic results. Not ever before an industry has made such a bold move giving away its most precious asset at no cost. The news, media and other internet players bet on advertisement, the music industry on illegal downloads and donations, encyclopedias on volunteers, nowhere did anybody give away anything for free before. It is still too early to tell if the industry will be seriously disrupted (from within) as some have suggested. Hurdles yet to be resolved by MOOC pioneers are value recognition (above incumbent degree options), mass adoption and financial sustainability. A myriad business model types have been proposed to insure long-term financial sustainability of MOOC development. Dellarocas and Van Alstyne [1] categorize about a dozen of them depending on who pays (states, students, employers, sponsors, platforms) and for what (content, analytics, activity, complementary services). Qingyun [2] suggest a taxonomy with eleven business models and associated pricing schemes. The outlook is quite uncertain given the evolution of the MOOC industry and inexistence of a dominant design. There seems to be an increase of players in MOOC digital transaction streams like in most other industry digitization processes. Kalman [3] argues that MOOCs alone are not sustainable business models unless complementary...
services are bundled within a wider digital learning offering. Cusumano [4] stresses the relationship between financial sustainability and policy regulations.

This paper focusses on what should be pricing options endowing sustainable business models such as the ones referenced above. We are attempting to break new ground since there is no research yet in the realm of optimal pricing approaches for MOOCs. Furthermore, by reviewing the literature and interviewing 83 industry agents, Hollands and Tirthali [5] found that financial sustainability is the only goal (out of 6) where MOOC pioneers have not made significant progress. The computational models presented are inspired in previous work in transaction streams [6], computational modeling of business processes, zero-entry barriers of e-commerce models [7] and legal programming challenges [8]. Several sections are based on these previous publications by the author.

The paper is organized as follows. We first review basic electronic markets literature and previous complexity results and provide a computational model of MOOCs business models to enable proving complexity results of certain market properties. We then prove that optimal pricing and finding the cheapest price for a MOOC are NP complete problems. The proofs extend naturally to other MOOC metrics such as student effort, mastery or memory retention. We then explore pricing policy recommendations for MOOC content providers, value added services providers and intermediaries. The insights from the computational model are used to outline a simplified Pricing Model we call MOOCIX. We conclude briefly mentioning areas of future research.

2 MOOCS TRANSACTIONS STREAMS AND ELECTRONIC MARKETS

A transaction is the establishment of a contract between a set of agents (such as people and firms) to perform a given action. By "contract" we mean an agreement between a set of agents to perform a course of actions (usually with detailed implicit and explicit conditions and alternative paths). Electronic contracting focuses on negotiation of the terms and conditions of the contract, and the monitoring of contract performance [9]. Examples are purchasing a cinema ticket, placing an add on a newspaper, purchasing a book, etc.

Previous work has stressed the roles of markets and hierarchies as distinct mechanisms for coordinating the transactions related to the flow of intangible goods, materials or services through adjacent steps in the value-added chain [10]. Markets coordinate the flow through supply and demand forces between different individuals and firms. By reducing the costs of coordination, Malone et al. [10] contend that the evolution of information technology is leading to a shift toward proportionately more use of markets compared to hierarchies to coordinate economic activity. They also argue that electronic markets are a more efficient form of coordination for certain classes of product transactions whose asset specificity is low and whose products are easy to describe.

According to their view, electronic markets will evolve from electronic single-source sales channels to biased markets where one of the providers uses the market transaction mechanisms in its favor, to unbiased markets, and finally to personalized markets. Personalized markets are those in which the customers can use customized aids in making their choices. For example, some airline reservation systems allow the user to set preferences such as departure time, seating assignment and rates which are then used in subsequent transactions. The airline market is therefore customized to the users - different users have different options depending on their preferences.

