An Ontology-Based Temporal Scheduling and Complexity Reduction Model for Timetabling Knowledge Management

Patience Usoro Usip[*], Enobong Moses Umoren, Udoinyang Godwin Inyang, Mfonobong Maurice Ntekop

University of Uyo, Computer Science Department, Faculty of Science, 520003, Uyo, Nigeria

A R T I C L E I N F O	A B S T R A C T
	Technological tools such as Semantic Web and ontologies have contributed immensely to the reduction of most managerial complexities on daily basis. However, most resolved complexities are solely dependent on the emerging issues which are not general enough to accommodate some specific domain challenges. Temporal scheduling complexities occur in several domains including timetabling, but inter-departmental courses allocation on a general institutional timetable has not been considered. This poses serious managerial concerns and threats to the academic performance of any institution. This paper is aimed at formalizing an ontology model for analyzing temporal scheduling complexities in an existing schedule and giving optimal possible time scheduling resolutions showing their reduction rates for efficient and intelligent knowledge management. Protégé was used in modeling the domain ontology described in the resulted ontograf. The semantics of the formal ontology model is described based on Allen's interval temporal relations. The use of ontologies make way for easy and intelligent reasoning with knowledge and exposes the need for such knowledge management tool to manage the influx of the numerous data.

1. Introduction

This paper is an extension of work originally presented in the Future Technologies Conference 2016 [1], where the use of ontologies in semantic web was established as an efficient and intelligent tool for managing timetabling knowledge [2]. Domain specific concepts and some complexities were presented in figures 2, 3(a) and 3(b) of the conference paper. The Allen's interval-based temporal relations [3] were considered suitable and introduced for inferencing [4] in the sample rules used to describe possible resolutions to the highlighted complexities in the conference paper. Allen's interval-based temporal relations were introduced to handle time durations and is adapted or reused for formal description of time periods for complex knowledge management as seen in the timetable domain. However, this paper extends works done in the conference paper as it aims at formalizing an ontology model for analyzing temporal scheduling complexities in an existing schedule and providing optimal possible time scheduling resolutions showing their reduction rate for efficient and intelligent knowledge management.

In recent years, the explicit formal specifications of terms in any given domain and relations among them have gradually moved from the realm of Artificial Intelligence (AI) laboratories to the desktops of domain experts. The representation of domain concepts in structural ways - definition of concepts (classes and properties) and mapping the relationships among the defined concepts, form the basis of Ontology [5]. From the computer science perspective, it is a formal naming and definition of the types, properties, and interrelationship of the entities that really or fundamentally exist for a particular domain of discourse [6] [7]. For example, in a lecture timetable domain, courses, timeslots, students, lecturers, venues and lectures are some of the concepts from which the timetabling application domain can be described. These concepts and their meanings together constitute ontology for timetable and can be used as common knowledge for communication among educational stakeholders and provides information for the development of a timetable information system.

Ontologies exist in diverse forms - lexicons, dictionaries, thesauri, and logical models described in languages such as first order logic (FOL). Lexicons provide standard lists of words

[*]Corresponding Author: Patience Usoro Usip, Computer Science Department, University of Uyo, Uyo, +2348060546140, patiencebassey@uniuyo.edu.ng, patceeng@gmail.com

(vocabulary) in a language with corresponding knowledge of how each word is used. Hence a lexicon can be seen as an index for mapping written form of a word with the information about that word [8]. Dictionaries can be organized to form hierarchies-taxonomies, meronomies, mereologies and merons, according to specific relations. Related terms, linguistics objects are added to any given collection of terms through thesauri. Ontologies provide standardization of the terms used to represent knowledge about a domain in any of these forms. They can as well support inference with FOL or its subset, by deriving new facts from a collection of facts and enforcement of consistency. It also allows for sharing common understanding of the structure of information among people or software agent; reuse of domain knowledge; making domain assumption explicit; separating domain knowledge from operational knowledge; and analyzing domain knowledge. These considerations are clearly useful for knowledge management, especially when large amounts of knowledge are being processed. Ontology is increasingly used in various fields such as Knowledge Engineering, Artificial Intelligence(AI) and computer science, in applications related to knowledge management (KM), natural language processing.

