ISSN No:-2456-2165

Goal Dependency Architecture using Object-Role Modeling in Role-Relationship of Object-Oriented Databases

Emecheta N. C.
Dept. of Computer Science
University of Port Harcourt Choba, Port Harcourt, Nigeria

Abstract:- **Organizations create wide use of information systems to take care of planning and decision making as well as controlling to effect good advantages. Organization is also a complex object that integrates different ideas such as strategy, processes, people, technology and information. This paper presents the vital concepts for modelling an organizational role-relationship using Object-Role Modelling for Goal Dependency Architecture in object oriented database. Object-Role Modelling (ORM) is used to make the design processes easier by using natural language and natural drawings that can be illustrated with examples and also inspecting the material in terms of simple or basic facts. Application area or universe of discourse selected in the paper is Conference management system. This is selected due to the multi-tasking involved during paper reviews on a distributed environment. The result of the system shows that a complex system can be developed via object-role model which simplifies a system and make it easy for object-oriented processes to communicate well with an object-oriented data.**

Keywords:- Role-relationship, Goal dependency, Architecture, Object-Role Modeling, Conference management system.

I. INTRODUCTION

Object-Oriented database (OODB) is database that denotes data in custom of objects and classes. In object-oriented language, object is a real world entity while a class is collection of objects. Object-oriented database use the vital principles of object-oriented programming (OOP). Object database stores complex data directly, without mapping to relational rows and columns. This makes it suitable for application dealing with complex data. Objects have many-to-many association which gain access to data by using pointers. Object-Role modelling (ORM) is a technique for modelling and querying information systems at the theoretical level and as well mapping between theoretical and logical levels. ORM comes in various flavours with Natural language Information Analysis Method (NIAM) inclusive. Building on the object-role model Nijssen's information analysis method NIAM (Halpin, 1995) and it's descendants (Halpin & Proper, 1995) provide a deeper account of the linguistic role of roles. They employ so called fact types which are the equivalents of linguistic statements associating properties which expresses the relationship of objects as the sole data structure. Each fact type involves a number of roles which correspond to the places of a predicate.

Organization is represented as a complex object; it deals with conflicting ideas such as people, business process, value chains and information system as well as technology. Expressing the fact about organization demonstrates a challenging task. It requires numerous ideas to be represented in a clear integrated way, not as a set of distinct and independent elements.

For an organization to transform, it must be self-aware. This means that the information on the organizational concepts is widely shared and agreed on. This gives room for minimization of mis-match between organization's actual state of affairs and state as observed by diverse stakeholders.

Detecting the architecture of innovativeness should be considered as an essential step for an organization that renders it vital to be ready to act rather than react: and then to be able to understand whether its element are aligned. In this paper, the term architecture stands for the fundamental arrangement of the components within any kind of socio-technical system, as well as their relationships to each other and the environment, and the design rules for developing and structuring the system (IEEE, 2000). The components are illustrated in form of model while reducing irrelevant and redundant features. Design rules specify the development and structuring of model. This specifies the types of components and their relationships and consistency conditions for the use of components and their relationships.

II. ORGANIZATIONAL SCOPE AND CONTRIBUTIONS

An organization is a complex man made system that comprises a formal group of people, which share one or more goals (Scott, 1997). A system is a collection of unified elements within an integrated whole. The system idea is often used to describe a set of objects which interact, and for which a model can frequently be fabricated.

A dependency is an association stating that one thing uses the facts and services of a different thing, but not necessarily the reverse. Dependency association can be used to show that one thing is using another. Record of dependencies between classes is used to show that one class uses operations from another class or it uses variable arguments typed by the other class. Dependencies often

show required interfaces of a class. They do not model structural relationships.

Architecture is designed bearing in mind of its expected usage. It is about specifying how the things on the enterprise operates and interact having its means specified. It is to allow on one hand describing organization, business processes, business information systems, and on the other assessing the alignment between these concepts.