Bakos [11, 12, 13] analyses the impact of electronic markets through the analysis of search costs. Buyers must, directly or indirectly, pay search costs to obtain information about prices and product offerings available in the market. Electronic markets have a vast impact on search costs because of the coordinating effect of information technology. Using economic theory, he shows that this reduction in search costs plays a major role in determining the implications of these systems for market efficiency and competitive behaviour. For example, this reduction results in direct efficiency gains from reduced intermediation costs and in indirect but possibly larger gains in allocation efficiency from better-informed buyers. The benefits realized by market participants increase as more organizations join the system, leading to network externalities and resulting in increasing rents for the consumers [14]. Our findings suggest that when the complexity of electronic markets increases, entry barriers are diminished and it is not possible to compute the optimal price. This in turn results in the opposite effect than the one suggested by Bakos [11, 12, 13].

The logic of Bakos [11] is based on 5 economic characteristics of electronic market systems (search costs, increasing return, switching costs, entry costs, maintenance costs) that have been changed by market and technological developments around the Internet over the last thirty years. The first two
characteristics that he analyses have been enhanced. First, search costs as stated above are reduced
(through browsing, apps or robot shopping) and, second, the benefits realized increase as more
organizations can join the electronic markets. However, the last three have been drastically changed
by open standards and the vast market adoption of the Internet and mobile devices as a platform for
economic activities. Indeed, Bakos’ [11] third argument was that “Electronic marketplaces can impose
significant switching costs on their participants”. However, in most Internet enabled marketplaces, the
switching costs are often reduced (or nil). This is because the Internet provides an open standard for
the information transmission (i.e. a dedicated line is no longer an entry barrier as it was when he
published his research) and a program can be used to create an interface that can reduce the
switching costs by translating among alternative interface options (i.e. the interface software is no
longer always a switching barrier). The entry costs (point four of Bakos argument) defined as large
system development, and maintenance costs have also been substantially reduced. Many open
solutions exist that can be adopted and integrated into a solution with a fraction of the development
cost that has historically been necessary. Python programming is often defined as “glue” programming
because of the stock of pre-existing open source code. This explains why early success stories such
as airline reservation and hospital supply were produced in markets with big demand for immediate
and distributed coordination.

In [6] we claim that before a transaction is completed, five processes need to be enacted: player
selection, contract condition setting, contract signature, contract storage and transaction action. We
refer to them as the transaction processes [6]. Transaction streams are there defined as electronic
markets in which more than one organization control the first four transaction processes. For example,
in the Amazon Associate Program, the referring Internet site controls the player selection while
Amazon controls the rest of the transaction processes. Underneath each process many related actors
take part and create more relations and transactions as well. For example, Google may be involved in
the player selection and the contract condition setting through advertising [15]. This in turn transforms
the Amazon associate transaction into a series of transactions streams, in which more than one
organization control the first four transaction processes. This can evolve into an increasing number of
transactions where hundreds of economic agents are involved in what for the customer appears as a
single and simple transaction.

The educational field is timidly entering the transaction streams arena through MIT Open Courseware
(timid because it restricts revenue from its content) or the tool Custom Courses on edX (CCX) 3
developed by MIT in 2015 (timid because it restricts the personalization and cross-sharing of content).
In contrast with the educational field, in the insurance industry complex examples have been
developed, and a complex transaction stream can emerge when a user “clicks” on an insurance robot
aiming to purchase an insurance product [16] – we feel insurance is a good role model for academia
because “credentials” are, in a way, very related to “insurance” in that they provide a warranty to its
holder and affiliates. The robot itself calls a set of licensing agencies, endorsement agencies and other
companies such as modeling and advertising firms. Licensing and endorsement agencies in turn
search for insurance brokers that, through third-party predictive modeling, information providers and
endorsement agencies, asses what is the risk premium that should be used. The robot then collects all
the answers and provides the aggregated response to the user. Observe that insurance brokers
nurture their systems with information provided by other companies as well. This enables an
optimization of their hit-ratio while minimizing risk.

Player selection involves the selection of the economic agents that will be involved in the transaction.
An insurance contract represents an agreement between the service provider and the party insured. It
can also represent an agreement among three parties, the surety, the oblige and the principal. This
contract is called a policy and specifies the obligations of the involved parties. The policy covers
damages inflicted by the insured upon a third party if the damages were caused by an accident or
occurrence (with exceptions as specified in the policy). The insurer does not relieve the insured from
responsibility for committing malicious acts.