KM involves the acquisition, creation, use, representation, organization and advancement of knowledge in its many forms. As a requirement for effective KM, an understanding of how individuals, groups and organizations use knowledge is needed. One major aspect of an organization that require effective management of knowledge is timetable scheduling. This is because an organization's general timetable exhibit different levels of temporal scheduling complexities depending on the available resources. The threats posed by these complexities ranges from poor organization performance as seen in the domain of discourse (poor academic performance). Knowledge management is thereby required in attempt to reduce these complexities. Automation of timetabling process as seen in existing systems [9] [10] [11] does not resolve these complexities. This paper demonstrates the application of ontologies in timetable KM. It is aimed at analyzing any existing temporal schedule to discover all complexities and providing possible temporal schedules with reduced complexities (in time and space requirement) that tends towards optimality. This is achieved by utilizing Allen's temporal interval relations described in the following sections and evaluating the resulting optimal temporal schedule for proof and the need for adoption.

2. Formal Ontologies and Temporal Interval Relations

The time ontology has been considered by several authors with unique contributions to issues relating to temporal (time) representation and reasoning among others. Hobbs and Pan [12] considered temporal relations (TR) involving two subclasses of time— time instant and time interval. Allen's interval relations [13, 3] centers on various time intervals and possible reasoning involving co-operating concepts. As proposed by Allen [3], a framework for temporal reasoning, and all the relations proposed dealt with the directionality of time. In his proposal, intervals are the only temporal primitives in the temporal logic. Allen aimed at illustrating natural language sentences and to represent plans. To achieve the thirteen (13) basic relations between time intervals, with six of the intervals being the inverse of the other six: before, after, finishes, finished-by, overlaps, overlapped-by, starts, started- by, during, contains, meets, met-by and equal [3]. The basic interval temporal relations (Figure 1). These temporal relations depict and relate the actions and plans described in this paper.
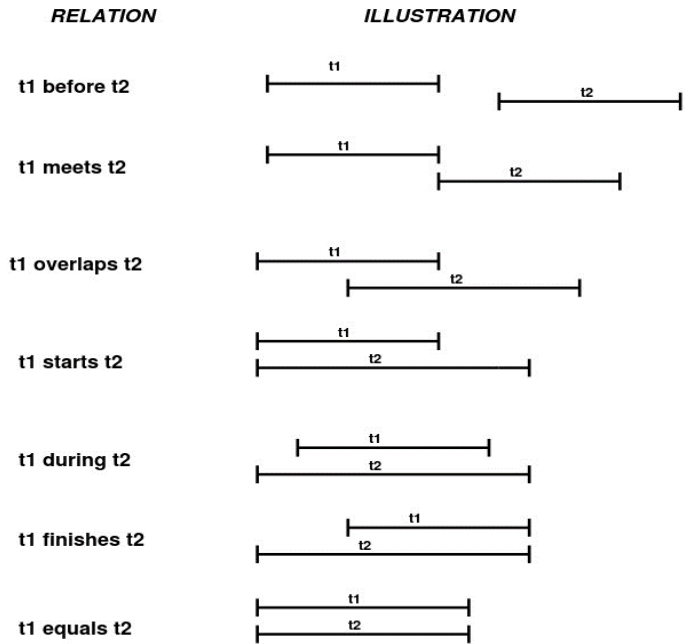

Figure 1: Allen's interval based temporal relations

This paper is targeted at formalizing the basic temporal ontology model for reasoning and inference towards an optimal time scheduling, after an analysis of an existing timetable to unravel all the complexities and evaluating feasible solutions towards achieving optimality. An illustration of how moment and point works on interval-based temporal logic is given in "moment and point in an interval based temporal logic". Moment is a non-decomposable period where the time is corresponding to instantaneous events while a point (a zero duration time) is where reasoning about the beginning and ending of event do arises. The axiomatization of the stated theory of time was done in terms of the simple relationship "meets" and subsumes the interval-based theory proposed in [3] thereby extending the theory to point-like time periods.

3. Timetabling Complexities

Timetable analysis reveals the major concepts - departments, timeslots, courses, students, venue and lecturers as objects responsible for complexities in timetable implementations. Three categories of courses exist in any typical higher institution; departmental, faculty-based and university-based courses. This paper considers the complexities arising from their inter-dependencies in scheduling the intra-departmental and inter-departmental (faculty-based and university-based) courses. Amongst other timetabling competency questions, are the following in the optimization of the general temporal scheduling ontology. They include:

i. Is the course offered by other departments, other than the host department?
ii. Are other courses for the same level in all cooperating departments allocated to the same time?
iii. Are students in higher levels in the cooperating departments also offering the course?
iv. Is the course lecturer teaching another course allocated at the same time?
v. Is the venue for course far apart from venue of the preceding or succeeding courses (courses before or after)? That is, are the respective venue for teaching the courses on different campuses? or are they within the same campus but widely separated?