A. Role

A role is an entity's observed behavior inside the parameters of a certain collaborative setting. Therefore, when an entity works with a group of other entities in the context of an activity, a role represents the externally observable features of that entity. A stereotyped relationship connects an entity to zero or more role classes. Each role in the context of particular collaborations defined by the role model represents a subset of its extrinsic or external attributes. The purpose of roles is to distinguish between the various issues that come up when distinct entities work together to complete a task. A play relationship can be used to link a role to several different things. When a role is bound to an entity, a particular instance of that entity can express the behavior specified by the role. It also implies that the role's characteristics and execution strategy will be included in the entity's feature set. A role is a type as well, and it can be categorized based on its characteristics and operations. In order to become a normal class, it can be generalized and aggregated. The most basic definition of a role is a designated position within a relationship. Codd, 1970 noted that if two or more places of a relationship in his relational model were declared to be of the same type, then the name of this type would not suffice to distinguish these places. In such a case role names should serve to identify the places in question.

The most fundamental account of roles as named places is presumably that by Falkenberg,(1976). In his object-role model (ORM) objects and roles are the sole primitives from which object types, associations and association types are derived. ORM's data model which allows nested associations is very similar to the so-called feature structures heavily employed in the unification-based branches of computational linguistic and knowledge representation K. H. Blasius et al, 1989, Carpenter, 1992.

B. Conference

A conference is a gathering where people discuss a certain subject. A conference can be an academic gathering where scholars formally present their findings and engage in seminars and other activities.

Web-based software that assists the organization of conferences, particularly scientific conferences, is known as a conference management system. It supports the operations of the program chair, conference planners, writers, and reviewers. Every year, academics and researchers in their fields of interest must attend at least one conference. Many stakeholders participate in various conference tasks at such a meeting. They consist of the chair of the program committee, the reviewers on the program committee, the general chair, the publicity chair, the authors, etc. A mechanism must be in place for a conference organization to be successful. The stages of organizing a conference include calling for papers, accepting submissions, reviewing submissions, discussing submission revisions, notifying authors, and many more. Academic conferences are becoming more and more common as a result of the availability of technological technology that affects every aspect of our reality. An huge increase in the quantity of articles submitted coincides with this. The size of the program committee must be greatly increased in order to handle the volume of papers and maintain manageable reviewing workloads. It is therefore judged impractical to schedule a face-to-face program committee meeting in order to examine and discuss paper submissions. In light of the aforementioned information, it is essential to create an online conference management system that makes conference planning easier.

C. Survey Problems(s)

Before a conference management system can be created, there are a number of requirements that must be satisfied. Modeling can be further pursued by breaking down a goal into smaller goals and by using potential alternatives to accomplish a goal. Alternatives are defined by several contributions and represented by OR-decomposition. At this point, soft goals can also be used to reflect non-functional Diagram requirements. Choosing one option over another results in the accomplishment of different soft goals. In this manner, it is possible to evaluate various options and decide which is best. Goal dependency diagrams may be generated dynamically. Every actor in the model has the ability to open (Close) their goal diagrams, which are represented by balloons attached to the appropriate actors. As a result, the tool will be able to assist the analyst in selecting the components for analysis.

