Formalisation of Transition Between Occurrents in A Domain Ontology

Ibiyinka T. Ayorinde *Member, IAENG* and Babatunde O. Akinkunmi

***Abstract*—Occurrents are dynamic entities in a Top-level ontology which consists of states, processes and events. In this study, only events defined as temporally bounded happenings, where if one or more participants in that event change, may strictly cause other events. Events may initiate or terminate states and may initiate or terminate processes. States (of the world) only affect causation in as much as they can allow events to cause other events or processes to perpetuate other processes. The causation relation between processes and events was formalised using first order logic. The study is applied to appointments in an organisational domain which gives a variant of the relation between processes and events in the literature. The formalisation will help improve knowledge modelling in organisations.**

Keywords: Appointment, Domain ontology, formalisation, Occurrents

I. INTRODUCTION

In computer science, ontology is a formal representation of knowledge by a set of concepts within a domain and the relationships between those concepts. Ontologies are used in Artificial Intelligence (AI), the Semantic Web, Systems Engineering, Software Engineering, Biomedical Informatics, Library Science, Enterprise Bookmarking and Information Architecture as a form of knowledge representation about the world or some part of it [1], [2]. A domain ontology is a formal and consensual dictionary of categories and properties of entities of a domain and the relationships that hold among them [3]. According to Guarino (1998), Top-level ontologies describe very general concepts like space, time, matter, object, event and action among others, which are independent of a particular problem or domain: it seems therefore reasonable, at least in theory, to have unified top-level ontologies for large communities of users. [4].

According to Galton (2006a), occurrents are dynamic entities in a Top-level ontology which consists of states, processes and events [5]. Occurrents occur in time and they are also called perdurants. Perdurants have temporal qualities whose qualia are temporal regions. A temporal region is an occurrent entity that is a part of time (of the whole of time) [6]. An occurrent is not wholly present at any time less than its entire duration. Rather, it has temporal parts, which may have different properties but the occurrent itself does not undergo change [7]. Occurrents are extended in time; they have temporal parts and thus they can be partitioned via partitions of the temporal dimension [8].

Since ontology is not just about objects of the world alone but also, is a world of constant change, ontological tools must be fashioned in such a way as to accommodate the fact of this change. The changes in the world include processes and events [8]. The study of processes and events helps to understand what is involved in ontological and semantic analysis, within the framework of formal ontology and logic. [9]

Manuscript received December 11, 2018; revised December 14, 2018.

I. T. Ayorinde is with the Department of Computer Science, University of Ibadan, Ibadan, Nigeria. (Phone: +234803-528-9814; e-mail: temiayorinde@ yahoo.com).

B. O. Akinkunmi is with the Department of Computer Science, University of Ibadan, Ibadan, Nigeria. (e-mail: ope34648@yahoo.com).

II. STRUCTURE OF THE DOMAIN ONTOLOGY

Fig. 1 shows different processes that can take place on a state entity (appointment) and these processes usually result into either state sequences or events. For instance, Retirement/Resignation/ Termination/ Dismissal processes on temporary/probationary/tenured appointments resolve into "LeftService". The formalisation of the axioms are discussed in section III.

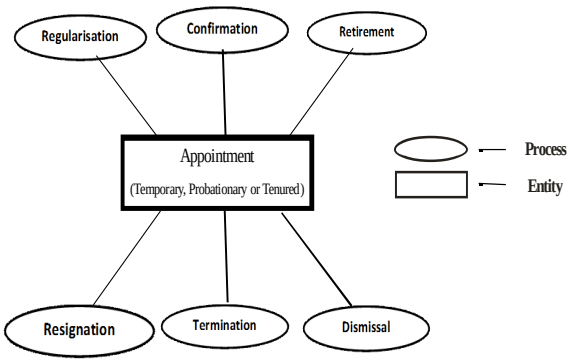

Fig. 1: Different Processes on an Appointment

A. Domain Sorts

The domain entities are described as follows:

(i). Domain of Persons: These are individuals in the domain. Variables can also be used to represent them. Examples are x, y and z among others.
(ii). Domain of Time Point, t, where t represents a calendar date.
(iii). Domain of Time Intervals (t_1, t_2) where (t_1, t_2) is a pair of time points.
(iv). Domain of Appointments
(v). Domain of Kind of Appointment = {Temporary, Probationary and Tenured}
(vi). Domain of Events = {Regularise, Confirm, Retire, Dismiss, Terminate, Resign}
(vii). Domain of Processes = { Resignation, Confirmation, Retirement, Dismissal, Termination, Regularisation}

Note that constants start with capital letters while variables start with small letters.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. Predicates and Their Signatures

In this study, the language uses the following predicates whose signatures are defined below:

(i). OccursP: Process x Time-interval → Boolean
OccursP is a relation between process and time interval and it returns a boolean expression. Example is: OccursP(p, (t_1, t_2)). This simply says that a process p occurs within a time interval (t_1, t_2).

(ii). Occurs: Event x Time-point → Boolean Occcurs is a relation between an event and a time point and it returns a boolean expression. Example: Occurs(e, t). This simply says that an event e occurs at time t. For instance, an appointment takes effect from 21 July, 2018.

(iii). StatusA : Appointment x Time-point x Kind of Appointment → Boolean
StatusA is a relation between an appointment, a time point and kind of appointment. It returns a boolean expression. Example: StatusA(a,t,Temporary). This example says that the status of an appointment at time t is Temporary.

(iv). Cause: Process x Event → Boolean
Cause is a relation between a process and an event. Example: Cause(p, e). Here, process p causes an event e.

(v). Commences: Appointment x Time-point → Boolean
Commences is a relation between an appointment and a time point. Example: Commences(a,t). From this example, an appointment commences at time t.

(vi). Confirm: Appointment x Time-point → Boolean
Confirm is a relation between an appointment and a time point. Example: Confirm(a,t_1). From this example, an appointment a is confirmed at time t_1.

(vii). Dismiss: Appointment x Time-point → Boolean
Dismiss is a relation between an appointment and a time point. Example: Dismiss(a,t_1). This simply says that an appointment was dismissed at time t_1.

(viii). Extend: Appointment x Time interval → Boolean
Extend is a relation between an appointment and a time interval. Example: Extend(a, (t+365, t_1). Here, an appointment a is extended for a certain period of time interval.

(ix). GrossMisconduct: Person x Time point → Boolean
GrossMisconduct is a relation between a person and time point. Example: GrossMisconduct(Holder(a),t_1).

(x). Ill-Health: Person x Time-point → Boolean
Ill-Health is a relation between a person and a time point. Example: Ill-Health(Holder(a),t_1)). This expression simply says that the holder of an appointment was reported to have ill health at time t_1.

(xi). Lapse: Appointment x Time-point → Boolean
Lapse is a relation between an appointment and a time point. Example: Lapse(a, t+365). Here, an appointment lapses after a specified time.

(xii). LeftService: Person x Time-point → Boolean
LeftService is a relation between a person and a time point. Examples are: LeftService(Holder(a),t_1) and LeftService(x, t′).

(xiii). Misconduct: Person x Time point → Boolean
Misconduct is a relation between a person and time point. Example: Misconduct(Holder(a),t_1).

(xiv). Quash: Appointment x Time-point → Boolean
Quash is a relation between appointment and time point. Example: Quash(a,t_1). This simply says that an appointment quashes at time t_1.

(xv). Regularise: Appointment x Time-point → Boolean
Regularise is a relation between an appointment a time point. Example: Regularise(a,t_1). From this example, an appointment is regularised at time t_1.

(xvi). Resign: Appointment x Time-point → Boolean
Resign is a relation between an appointment and a time point. Example: Resign (a,t_1). From this example, an appointment is resigned at time t_1.

(xvii). Resume: Person x Time-point → Boolean
Resume is a relation between a person and a time point. Example: Resume(Holder(a),t_2)).

(xviii). Retire: Appointment x Time-point → Boolean
Retire is a relation between an appointment and a time point. Example: Retire(a,t_1). From this example, an appointment is retired at time t_1.