Each of these concepts has peculiar attributes. These concepts relate in diverse ways and some of the relations exist in reverse form. The relations include *owns, has, can-be, offers, teaches, holds-in, lectures-in*, and *are-assigned-to*. In consideration of the fact that most resources of the university (such as lecturer, venue, and courses) are shared and in some other cases limited, departments compete for these resources. A university has a given number of programmes running on specified number of campuses. Campuses or lecture venues widely separated far apart from each other will exhibit a high level of complexity with numerous inter-dependencies. A sample model in fig. 1 with the departmental general timetable ontology for each department was described in [1] and shown in the resulting ontograf (Figure 2) from protégé. Almost all the components of the ontology are shared, making it needful for interoperability amongst relating departments. An attempt to link the ontologies for all the relating departments will make the resulting ontology cumbersome and more complicated largely because components of the ontology are not owned by a department, the number of inter-dependencies will generate some form of conflict of interest.
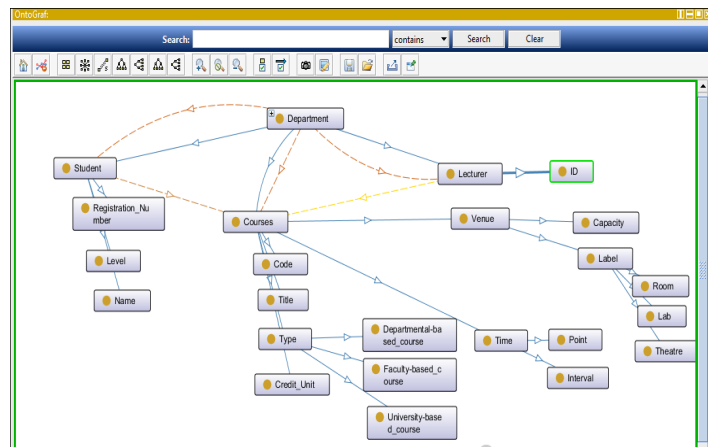

Figure 2: An Ontograf of a Departmental Timetable Ontology

Suppose course c_i is to be taught by lecturer L_l, scheduled to hold at venue v_1 at time t_i and another course c_j to be taught be the same lecturer L_l at same time t_i in another venue v_2, exposes some form of complexities. Again, suppose c_i and c_j are both scheduled to hold at v_1 at the same time t_i bring to bear another level of complexity that is predominant in the domain. That is a groups of students offering a particular course holding at time t_1 at campus U_1, and also offering another course c_j holding at time t_2 at campus U_2, where time t_1 meets time t_2. The meeting time does not give

allowance for the participating students to travel from campus U_1 to campus U_2 (or from venue vi to venue v2 with campus U_i). A general model of these complexities is shown in Figure 3.

A view of the various complexities based on the Allen's interval relations results in the following temporal complexity types: overlapping time (which also includes starts, ends and during relations) for courses taught by same lecturers and offered by same level of students. Meeting time complexity also results for courses offered by inter campus students. Other complexity types, not handled in this paper, include venue-clash and carrying capacity check.
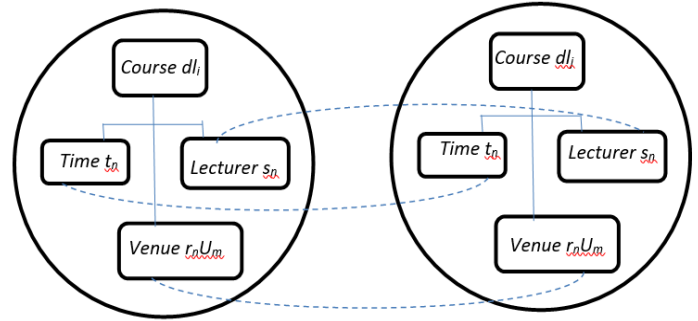

Figure 3: A course allocation complexity model

where d is the department's code, l is the level of students, t represents the timeslot, s is the lecturer, r the room number while U is the campus. $i,j,n,m>0$ are instances of the various objects

From the general course allocation complexity model in Figure 3, several levels of complexity are defined to include:

- Departmental Complexity (Regular)
- Departmental Complexity (Carryover)
- Faculty-Based (Inter-Departmental) Complexity (Regular)
- Faculty-Based (Inter-Departmental) Complexity (carryover)
- Inter-Faculty/Same Campus Complexities (Regular)
- Inter-Faculty/Same Campus Complexities (Carryover)
- Inter-Faculty/Inter-Campus Complexities

A generalized timetable ontology will expand the departmental timetable ontology is obtained from a mapped departmental timetable ontology as modeled in [1]. This require the mapping of domain resources ranging from, past results, course registration list, departmental brochure for details of all level courses and lecturers, campus location (or distance apart), etc.