ISSN No:-2456-2165

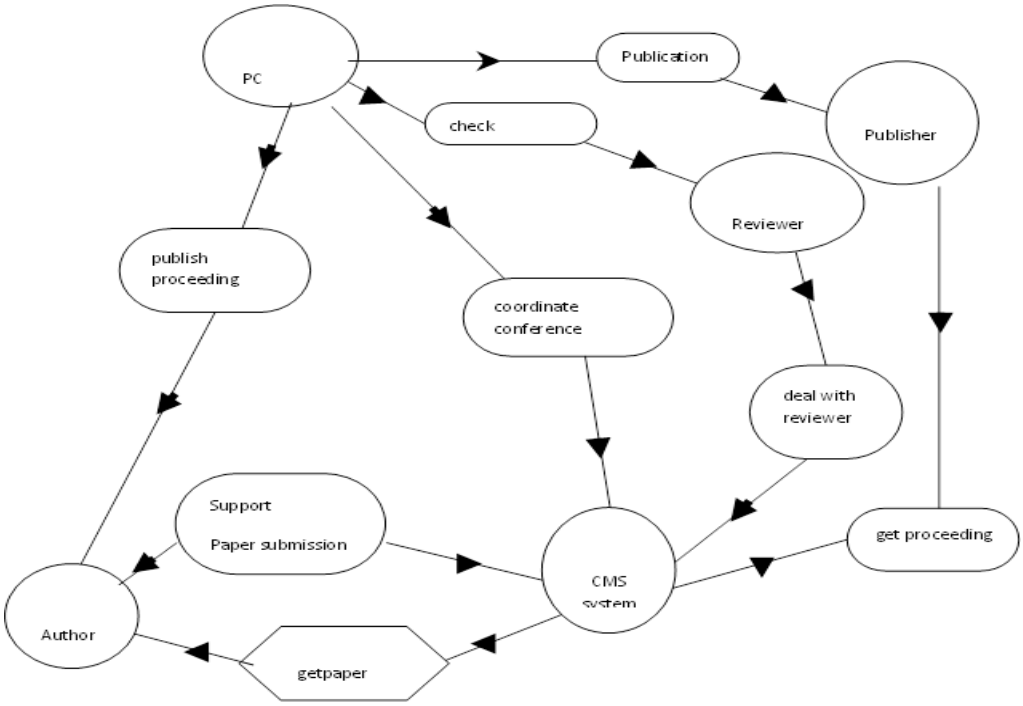

Fig. 1: Early Requirements of CMS: Goal Dependency Design

III. THE SYSTEM ARCHITECTURE DESIGN

The overall structure of the system is made up of all of its subsystems and the relationships between them.

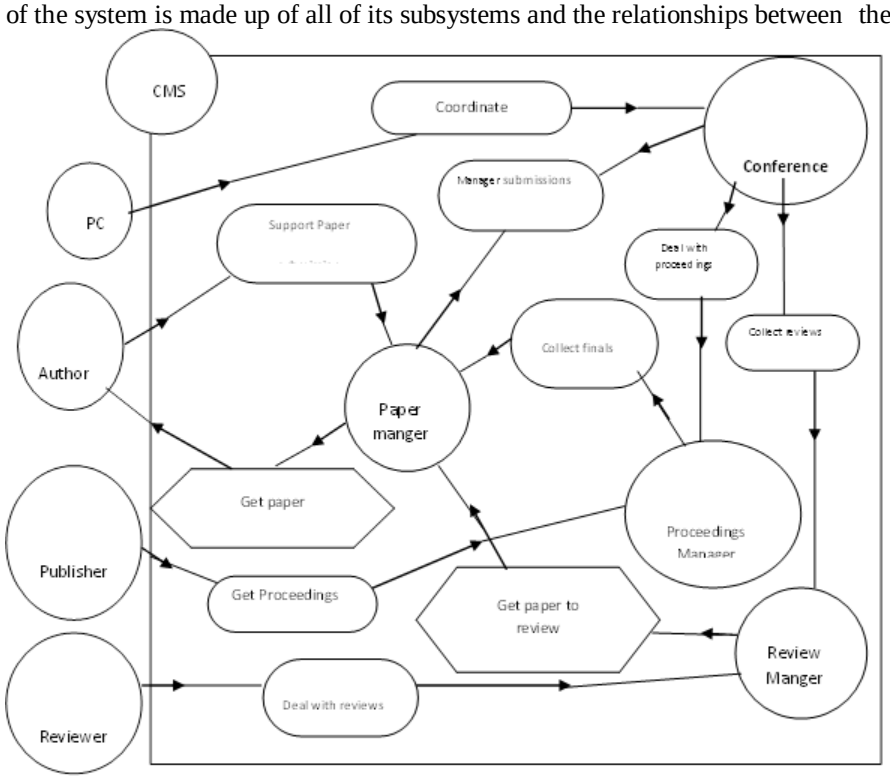

Fig. 2: The database architecture of the conference management system

Multiple tables in MySQL are used to hold data, and each table is connected to the others by a primary key field. For instance, we have a table of information on academics that provides specialized information about academics, and a table of information on great industrials or organizations called industrials. There is a common field (identification number) in both tables. By entering the identifying number, a third table in a relational database is created using the information from all of the other tables. The graphical representation for designing academic, industrial, or organizational conferences is displayed in Figure 2. The design clearly demonstrates how the various parts interact with the system's actors or users.