The contract condition setting refers to the process by which the involved parties negotiate the details
of the action to be performed. This activity is related to the interactive exchange of messages between
players involved [17], which was applied to insurance but could well be applied to academic credential.
When an academic credential is requested, certification providers calculate the minimum requirements
to be achieved by the credential and fix the assessment at an adequate level to reflect this level.
Academic providers will need to assess a service provider about past behaviour to determine

3 http://blog.edx.org/empowering-teachers-and-students-around-the-world
credential options. Degrees require an assessment of the assumed mastery and settling disputes requires judgment, neither of which are easily automated. In other words, credential purchasing transactions are, in general, too complex for current electronic market models.

Contract signature refers to the binding step in the process in which the transaction players agree on a course of action that clarifies how the transaction activities will be performed. This can be a short and standard agreement or a long, detailed and customized contract. Policies are agreements certified by the signatures of the parties. The electronic representation of a credential should include the following information:

1. The names of the parties: The name of the credential provider, the name of the content provider and a description of the oblige (if needed).
2. The subject of the credential and the learning skills covered.
3. The period during which the credential will be in force.
4. The limits of liability (an individual limit, an aggregate limit of liability or possibly no limit in the cases of endorsement or license).

Licensing and endorsements are mechanisms that may be used to gain confidence in the service provider and play the role, described by Ba et al. [18], of a trusted third party for electronic commerce transactions which could well be applied to learning. A license is a credential that indicates that a service provider is legally authorized to provide a service or certification. It indicates that the service provider has been found to meet certain minimal qualifications required by law, and that is subjected to regulations and sanctions if it violates the law. An endorsement provides assurance that a service provider meets more rigorous requirements determined by the endorser, and usually provides information about the quality of a service provider. The confidence in an endorsed service provider depends in part on one’s confidence in its endorsers. Endorsements do not provide compensation for damages incurred while interacting with service providers. They provide a mechanism for clients to better evaluate and reduce the risk involved in dealing with service providers. The concept of an academic credential, licensing and endorsements are related, differing only in the limits and source of compensation in the event of a dispute. Insurance policies can also be included to provide a contractual responsibility to the insurance providers. The development of "institutional trust" will lead to a decrease in the consumer-perceived level of risk of transactions over the Internet [19, 20].

Before a transaction contract is signed, an assurance credential is granted to a server after meeting requirements imposed by the server issuing the credentials. The credential mechanism can be more tailored to the transaction needs than standard official licenses. They can also be tailored to the user so that the transaction risk is managed dynamically by the user and not by the regulatory environment exclusively. Furthermore, proxies can be set in such a way that all the transactions realized by a given educational organization are endorsed and certified by an independent and trusted firm. Observe that the mechanism behind this process is similar to how advertising networks work but with a different purpose: instead of exchanging user demographic information to manage the transaction stream, the system exchanges endorsement and certification preferences. Insurance credentials can be defined as proxies. For some educational degrees one can assume that governments will be involved so that a degree can be pursued that is legally recognized in several countries.

The verification of the academic credential can illustrate how transaction streams may be implemented in academia following the lead taken by other industries already mentioned such as advertising and insurance. When a credential is received from a service provider, the client validates the credential in two steps. First, the proxy is verified cryptographically. It may require further interactions with others servers. Second, the information presented in the proxy is extracted and compared against the user's and application’s policy for server selection. Furthermore, the service provider must authenticate itself to the client using the authentication protocol used by the system [16]. In the most basic arrangement there are three parties, the authentication server, the service provider and the customer. We contend that through transaction streams the number of parties to obtain a degree can escalate to the thousands if proper standards are developed.