C. Reification of Terms

Reification refers to the use of terms in first order logic (FOL) to express concepts normally expressed using predicates, operators or even complete propositions [8]. For instance, the formula Run(x) which says that object x runs can be reified by ∃e. Run(x, e) ∧ Occurs(e, t). In this case, the running event has been made into a thing (or a term in the language). Semantically, to reify a concept is to accord it full ontological status, so that it becomes an entity that can be ascribed properties to and, in principle, quantify over [10], [11].

Events, Processes and Appointments were reified in this formalisation. States were not reified because they can be defined by their time limits. Also, states are permanent and reifying them may not be necessary.

In any organisation, appointment is a complex entity. It is an abstract concept, that is, it is not a physical concept. It is made a thing by reifying it, that is, we assume it physically exists and hence has its own structure (or attributes) like: owner of the appointment, time of the appointment, and kind of appointment among others. Consider the expression:

∃a. Commences(a, 01-10-2018) ∧ Owner(a, John) ∧ KindA(a, Temporary).

From the above axiom, the following deductions can be made:

(i). There exists an appointment which commences on October 1, 2018.
(ii). The owner of the appointment was John and
(iii). The kind of appointment given was Temporary.

In this study, events and processes have complex structures but reifying them helps to represent them appropriately. An event can be quantified over as shown:

∃e. Occurs(e, t) ∧ Agent(e, Abiola)

The above axiom simply says that there exists an event e such that Abiola was the agent through whom the event occurred.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

III. TRANSITION BETWEEN DIFFERENT LEVELS

Fig. 2 shows transition from processes which resolved into state/event sequences. It will be noted that during these transitions, the initiation of processes lead to the occurrence of certain events and these events eventually stopped whenever the processes were terminated.

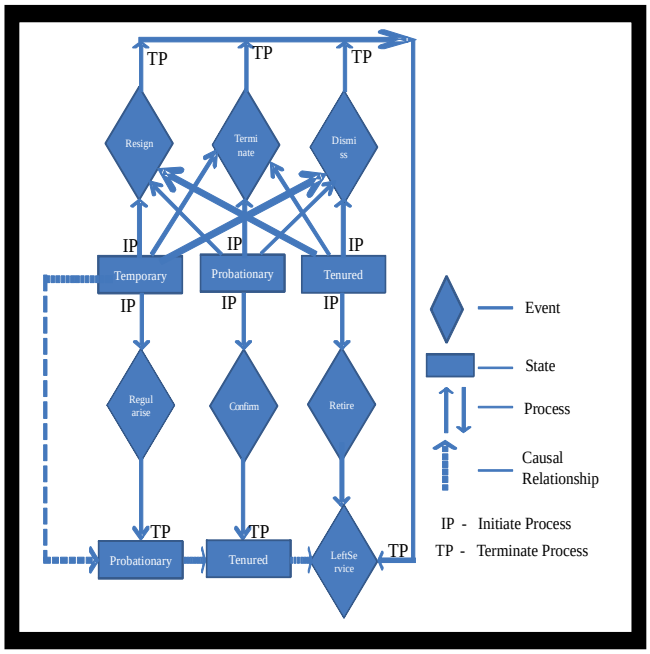

Fig. 2: Processes on Appointments resolved into State/Event

A. *From Processes to States*

From figure 2, the following deductions can be made:

(i). Regularisation process on a temporary appointment resolved into a probationary appointment(Axiom 1).

Axiom 1

Temporary → Probationary

A temporary appointment is regularised after one year of taking up such an appointment to make it probationary.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Temporary) \wedge \exists t_1.\ t \leq t_1 \leq t+365 \wedge Regularise(a,t_1) \rightarrow StatusA(a, t_1, Probationary) \wedge \forall t'.\ t \leq t' \leq t_1.\ StatusA(a, t', Temporary)$

(ii). Confirmation process on a probationary appointment resolved into a tenured appointment (Axiom 2).