ISSN No:-2456-2165

Through the support paper submission and the Get_paper action, the author communicates with the paper management. Get_paper_to_review action and Deal_with reviews are two ways that reviewers communicate with review managers. On the other side, Publisher uses the Get_Proceedings process to communicate with the Proceedings Manager. The PC communicates with everyone via a coordinated method. Subsystems are agents that can operate independently and communicate with others through message passing, according to the multi-agent system concept.

The engineer will improve the system actor by incorporating sub-actors, who are in charge of really carrying out the system's primary aims, in order to construct the architectural design. A portion of the goal dependence model for the paper Manager and progressing manager, two of the sub-actors, is shown in Figure 3.

The conference system application used in the design of the proposed system for organization or academics conference online portal utilizes the frame work for software design.

The model clearly shows the dependency between the conference manager and conference publisher within the framework of the content management system. The dependency will definitely aid the process of designing the object-role model components required in the implementation of the system and the relationship between the components as documented in the design.

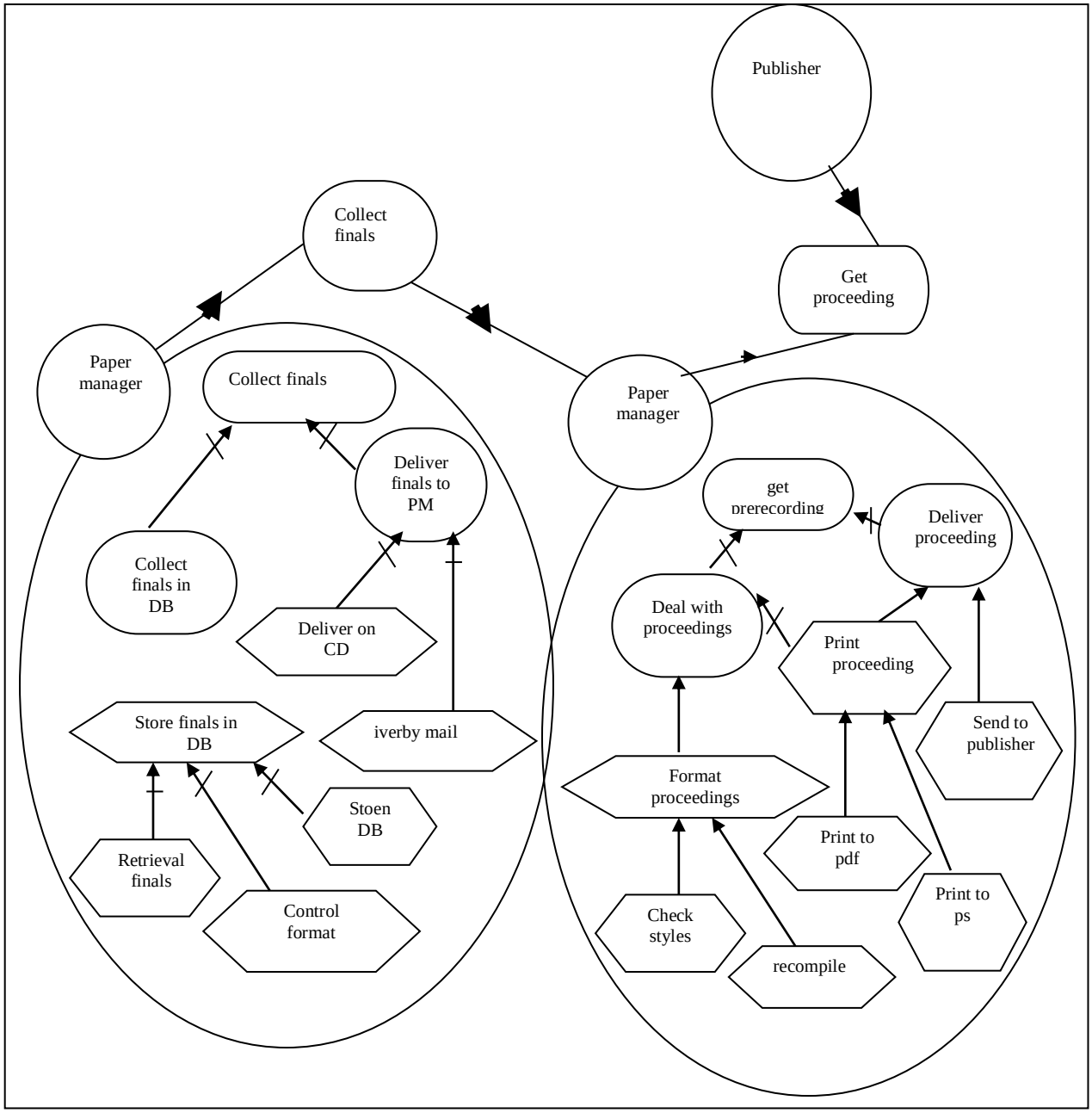

Fig. 3: The Combine goal dependency architecture of the conference management system.

ISSN No:-2456-2165

A. Object Role Model Design

A system model can be created conceptually using the Object Role Model (ORM) method, which also maps between conceptual and logical (e.g. relational) levels. Instead of recasting the application in terms of implementation data structures, the program is represented in terms that are easily understood by users. ORM includes a formal, textual specification language for both models and queries, as well as a formal, graphical modeling language (Terry, 2012).

Object Role Modeling got its name because it views the application world as a set of objects (entities or values) that plays roles (parts in relationships). It is sometimes called fact-based modeling because ORM verbalizes the relevant data as elementary facts. These facts cannot be split into smaller facts without losing information.