Table 1 gives a justification of complexities in all the cases in consideration and the possible resolution based on interval temporal relations.

4. System Framework

Figure 4 gives the architecture of the proposed Ontology-Based Temporal Scheduling Framework showing the relevant components for the achievement of its set goal of producing an optimal time schedule.

Table 1: Complexity description by type and possible time scheduling resolution

Case number	Case description	Complexity type	Resolution
1.	Carryover /repeating students in the host department.	Overlapping- Time	If carryover students are offering course c_i at time t_1, then time t_2 for course c_j (time for their level course should not overlap).
2.	Carryover / repeating students in the serviced department.	Overlapping- Time	If carryover students are offering course c_i at time t_1, then time t_2 for course c_j (time for their level course should not overlap).
3.	Regular students (students at that level) in the services department.	Meeting- Time	If course c_i is scheduled at time t_1 in campus U_1, (for e.g 10-12) and course c_j is also scheduled at time t_2 in campus U_2, where the end-time of course1 c_i meets with the start-time of course c_j, the course should be rescheduled.
4.	Assigning the same lecturer to teach at different campuses, without giving sufficient time interval.	Meeting- Time	A lecturer assigned for course c_1 at time t_1 in campus U^1, and also assigned for course c_j at time t_2 in campus U_2, where the end-time of course c_i meets with the start-time of course c_j, should be rescheduled.
5.	students at same level and carryover/repeaters student in the services department.	Starts/ Finishes	If course c_i and course c_j are of different department, then course c_j should not start at the same time with course1 and finishes after course c_i
6.	Students in their regular year of studies and carryover student of different department, but the same faculty.	During	If course c_i and course c_j are of different department , d_i and d_j respectively, but of the same faculty, then course c_i should not hold during course c_j
7.	Students (both regular students and carryover student) at another campus, offering a course holding at a different campus.	Location Campus location (distance apart)	If course c_i is scheduled for time t_1 in campus U_1, and course c_j is also scheduled for time t_2 in campus U_2, then time t_1 should not meet t_2
8.	The class size and the carrying capacity of the lecture venue.	Space (carrying capacity)	If the number of students offering course c_i is greater than the venue assigned, then course c_i should be re-scheduled for a bigger venue.

Figure 4: Architecture of the Proposed Ontology-Based Time Scheduling System

The major components of the system as depicted in Figure 4, are described as follows:

(i) Universe of Discourse: the universe of discourse where the domain knowledge under consideration is obtained. It consists of all the domain concepts — courses, venues, universities, timeslots, lecturers, students, students' status, departments, faculties, levels, campuses, locations, timetable and other aspects of the timetabling domain

(ii) Fact Database: this includes the values and instances of the domain concepts, the resources, relations and a mapping of facts to concepts and resources. It also has the hard and soft constraints.

(iii) Time Ontology; this component stores the domain specific rule (owns, has, offers etc.) which will be considered while mapping the resources.

(iv) Allocation Reasoner: this is where reasoning and inferencing take place based on the Allen's interval-based temporal relations. The sets of rules arising from the competency questions and some already established constraints, are also part of this component. the Allocation Reasoner works in collaboration with the domain specific rules.

(v) Ontology Evaluation Approach: Criteria from the gold or golden ontology evaluation standard helps in assessing the set of feasibilities obtained from the Allocation Reasoner towards optimality of the timetable.

(vi) Optimal Time Schedule: This gives the resulting possible time schedule showing the resolutions made towards optimality.

To manage the complexities called for the optimized general time schedule with an allocation reasoner with interval-based temporal relations operating in-between the cooperating departmental timetable ontologies. Shared components of the optimized general timetable ontology. The interval-based temporal relations/rules serve as the instrument for managing the identified complexities in the timetabling system. With the interval-based temporal relation allocation of courses to venues, allocation of courses to timeslots (time duration), and assignment of lecturers to courses/venues are efficiently done.

4.1 Proposed Complexity Reduction Rules

From competency questions (i) to (v) identified in section 3.0, rules R1 to R12 addresses the complexities as well as adopting the Allen's interval relations in the proposed resolutions. It is assumed that all departments within the same faculties are domicile in the same geo-location:

R1: IF student of department d_i *offers* course c_i AND department d_i NOT *owns* course c_i THEN time t_1 for course c_i **NOT** *overlaps* time t_2 for course c_j in department d_2.