Axiom 2:

Probationary → Tenured

A probationary appointment is confirmed after three years of taking up such an appointment to make it tenured.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Probationary) \wedge \exists t_1.\ t \leq t_1 \leq 3*365 \wedge Confirm(a,t_1) \rightarrow StatusA(a, t_1, Tenured) \wedge \forall t'.\ t \leq t' \leq t_1.\ StatusA(a, t', Probationary)$

(iii). Retirement process on tenured appointment resolved into "LeftService" (Axiom 3)

Axiom 3:

Tenured → Retirement

A tenured appointment is retired after thirty five years of taking up such an appointment or if the holder of the appointment is sixty years old.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Tenured) \wedge \exists t_1,t''.\ ((t \leq 35yrs \leq t_1) \vee ((t'' = 60yrs \wedge Age(Holder(a), t'))) \wedge Retire(a,t_1) \rightarrow LeftService(Holder(a),t_1) \wedge Quash(a,t_1) \wedge \forall t'.\ t \leq t' \leq t_1.\ StatusA(a, t', Tenured)$

(iv). Resignation process on temporary/probationary/tenured appointment resolved into "LeftService" (Axioms 4-6).

Axiom 4:

Temporary → Resignation

A temporary appointment is said to have terminated if the holder of the appointment resigns within one year of taking up the appointment.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Temporary) \wedge \exists t_1.\ t \leq t_1 \leq t+365 \wedge Resign(a,t_1) \rightarrow LeftService(Holder(a),t_1) \wedge Quash(a,t_1) \wedge \forall t'.\ t \leq t' \leq t_1.\ StatusA(a, t', Temporary)$

Axiom 5:

Probationary→ Resignation

A probationary appointment is said to have terminated if the holder of the appointment resigns within three years of taking up the appointment.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Probationary) \wedge \exists t_1.\ t \leq t_1 \leq 3*365 \wedge Resign(e,t_1) \rightarrow LeftService(Holder(a),t_1) \wedge Quash(a,t_1) \wedge \forall t'.\ t \leq t' \leq t_1.\ StatusA(a, t', Probationary)$

Axiom 6:

Tenured → Resignation

A tenured appointment is said to have terminated if the holder of the appointment resigns within thirty five years of taking up the appointment.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Tenured) \wedge \exists t_1.\ t \leq t_1 \leq 35yrs \wedge Resign(a,t_1) \rightarrow LeftService(Holder(a),t_1) \wedge Quash(a,t_1) \wedge \forall t'.\ t \leq t' \leq t_1.\ StatusA(a, t', Tenured)$

(v).Termination process on temporary/probationary/tenured appointment resolved into "LeftService" (Axioms 7-9)

Axiom 7:

Temporary → Termination

A temporary appointment can be terminated (within one year of taking up the appointment) due to the holder's misconduct or illhealth.

$\forall a,t.\ Commences(a,t) \wedge StatusA(a,t,Temporary) \wedge \exists t_1.\ t \leq t_1 \leq t+365 \wedge (Misconduct(Holder(a),t_1) \vee$

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Ill-Health(Holder(a),t_1)) ∧ Terminate(a,t_1) → LeftService(Holder(a),t_1) ∧ Quash(a,t_1) ∧ ∀t′. $t \le t' \le t_1$. StatusA(a, t′, Temporary)

Axiom 8:
Probationary → Termination
A probationary appointment can be terminated (within three years of taking up the appointment) due to the holder's misconduct or illhealth.

∀a,t. Commences(a,t) ∧ StatusA(a,t,Probationary) ∧
∃t_1. $t \le t_1 \le 3*365$ ∧ (Misconduct(Holder(a),t_1) ∨
Ill-Health(Holder(a),t_1)) ∧ Terminate(a,t_1) →
LeftService(Holder(a),t_1) ∧ Quash(a,t_1) ∧
∀t′. $t \le t' \le t_1$. StatusA(a, t′, Probationary)

Axiom 9:
Tenured → Termination
A tenured appointment can be terminated (within thirty five years of taking up the appointment) due to the holder's misconduct or illhealth.