The presence of insurance credentials is a first step to improve confidence in an educational provider. This confidence must depend in part on the confidence in the endorser or accreditation provider. An extensive network of relations can be activated to create confidence. Lai, Medvinsky, & Clifford [16] proposed a system based on a network of trust relationships that includes service providers, licensing agencies, insurance providers and endorsement agencies.
The example of figure 1 is based on the insurance industry but it can also apply to the educational industry ("which we illustrate in brackets"). Client C (or "student") requests service ("learning content or accreditation service") from service provider S2. To provide this service, S2 subcontracts to service provider S1. C's confidence in the composite service depends on the assurance provided for both S1 and S2. To improve customer confidence, S1 and S2 obtain a liability insurance policy from insurance provider I1. As long as C has confidence in I1 it is assured that C will be compensated in the event of damages caused by S1 or S2. In this example, C does not have confidence directly in the insurance ("accreditation") provider, but will accept the endorsement of E3, an organization that rates insurance ("educational") companies. Client C will also find that service providers S1 and S2 are licensed by licensing agency L2, indicating that L2 has found each server competent in offering its services. The licensing authority L2 has not been endorsed directly, but is recognized as the appropriate licensing agency by C [16].

![Diagram of trust relationships between service providers, licensing agencies, and endorsement agencies.]

**Figure 1: Example Network of Trust Relationships. Source: [16]**

### 3 Modeling Digital Learning Transaction Streams and the Learning Transaction Streams Price Search Problem (LTSPSP)

In this section, we will introduce a computational model for learning transaction streams that will allow us to prove that searching for the best price is an NP-complete problem. This means that finding the best price will be increasingly complex as the size of the data to consider increases. The computational increase for NP-complete problems is such that linear increases in the input data result in exponential increases in the time needed to solve the problem. Even with abundant computational power, exponential demands soon collapse any existing (and foreseeable) computer. A well-known problem that is NP-complete is the Travelling salesman problem. Indeed, finding an optimal route (in terms of distance or time) to visit a number of cities is NP-complete. From a practical point of view this means that one needs a simplified or semi-optimal algorithm to establish the route. Observe that it would not be difficult to write a program that finds the optimal solution by trying all possible alternatives. The issue is not whether the algorithm exists to compute the solution but what are its computational requirements. If one has 100 cities, visiting all of the alternatives requires to visit the 100 factorial combinations (that is 100*99*98*97*...*3*2*1 or 9.3*10^157 quite a large number (in fact, about the square of the number of atoms in the universe). What makes exponential problems hard to manage is that a little increase in the input size results in a much greater increase in the computational requirements. For example, if instead of 100 we have 150 cities (an increase of just 50%), the above example would yield computational requirements of 1.8*10^252 years (an increase of 10^95 times!). Observe that there may be cases in which a trivial solution exists (or one without exponential requirements), for example if all cities lie in a straight line. There are many other problems that are of this nature and for which we do not know if there are non-exponential solutions. In fact, all of these problems belong to a class of problems (or computational languages) that we call NP-complete. One can prove that if there is a non-exponential solution for one of these problems, then there is a non-exponential solution for all of them. This is one of the most challenging open research problems in computer science, which is often quoted as the P=NP? question. In the rest of this section we introduce a computational model of transaction streams that enables computational complexity proofs.
We will call market agent any participant in any of the learning transaction processes and we will represent it with the notation N. A market agent service will be the digital learning good or service provided by the market agent, which in the case of MOOCs will be a basic learning asset (such as a video lecture, a test or a solved problem set). A market agent learning service price will be the price allocated to the digital learning good or service provided. Please note that this price will, in general, be different for each service provided. A given intangible digital learning good, as perceived by the end user, will be provided through a transaction stream involving a set of educational market agents: (N1, ..., Ni, ..., Nn). In the case of MOOCs, the transaction stream goods can be components of a possible class that may be personalized for a particular offering and can include payment and advertising features too. Market learning agent Ni is providing a service SERVi to another market learning agent (its client) that we call CLIi (the client itself will be one of the N's involved in the transaction stream). To provide this service, Ni requires input from a host of suppliers SUPi,1, ..., SUPi,k (note that each supplier will in turn be one of the N's involved in the transaction stream).