Simple, value, composite, and nested are the four fundamental types of data objects that object role modeling acknowledges. A simple object is one in which real world instances are designated -- uniquely identified -- by a single data element; i.e., a single data element comprises the primary key (Stanley, 2012). If real-world persons are to be labeled in the database by an identification number, Person_id, rather than by their names, Figure 4.11 illustrates how they would be represented in an object role model. Person is a simple object in this context.

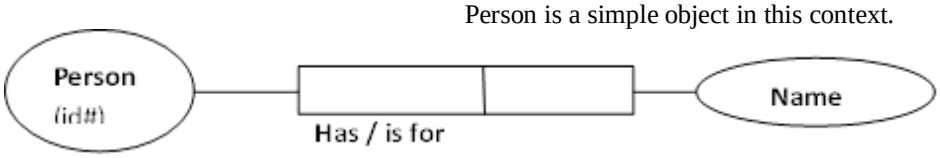

Fig. 4: Simple and Value Object

A value object, which represents a straightforward scalar property, is comparable to a name (label), number, or date. Name is a value object in Figure 4. An object-person is distinguishable from his name although there is a relationship, according to the object-oriented viewpoint. Composites are created by combining value and simple objects with a key. A nested object is one whose existence and identity are derived from the connection between two other objects. To "objectify" the relationship, a "circle" (a rectangle with rounded corners) is drawn around it. Then, an item that is nested can interact with other objects.

B. Analysis of UoD -Conference Management System

A conference is a gathering of people where they can present and talk about their work. The topic may be intellectual or relate to other spheres of life. Academic conferences provide as a forum for researchers to discuss their findings. Work is typically presented in the form of a brief, succinct presentation lasting between 10 and 30 minutes, usually with time for discussion. The work may be collected in writing form as academic papers that are assessed by a group of chosen scholars from a wide geographic area and then published as conference proceedings.

A web-based program called a conference management system aids in the planning of academic conferences. It supports the relevant tasks of the program chair(s), conference organizers, writers, and reviewers. A conference management system can be regarded as a domain-specific content management system. Similar systems are used today by editors of various journals (Barrett, 2000).