R2: IF course c_i *holds* at time t_1 and course c_j holds at time t_2 AND t_1 *equals* t_2 AND students offer c_i and c_j THEN t_1 and t_2 *overlaps*.
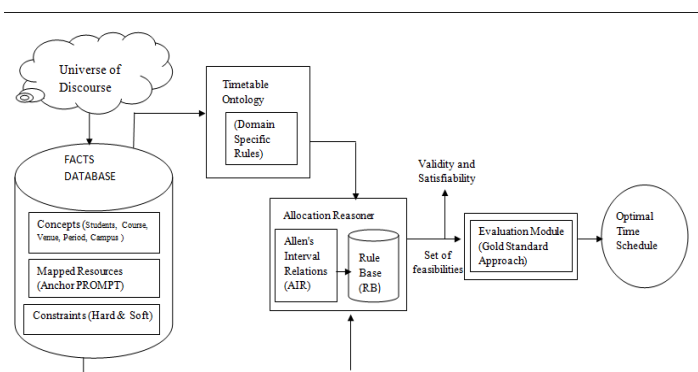
R3: IF course c_i *holds* at time t_1 and course c_j holds at time t_2 AND time t_1 is ***during*** time t_2 AND students offer course c_i and course c_j THEN t_1 and t_2 ***overlaps***.

R4: IF course c_i *holds* at time t_1 and course c_j holds at time t_2 AND t_1 ***starts*** with t_2 AND students offer c_i and c_j THEN t_1 and t_2 ***overlaps***.

R5: IF course c_i *holds* at t_1 and course c_j holds at time c_2 AND t_1 ***finishes*** at t_2 AND students offer c_i and c_j THEN t_1 and t_2 ***overlaps***.

R6: IF c_i and c_j *belongs* to the same department, d_1, THEN scheduled c_j to t_i AND c_j to t_j such that $t_i \neq t_j$ (**NOT** ***overlaps***).

R7: IF c_i and c_j are of different department, d_i and d_j respectively, but of the same faculty, THEN time for c_i should **NOT** ***overlaps*** with c_j or c_i **NOT** ***during*** c_j.

R8: IF t_i for c_i and t_2 for c_j ***meets***, THEN t_j for c_j will be changed and scheduled for another time (**NOT** ***meets***)

R9: IF c_i and c_j are courses of different faculty, THEN their time **NOT** ***meets*** and **NOT** ***overlaps***

R10: IF time for c_i and c_j are ***equals*** AND venue assigned for c_i is the same venue assigned for c_j, THEN (c_i OR c_j) should be rescheduled for different time.

R11: IF time for c_i and c_j are ***equals*** or ***meets***, AND the lecturer assigned for ci in campus U_1 is also assigned for c_j in U_2 within the time interval of one hour, THEN c_j should be re-scheduled for another time.

R12: IF c_i is scheduled for t_i in U_1 AND c_j scheduled for t_j in U_2 THEN t_i **NOT** ***meets*** t_j.

R13: IF carryover/repeating students are offering c_i scheduled for time t_i THEN t_j, time for their level course, c_j **NOT** ***overlaps*** t_i.

R14: IF c_i in campus U_1 and course c_j in U_2, THEN time t_i for c_i and t_j for c_j **NOT** ***meets*** or **NOT** ***overlaps.***

R15: IF number of students offering c_i is greater than the venue carrying capacity, THEN re-schedule c_i.

Formally expressing the above rules in FOL results in axioms A1 to A15. Recalling from the conceptualization of the domain concepts: **student, lecturer, time**, **venue** and **campus** with domain-specific relations: *owns, has, offers, holds* and *teaches* does not resolve time scheduling complexities when considering inter-departmental course allocation. This called for the adoption of the Allen's interval relations which includes temporal relations such as ***overlaps. meets, equals, during*** and their negations for a more explicit representation. Axioms A1 to A15 constitute the core of the ontology reasoner that results in the optimal time scheduling process.

A1: $\forall Student, Course1, Time1, Dept1 \exists Course2, Time2, Dept2.$
$offers(Student, Course1) \wedge owns(Dept1, Course1)$
$\wedge offers(Student, Course2) \wedge owns(Dept2, Course2)$
$\Rightarrow cooperates(Dept1, Host, Dept2, Serviced).$

A2: $\forall Student, Course1, Time1, Course2, Time2.$
$holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge equals(Time1, Time2) \wedge offers(Student, Course1) \wedge offers(Student, Course2) \Rightarrow overlaps(Time1, Time2).$

A3: $\forall Student, Course1, Time1, Course2, Time2.$
$holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge during(Time1, Time2) \wedge offers(Student, Course1) \wedge offers(Student, Course2) \Rightarrow overlaps(Time1, Time2).$