To determine the market agent learning service price, we need to know the context in which the service is being provided, that is the (N1, ..., Ni, ..., Nn) as well as the SERVi, CLIi, SUPi,1, ..., SUPi,k for each node Ni. We will call an agent arrangement the set (N, SERVi, CLIi, SUPi,1, ..., SUPi,k). In general, not all agent arrangements of nodes will provide a feasible learning transaction stream. We can define a function, the learning transaction stream feasibility function which determines whether within a given service context or learning agent arrangement f(N, SERVi, CLIi, SUPi,1, ..., SUPi,k):→ (T,F) a learning service can be provided or not. If the result is true (the service can be provided) then we can define, for a given agent arrangement, a price function f(N, SERVi, CLIi, SUPi,1, ..., SUPi,k):→ Price. We will present the proof with a Price function but one could include other metrics such as hours to achieve accreditation, mastery level, long-term retention or any other metric/estimate.

A learning transaction stream arrangement will be a set of agent arrangements, which together, deliver a given learning service. Thus, by transaction stream arrangement we mean the market learning agents that need to be involved, the service that each of the market learning agents needs to provide and the order in which these market learning agents need to proceed for the digital learning service to be feasible.

The Learning Transaction Streams Price Search Problem (LTSPSP) is: “given a set of learning market agents (N1, ..., Ni, ..., Nn), and a learning service to be provided, say SERVo, what is the learning transaction stream arrangement that delivers the learning service with the minimum cost”. In the next section, we show that when the number of Nodes increases, the LTSPSP problem exponentially increases in complexity and, hence, is NP-complete.

4 MOOCs TRANSACTION STREAMS ARE NP-COMPLETE

To demonstrate that the transaction streams price search problem is NP complete, in [7] we reduced the traveling salesman problem (TSP) to the TSPSP problem. This is a common method of showing that problems are NP-complete and it consists in transforming any TSP instance into another problem with two warrants [21]. The first warrant is that such transformation is done with polynomial computational requirements. The second warrant is that once the LTSPSP problem is solved the TSP problem can be solved (using the LTSPSP solution) with polynomial computational requirements. Observe that if the two warrants are satisfied and if there was a polynomial solution for the LTSPSP problem, then there would be a polynomial solution for the TSP problem that would consist in transforming TSP problems into LTSPSP problems. The rest of this section reviews that proof and extends it to MOOCs. Note that reducing the LTSPSP problem into the TSPSP problem, which would be relatively simple, is not enough because the LTSPSP problem may be very simple (e.g. we could reduce the TSP of cities on a straight line to the TSP problem but it would not prove that that class is NP complete which it is not).

* It is important that each node Ni corresponds to a single market service agent transaction. If a company is involved in more than one service to provide a transaction, we will assume that it is a “different” market agent and we will include a new node Nj to represent it.
We begin by formally introducing the TSP problem, we follow by presenting a simplified instance of the LTSPSP lending us to the reduction of the TSP problem to the LTSPSP problem. When introducing the TSP problem, we follow the notation of (Hopcroft and Ullman 1979) and (Subirana 2000). We will define a graph G as a tuple of sets, the first being a set of nodes ($x_1, ..., x_i, ..., x_n$) and the second a set of edges ($e_1, ..., e_j, ..., e_m$). Each edge in a graph will consist of triples $e_i=(x_{ik}, x_{ij}, w_{ij})$ where $x_{ik}$ and $x_{ij}$ are the two nodes of an edge and $w_{ij}$ is the weight associated to the edge. We need to find a polynomial way of reducing the TSP problem on this graph to a LTSPSP problem on a set of market agents or nodes ($N_1, ..., N_i, ..., N_n$). Given a graph $G$, a Hamilton circuit is a circuit that visits each vertex exactly once and returns to its starting point. The traveling salesman problem is given a complete graph with weights on its edges, what is the Hamilton circuit of minimum added weight? It can be proven that the TSP problem is NP-complete [21]. In fact, determining whether a graph has a single Hamilton circuit is an NP-complete problem [21].