C. ORM of UoD Design-Conference Management system Design

In this essay, we have referred to the conference management system as the universe of discourse (UoD). In this section, we'll use ORM to map between conceptual and logical (for example, relational) levels as we describe the conference management system from a conceptual level. We will also provide a formal and graphical representation of the CMS system together with the textual specifications for models and queries. The CMS has many entities each relating to other entities within the system in a form that they cannot be separated without losing their collective perspective. In the system all conceptual factors viewed individually are elementary but they make a compound meaning when viewed as a whole. The only way the presentation as a whole can be realistically comprehended is if the system's object role model is used to link and relate the various entities. A role is a part played by an object in a connection and is shown as a box related to its object type in Figure 5. The Reviewer (a person object) in the straightforward illustration evaluates the Conference Papers (a document object, either physical or digital). The conceptual dot represents a required role constraint.

Fig. 5: Illustration of Role between two CMS objects

ISSN No:-2456-2165

The Reviewer plays the role of reviewing and the Conference paper plays the role of being reviewed. The keyword reviews appears on the part of the box connected to the Reviewer object indicating which object is playing the specified role.

IV. THE OBJECT ROLE MODEL CONCEPTUAL SCHEMA DESIGN

Figure 6 uses an object role model to describe a conceptual design for the conference management system. Without revealing the specific data components of the objects, this design makes it simple to comprehend the flow of activity and the interactions between objects.