∀a,t. Commences(a,t) ∧ StatusA(a,t,Tenured) ∧
∃t_1. $t \le t_1 \le 35$yrs ∧ (Misconduct(Holder(a),t_1) ∨
Ill-Health(Holder(a),t_1)) ∧ Terminate(a,t_1) →
LeftService(Holder(a),t_1) ∧ Quash(a,t_1) ∧
∀t′. $t \le t' \le t_1$. StatusA(a, t′, Tenured)

(vi). Dismissal process on temporary/probationary/tenured appointment resolved into "LeftService" (Axioms 10 – 12)

Axiom 10:
A temporary appointment can be dismissed (within one year of taking up the appointment) due to the holder's gross misconduct.
Temporary → Dismissal

∀a,t. Commences(a,t) ∧ StatusA(a,t,Temporary) ∧
∃t_1. $t \le t_1 \le t+365$ ∧ GrossMisconduct(Holder(a),t_1) ∧ Dismiss(a,t_1) →
LeftService(Holder(a),t_1) ∧ Quash(a,t_1) ∧
∀t′. $t \le t' \le t_1$. StatusA(a, t′, Temporary)

Axiom 11:
Probationay → Dismissal
A probationary appointment can be dismissed (within three years of taking up the appointment) due to the holder's gross misconduct.

∀a,t. Commences(a,t) ∧ StatusA(a,t,Probationary) ∧
∃t_1. $t \le t_1 \le 3*365$ ∧ (GrossMisconduct(Holder(e),t_1) ∧ Dismiss(a,t_1) →
LeftService(Holder(a),t_1) ∧ Quash(a,t_1) ∧
∀t′. $t \le t' \le t_1$. StatusA(a, t′, Probationary)

Axiom 12:
Tenured → Dismissal
A tenured appointment can be dismissed (within thirty five years of taking up the appointment) due to the holder's gross misconduct.

∀a,t. Commences(a,t) ∧ StatusA(a,t,Tenured) ∧
∃t_1. $t \le t_1 \le 35$yrs ∧ (GrossMisconduct(Holder(a),t_1) ∧ Dismiss(a,t_1) →
LeftService(Holder(a),t_1) ∧ Quash(a,t_1) ∧
∀t′. $t \le t' \le t_1$. StatusA(a, t′, Tenured)

B. From Events to States

Events are dependent on processes in the following ways:
(i). A durative event is "made of" processes
(ii). A durative event may be an instantiation of a complex routine, composed of a number of distinct process chunks representing different phases
(iii). A punctual event is usually the onset or cessation of a process

Processes of Regularisation (Axiom 1):
(a). Regularisation commences
(b). Ensures present status is temporary
(**c). Regularise (Event)**
(d). Time is within one year
(e). Ensures new status is probationary.

Process of Resignation – Axioms 4 – 6
(a). Resignation commences
(b). Ensures present status
(**c). Resigns at a time (Event)**
(d). Left Service.

C. From States to Processes/Events

States are the unchanging part of the domain. When an event causes a process to be initiated on an initial state, say, "Temporary", the process terminates with a final state "Probationary".

Temporary	➔ Regularisation	➔ Probationary
Probationary	➔ Confirmation	➔ Tenured
Tenured	➔ Resignation	➔ Leftservice
Tenured	➔ Retirement	➔ Leftservice
Temporary	➔ Resignation	➔ Leftservice
Probationary	➔ Resignation	➔ Leftservice

IV. CAUSAL AND CAUSAL-LIKE RELATIONSHIPS

The following deductions can be made:
(i). Only events (defined as temporally bounded "happenings" where one or more participants in that event change) may strictly cause other events.
(ii). Events may initiate or terminate states and may initiate or terminate processes.
(iii). Processes can perpetuate other processes
(iv). States only affect causation in as much as they can allow events to cause other events or processes to perpetuate other processes.

A. Event and State:
(i). The event "Confirm" Initiates the state "Tenured"
(ii). The event "Confirm" Terminates the state "Probationary"
(iii). The state "Probationary" allows an event (which is the commencement of a confirmation) to cause another event (which is the actual confirmation and still ensures present status is probationary until a process terminates the event). Hence, we can say that the process of confirmation maintains an initial state

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(Probationary) and it also perpetuates another process (like ensuring the present status is probationary, makes sure time is within three years or with an extension of six months and confirms the appointment) until it is terminated (where it ensures that the new status of the appointment is Tenured).

B. Event and Process:

(i). The event "Resign" initiates the process of Resignation
(ii). The event "Resign" terminates the process of being in service.

C. State and Process:

The process of regularisation maintains the state "appointment".