The set of market agents that we will construct will be one in which each graph node $x_i$ corresponds to a market learning agent node $N_i$. Think of this node as a learning asset that needs to be mastered. We will simplify the transaction stream feasibility function so that, to complete a transaction stream, all nodes need to be involved in a linear order (not in a tree order as in Figure 2), following always the available edges. What this means from a practical point of view is that you need to study the material in an order as defined by the edges of the graph. The cost will be equivalent to the weight of the edge (independently of the order chosen). Figure 3 shows one such graph in which all the market agents are represented and where arrows indicate whether a company can provide a service to another and what its price is. Note that with this simplified version of the LTSPSP problem we have reduced the TSP problem to it q.e.d..

The proof illustrates some of the inherent complexities in the LTSPSP problem. Namely, that of finding an optimum combination of services. As soon as MOOCs, or even Open Course Ware opens the possibility of aggregating services the complexity of finding the optimum combination explodes. However, there are other aspects of the LTSPSP problem that are also complex. An example is that of deciding whether an agent arrangement is feasible or not as computed by the transaction stream feasibility function.

The proof illustrates some of the inherent complexities in the LTSPSP problem. Namely, that of finding an optimum combination of services. As soon as MOOCs, or even Open Course Ware opens the possibility of aggregating services the complexity of finding the optimum combination explodes. However, there are other aspects of the LTSPSP problem that are also complex. An example is that of deciding whether an agent arrangement is feasible or not as computed by the transaction stream feasibility function.
5 IMPLICATIONS FOR PRICING POLICY, DIGITAL LEARNING TRANSACTION
STREAMS AND BUSINESS MODELS

The complexity of aggregating learning content is such that even optimizing a single subject may be
computationally intractable. This means standards need to be developed to facilitate aggregation and
personalization. An example is the tool Custom Courses on edX (CCX)\(^5\) developed by MIT on 2015.
The tool may be extended to enable general learning transaction streams (very much like Amazon
enables Amazon associates since 1996) but at the cost of making optimization problems intractable.
To overcome this limitation CCX should adopt a more generic flat fee pricing structure so that cross-
teacher learning assets can be created, very much like Spotify music playlists.

If nothing is done in terms of standardization, there is the risk of not enabling personalized learning. In
fact, a bigger challenge, confused with the standardization one, is posed by the desire to use
adaptive learning techniques, which make optimization practically impossible since decisions are
made real-time based on the students’ responses [22]. This is essential since recent research points
out that everything learned is forgotten unless is used, see in this proceedings Subirana et al 2017
[23]. The Open Courseware license does not allow for-profits to use the content preventing transaction
streams that can allow personalization or even investment in content [24, 25]. We feel a license sort of
like the GNU GPL license, enabling profit but requiring open sharing, would allow investment spurring
innovation, and in turn more free digital learning assets.

Thus, the complexity of aggregation suggests 3 ways of simplifying this otherwise intractable problem:

1 Free: Transaction streams are free and very open (route suggested for MIT Opencoursecware)
2 Flat-fee: Transaction streams have a flat fee (advertising) or subscription model (Spotify/CCX)
3 Legal Programming: An internet of value/blockchain approach is used to enable micro
   transaction aggregation using a legal programming approach that includes pricing and
   accreditation services.

We introduce a basic data structure in the next section to enable this third model.

6 MOOCIX: A MOOCS PRICE CLASS

To simplify the complexity of MOOCS transaction streams in a legal programming model, a standard
learning legal programming framework needs to be developed. One possibility is to use blockchain as
a backend to allow the creation of an Internet of Value as is being attempted by the Music Initiative
(http://open-music.org/). The objective is to develop a learning market architecture that can endow
students with the tools to have an adaptive and personalized learning experience using the optimal
learning assets available. The first step is then to develop some form of standard that allows a
program to create an automated legal transaction stream that is as optimal as possible. We now
present a simple proposal that can be the skeleton of such a standard.