A4: $\forall Student, Course1, Time1, Course2, Time2.$
$holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge starts(Time1, Time2) \wedge offers(Student, Course1) \wedge offers(Student, Course2) \Rightarrow overlaps(Time1, Time2).$

A5: $\forall Student, Course1, Time1, Course2, Time2.$
$holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge finishes(Time1, Time2) \wedge offers(Student, Course1) \wedge offers(Student, Course2) \Rightarrow overlaps(Time1, Time2).$

A6: $\forall Course1, Time1, Dept1, Course2, Time2.$
$owns(Dept1, Course1) \wedge owns(Dept1, Course2) \wedge holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge equals(Time1, Time2) \wedge equals(Level1, Level2) \Rightarrow \neg overlaps(Time1, Time2).$

A7: $\forall Course1, Time1, Dept1, Fac1, Course2, Time2, Dept2. Fac2.$
$owns(Dept1, Fac1, Course1) \wedge owns(Dept2, Fac2, Course2) \Rightarrow \neg overlaps(Time1, Time2) \vee \neg during(holds(Course1, Course2)).$

A8: $\forall Course1, Time1, Course2, Time2, \exists t.$
$holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge finishes(Time1, t) \wedge starts(Time2, t) \Rightarrow meets(Time1, Time2).$

A9: $\forall Course1, Time1, Dept1, Fac1, Course2, Time2, Dept2. Fac2.$
$owns(Dept1, Fac1, Course1) \wedge owns(Dept2, Fac2, Course2) \wedge \neg equals(Fac1, Fac2) \Rightarrow \neg meets(Time1, Time2) \wedge \neg overlaps(Time1, Time2).$

A10: $\forall Course1, Time1, Venue1, Course2, Time2, Venue2.$
$holds(Course1, Time1) \wedge holds(Course2, Time2) \wedge equals(Venue1, Venue2) \Rightarrow \neg equals(Time1, Time2).$

A11: $\forall Course1, Time1, Let, Camp1, Course2, Time2, Camp2.$
$holds(Course1, Time1, Camp1) \wedge holds(Course2, Time2, Camp2) \wedge equals(Time1, Time2) \wedge equals(Camp1, Camp2) \wedge teaches(Let, Camp1) \wedge teaches(Let, Camp2) \Rightarrow \neg overlaps(Time1, Time2) \vee \neg meets(Time1, Time2)$

A12: $\forall Course1, Time1, Camp1, Course2, Time2, Camp2.$
$holds(Course1, Time1, Camp1) \wedge holds(Course2, Time2, Camp2) \wedge equals(Time1, Time2) \wedge equals(Camp1, Camp2) \Rightarrow \neg meets(Time1, Time2)$

A13: $\forall Student, Course1, Time1, Dept1, Course2, Time2.$
$offers(Student, Course1, Time1) \wedge offers(Student, Course2, Time2) \wedge owns(Dept1, Course1) \wedge equals(Time1, Time2) \wedge equals(Level1, Level2) \Rightarrow \neg overlaps(Time1, Time2).$

A14: $\forall Student, Course1, Time1, Dept1, Course2, Time2.$
$offers(Student, Course2, Course2) \wedge holds(Course1, Camp1) \wedge holds(Course2, Camp2) \Rightarrow \neg meets(Time1, Time2) \neg overlaps (Time1, Time2).$

A15: $\forall$*Student, Course1,Venue1, Time1, nos,* $\exists$ *Venue2, ccap.*

    offers(Student, Course1, Venue1) $\wedge$ *holds(Course1, Venue1, Time1)* $\wedge$ *(nos > ccap)* $\Rightarrow$ *holds(Course1, Venue2)*

4.2   *Weighting of Complexities*

The severity levels of complexities are denoted by 0%, 50% 70% and 100% respectively based on the sources of complexity (time, space and location).

1. *Overlapping Time*:

Overlapping time complexities includes five relations in Allen's Interval-based Temporal Relations: s*tarts, finishes, overlaps, during* and e*quals*. In this paper, these five temporal relations are classified into two levels of complexities namely complete and partial. Complete overlap occurs if two or more courses offered by a group of Students begins and ends at the same time (Equals Relation – See Table 2) or the Start and End times of these course is within the start and end time of c_j (During Relation – See Table 3), then a complete overlap of weight 100% has occurred. However, the During relation will not apply in this domain since the university system does not have 4-hour lecture period. For example, instances of *equals* and *during* relations are given in Tables 2 and 3 respectively.