D. Change in Granularity Levels

A process might be regarded as a dynamic state i.e. a state of change, involving continuous change in some attribute of one or more of its participants. Both states and processes can be regarded as present at individual moments of time, in contrast to events, which normally inhabit extended intervals. But unlike states, processes can in themselves be causally efficacious, as the confirmation example shows: the commencement of the confirmation process gives rise to ensuring the present status is probationary and the overall period of service is within three and a half years if extension is granted and three years if not granted.

The causal and causal-like dependencies amongst states, processes and events in this domain revealed that what appears at one granularity level as a process may, when described at a finer granularity, be seen as a sequence of events thereby changing their ontological character. This was shown in the axiom: $\forall a,t.\ \text{Commences}(a,t) \wedge \text{Status}(a,t,\text{Tenured}) \wedge \exists t_1.\ t \leq t_1 \leq 35\text{yrs} \wedge \text{Resign}(a,t_1) \rightarrow \text{LeftService}(\text{Holder}(a),t_1) \wedge \text{Quash}(a,t_1) \wedge \forall t'.\ t \leq t' \leq t_1.\ \text{Status}(a, t', \text{Tenured})$. The event "Resign" initiates the process of Resignation and terminates the process of being in service while the "Resignation process" on a tenured appointment resolved into a state sequence ("LeftService").

V. CONCLUSION

From the formalisation, it was established that the event of an individual being offered an appointment occurs at time t whereas, the conversion process, say from temporary to probationary or from probationary to permanent spanned through a period of time. Hence, the research work has been able to distinguish between the nature of processes and events in this domain. Events were instantaneous while processes took place over intervals. The causal and causal-like dependencies amongst states, processes and events in this domain revealed that what appeared at one granularity level as a process was, when described at a finer granularity, seen as a sequence of events thereby changing their ontological character.

Also, the ability to distinguish between the nature of processes and events in this domain makes this study fits into the occurrents classification in the Artificial intelligence (AI) field. Hence, organisations can incorporate the ideas in this formalisation in order to make their ontology a dynamic one and for effective knowledge modelling.

REFERENCES

[1]. D. Man, "Ontologies in Computer Science. Didactica Mathematica", Vol. 31(2013), No 1, pp. 43–46, 2013.
[2]. T. Gruber, "A translation approach to portable ontologies", Knowledge Acquisition, 5, 1993, 199–220.
[3]. S. Jean, G. Pierra, and Y. Ait-Ameur, "Domain Ontologies: A Database-Oriented Analysis", Proc. of Web Information Systems and Technologies (WEBIST'2006), Setubal, Portugal, 11-13 April 2006.
[4]. N. Guarino, "Formal Ontology and Information Systems", Proceedings of FOIS'98, Trento, Italy, June 6-8, 1998. IOS Press, Amsterdam, pp. 3-15.
[5]. A. Galton, (Power Point Presentation) "On What Goes On: The Ontology of Processes and Events" University of Exeter, UK, 2006a.
[6]. A. Galton, "The Treatment of Time in Upper Ontologies", Formal Ontology in Information Systems. S. Borgo et al. (Eds.) 2018. IOS Press. DOI:10.3233/978-1-61499- 910-2-33.
[7]. J. F. Allen, "Towards a General Theory of Action and Time" Artificial Intelligence 23: 123-154, 1984
[8]. A. Galton, "On What Goes On: The ontology of Processes and Events". Proceedings of the Fourth International Conference of Formal Ontology in Information Systems (FOIS) 2006b, Baltimore, USA, 9th - 11th Nov 2006, pages 4-11.
[9]. L. Obrst, "Advanced Ontology Topics: Events, Roles, Artifacts", STIDS 2013 Tutorial November 12, 2013. Information Semantics Innovative Information Engineering & Biometrics Center for Connected Government MITRE.
[10]. A. Galton, "Reified Temporal Theories and How To Unreify Them", *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (IJCAI'91), Sydney, Australia, 24th - 30th Aug 1991, pages 1177-1182.
[11]. B. O. Akinkunmi and A. O. Osofisan, "Temporal Properties of Repetitive Entities", *Artificial Intelligence Review* 22*:* 291 – 338, 2004.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)