We suggest capturing course content and associated pricing modalities of a course by creating a
Matrix for every type of customer and course (this implies a taxonomy of customers, learning assets,
accreditation options). Such matrix would have a row for each course asset from the course that can
be bought and a column for each type of attribute that we want to capture including basic information
like price, contract URL, cost to develop, cost to “learn” and workflow status for the particular
customer. This solution has the advantage that it is computationally very efficient. However, to deal
with the full complexity of the use cases mentioned in earlier chapters, we need a more powerful data
structure that can deal with more complex arrangements.

As a starting point, we suggest MOOCIXs price class, which allows simple legal programming
inferences, benefits from object oriented programming and is based on the following data structures:

- Class learning\_asset\_URL(object): The system identifies two types of learning assets: atomic
  or compounded. Each learning asset would have a unique URI and can belong to various
  collections. Learning assets in Python (or any other suitable object oriented language) should
  be defined by a class so that collection inclusion attributes of a particular instance can be
  inherited. Attributes of a learning asset URI could be number, url, owner.

\(^5\) http://blog.edx.org/empowering-teachers-and-students-around-the-world
• **Class client**(object): The system identifies users (students, teachers, robots, analyzers) with a unique URI. Like in the class learning assets, users can belong to organizations. Organizations can be Universities, courses, degrees, companies, study groups.

• **Class cost_metrics**(object): Pricing, cost or workflow status for a given learning asset and user is defined by this class which also enables inheritance. In fact, we expect that most properties will be inherited. Please note that a given learning asset can be priced by different instances of this class. If a learning asset does not have a specific pricing, it means that it should be purchased as part of one of its parent collections. Here are some of the attributes that we propose for this class:
  
  o **Digital Element**: Contains the URI of the learning asset.
  
  o **Client**: Contains the URI of the client

  o **Pre-conditions**: This attribute would contain a Boolean function that given an instance of a client, validates the conditions that need to be met for the client to purchase the learning asset.

  o **Release-conditions**: This would calculate until when the client can use the content.

  o **Activity**: This attribute qualifies the type of activity that can be done with this learning asset. Do the class from scratch, just view a video once, enable syndication, allow deep learning, review it for a research project etc.

  o **Cost**: Associated cost related with this activity. The costs may include books that need to be purchased, time that needs to be devoted or tests that need to be performed.

  o **Pricing**: There is not one single pricing scheme.

  o **Legal programming** – contracts. What are the available licensing and contracting options (links to contracts)

  o **Associate** breakdown: What share of the revenue can you pass to your own students assuming the pre-conditions are met.

  o **Coupon**: Number, discount ID and discount services that it applies to.

  o **Workflow Order Status**: including accreditation level achieved

  o Behavioral Modelling: Services that need to be informed each time a pricing request is produced.

7 CONCLUSIONS

In this paper, we have reviewed various pricing schemes and business models proposed in the literature to provide strategic sustainability of MOOCs, and we have proven that, under very realistic assumptions, the complexity of emerging alternatives is such that basic price optimization tasks, cost optimization and optimal design are NP-Complete. For students, the total learning cost also includes the time devoted to the material. Mastery, retention and versatility are a function of student preferences and MOOC effectiveness in achieving these goals for the student. This means that when comparing MOOC pricing schemes, the cost associated with studying time must also be taken into account. This increases the casuistic of student MOOC selection process and also the pricing use case space. In this paper, we have focused on pricing. We feel there is a natural way of extending the research presented here to include a learner’s utility function.

We have also presented briefly MOOCIX, a price class that can be the basis for an efficient legal programming framework that enables learning transaction streams. The objective of this design is to make content aggregation and payment as simple as possible for the benefit of the students [24, 25]. Without such a framework, the market will most likely evolve into a non-efficient market where the full benefit of transaction streams will not emerge and education will become sub-optimal, with a few key players controlling it. A lot of subsequent research is required to design an open and efficient system.

REFERENCES


[3] Kalman, YM 2014, 'A race to the bottom: MOOCs and higher education business models', Open Learning, 29, 1, pp. 5-14, Education Source