Table 2:   equals Relation

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00–12:00 NOON
c_i			
c_j			

Table 3:   during Relation

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00– 12:00 NOON
		c_i	
	c_j		

A partial overlap is described using Allen's Interval-based Temporal Relations as follows; *overlaps*, *starts* and *finishes* shown in Tables 4, 5 and 6 respectively. These relations under partial overlap are assigned a complexity of 50%. Examples of partial overlaps are given in Tables 4, 5 and 6

Table 4:   overlaps Relation

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00– 12:00 NOON
c_i			
	c_i		

Table 5:   starts Relation

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00–12:00 NOON
c_i			
c_j			

Table 6:   finishes Relation

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00– 12:00 NOON
c_i			
	c_j		

In the resolved overlap, the desired Allen's interval-based temporal relation for the resolution of overlapping time complexity is *before* in Table 7 and Table 8. When course c_i is *before* course c_j a 0% complexity is recorded.

Table 7:   before Relation (A)

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00– 12:00 NOON
	c_i		
			c_i

Table 8:   before Relation (B)

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00– 12:00 NOON
	c_i		
			c_j

2. *Meeting Time Complexity*

If courses c_i and c_j hold in geographically dispersed Venues (R_nU_m, $R_{n+1}U_{m+1}$) and it takes a student time t_n to move from venue R_nU_m to $R_{n+1}U_{m+1}$, there exist a time complexity if course c_j starts immediately course c_i ends as shown in Table 9.

Table 9: Meeting Time Complexity (100%)

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00 – 12:00 NOON
	c_i		
			c_j

The 70% is for time to reach the venue and 100% is if there is additional time for the student to settle in the class before the actual lecture start time, an additional 30% weight is added. A 70% and a 100% resolution of the Meeting time complexity is as shown in Tables 10 and 11

Table 10: 70% Resolved Meeting Time Complexity

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00 – 12:00 NOON
	c_i		
			c_j

Table 11: 100% Resolved Meeting Time Complexity

8:00 – 9:00 AM	9:00 – 10:00 AM	10:00 – 11:00 AM	11:00 – 12:00 NOON
c_i			
			c_j

5.   Implementation Results and Discussions

The timetable showing schedules for two departments located at different campuses are used for the implementation of the proposed system.

These two departments offer common courses — CSC 211, STA 211, CSC 111. The natures of complexity are (meeting and overlapping times) clearly. Analysis of existing system shows the course under study, CSC 211 with Geo-informatics students in 300 level, who may be carrying over the course. It also handles the distance apart between the two campuses, being the venue for these two courses. Another case considers a complete overlap where CSC 211 and STA 211 from two different departments at two different venues, though in the same campus. Here, students of both departments are offering CSC 211. The analysis of these complexity levels with given assigned weights on the identified complexities is as shown in Table 12.

Table 12: Complexity weights of the existing time schedules

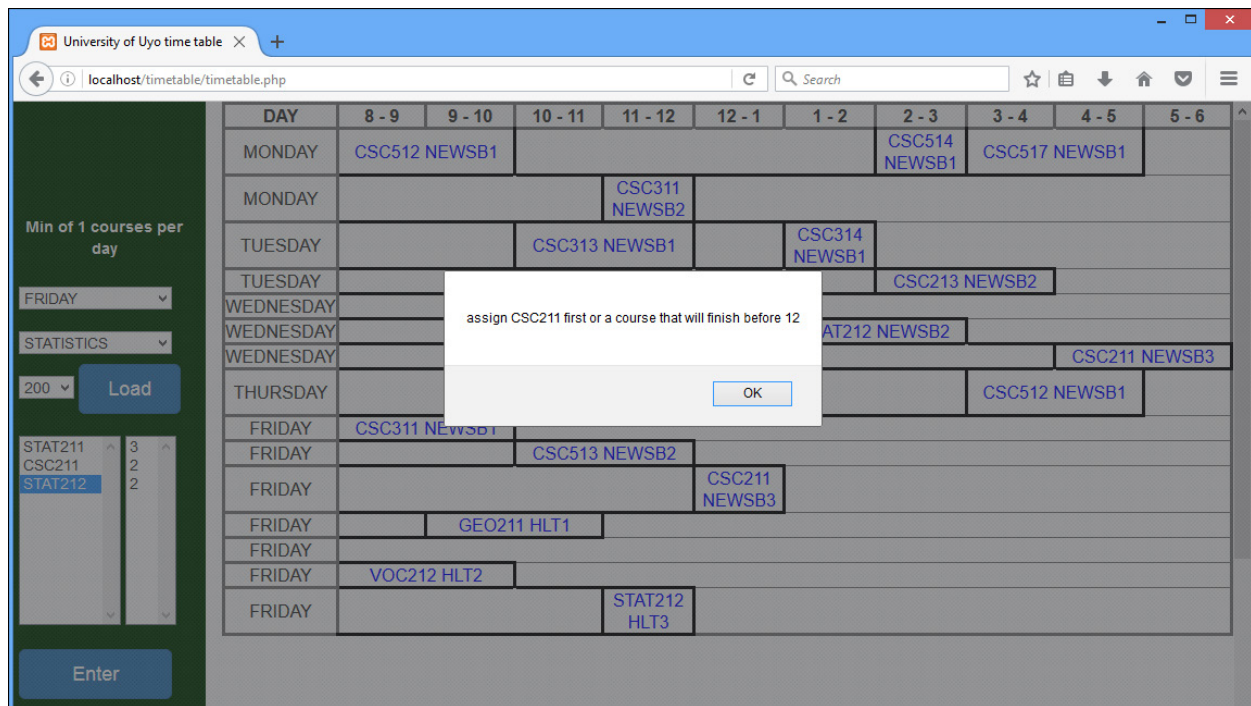| Case No	Complexity Type	Complexity Weight (%)
1	Meeting Time	100
2.	Overlapping Time (*equals*)	100

Figure. 5: Active Resolution of Meeting Time Complexity.

5.1 Resolutions with proposed Time Scheduling system

To resolve the time schedules, the complexity weights of the two cases should be 30% and 0% for meeting time and overlapping time respectively. The screen shots in Figure 5 gives the resolution of the meeting time complexity for the scheduled GEO 211 and CSC 211. The screen shot also shows a prompt to reschedule the STA 212 course as shown on the white pop-up window. That is the overlapping time complexity resolution, where the prompt calls for the relocation of the statistics (STA) course resulted. This is because CSC 211 is offered by many other departments, thereby having increased level of inter-dependency than the STA 212 course.

6. Conclusion

The proposed optimized time scheduling system analyzes any given existing time schedule and identifies the complexity types in existence. It provides the users the window to reschedule any time with identified complexity to obtain an optimized time schedule as shown in this paper. The rules in the reasoner are based on Allen's interval-based temporal relations as well the domain specific relations. Knowledge management as seen in the university timetabling complexities is possible with the mapping of the departmental timetable ontologies shown in the ontograf with all other required resources in the databases of fact. As a further work, the formal model that describes how the embedded semantics in the given rules are used for inferencing will be implemented on the ontology.

References

[1] P. U. Usip and M. M. Ntekop. "The use of ontologies as efficient and intelligent knowledge management tool", *In Proceedings of Future Technologies Conference. 6-7 December 2016, San Francisco*. pp 626-631. **DOI:** 10.1109/FTC.2016.7821671

[2] M. E. Saleh. Semantic-based query in relational database using ontology. Journal on Data and Knowledge Engineering. 2011.

[3] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26. 1983. (11): 832-843.

[4] P. U. Usip and E. Umoren. Towards a formal interval-based temporal ontology for an optimized timetable management. In Proceedings of Transition from Observation to Knowledge to Intelligence Conference (TOKI 2016), Lagos. pp. 293-304

[5] K. H. Shafa'amri and J. O. Atoum. A framework for improving the performance of ontology matching techniques in semantic web. International Journal of Advanced Computer Science and Applications, 2012. 3(1) 8-14.

[6] B. Coppin. Artificial Intelligence Illuminated. Jones and Bartlett Publishers, Inc., 2004. Sudbury

[7] D. Waltz. Artificial Intelligence: realizing the ultimate promises of computing, AI magazine. 1997) 18, 49–52

[8] I. M. Jurisica and J. Eric. Ontologies for knowledge management: an information systems perspective. Knowledge and Information Systems. 2004. 6, 380–401. DOI 10.1007/s10115-003-0135-4

[9] D. Abramson. Constructing school timetable using simulated annealing: sequential and parallel algorithm. 1991.

[10] E. K. Burke and S. Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*. 2002. 140. 266-280.

[11] A. O. Modupe, O. E. Olusayo and O. S. Olatunde. Development of a university lecture timetable using modified genetic algorithm approach. International Journal of Advance Research in Computer Science and Software Engineering. 2014. 4. 163-168.

[12] J. R Hobb and F. Pan. *An ontology of time for semantic* web. 2004. 3, 66-85.

[13] M. Hamalatha, V. Uma and G. Aghila. Time ontology with reference event based temporal relation (RETR). *International Journal of web and Semantic Technology* (IJWSSY). 2012. 3.