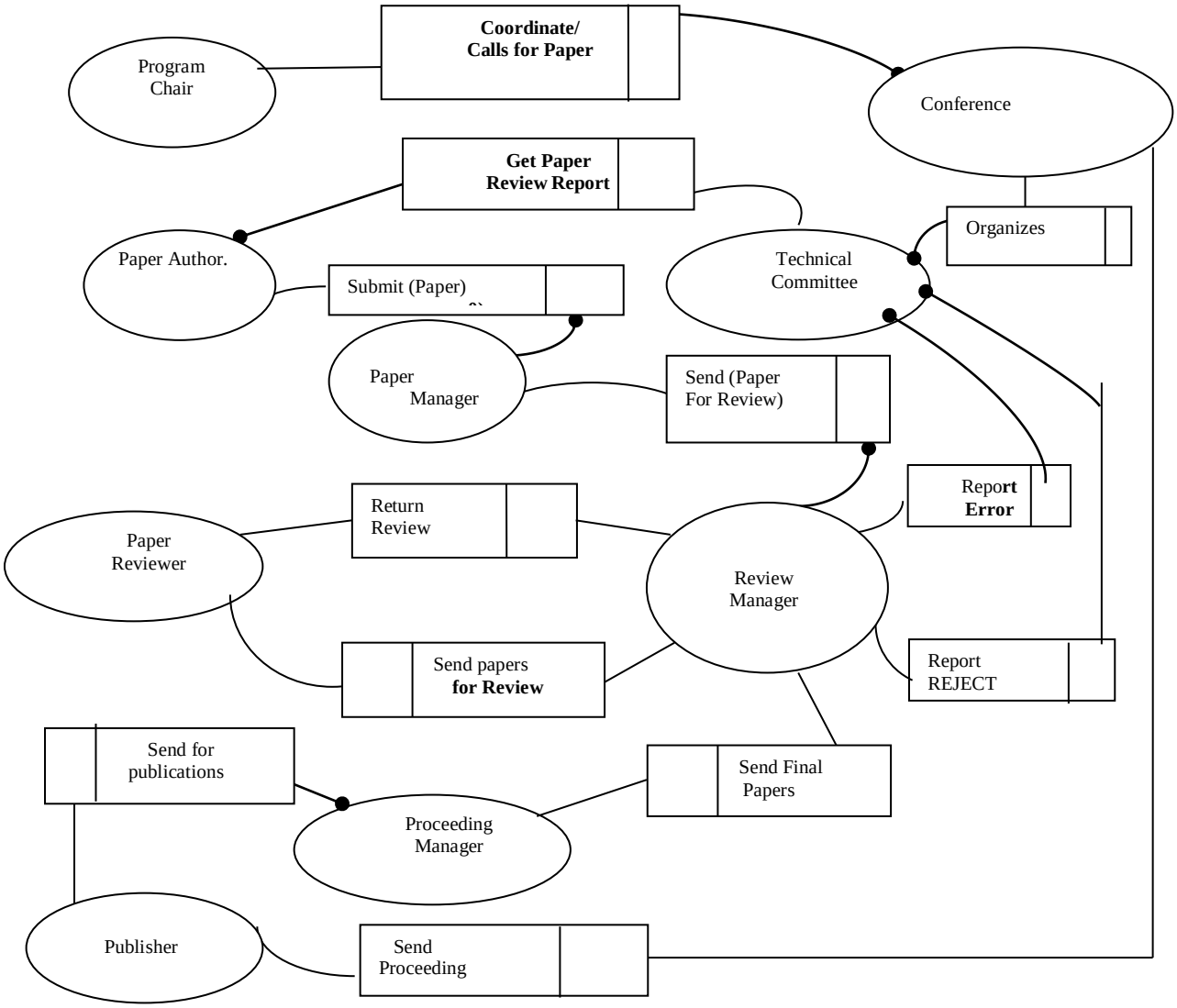

Fig. 6: Conceptual Design of the Conference Management System using ORM

In other to explain in some detail the design developed in Figure 6 above, we will partition the design into four major sections and provide a more detailed description of the design so that when the system is examined as a whole it will become clearer.

A. Section I: Coordination of Conference and Call for Paper

When we examine the diagram in Coordination of conference and Call for Paper section of the Conceptual Conference Management System Design (CCMSD) in Figure 7, it is very clear that Program Chair Coordinate and Calls for Paper for the Conference. The Program Chair plays the role of Coordinating and the Conference plays the role of being coordinated. Similarly the Technical Committee Organizes the Conference and the Conference is organized by the Technical Committee. These interactions of objects have situations where each object is clearly assigned a role in the conference management system.

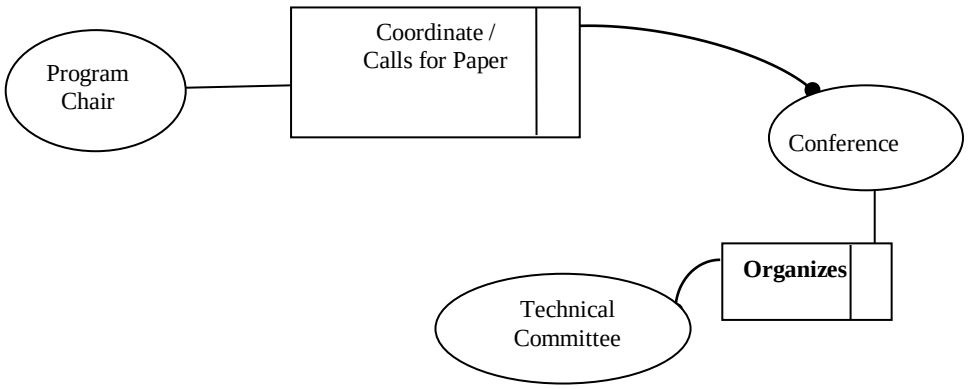

Fig. 7: Coordination and Call for Paper section of the CCMSD

This roles form the operational components required in the developmental and implementation stages of the system development. Since the system need the operation to be able to clearly define what each class does in the code developed for the system. It is also from the classes that the objects are extracted during the execution of the system. The three objects Program Chair, Conference, and technical committee play the roles in the call for paper section of the design.

B. Section II: Paper Management Activities Section of the CCMSD

The diagram in Figure 8 clearly shows the objects that are components of the paper management activities of the Conference management system. These objects include the paper Author, the paper manager and the technical committee. Other objects include the paper reviewer and the review manager. These objects interact in a specific manner aimed at handling paper using the led down procedure. The Author object triggers the inter-communication by responding to call-for-paper and submitting a paper to the Paper Manager. The Paper Manager sends the paper to the Review Manager for allocation to the various paper reviewers.

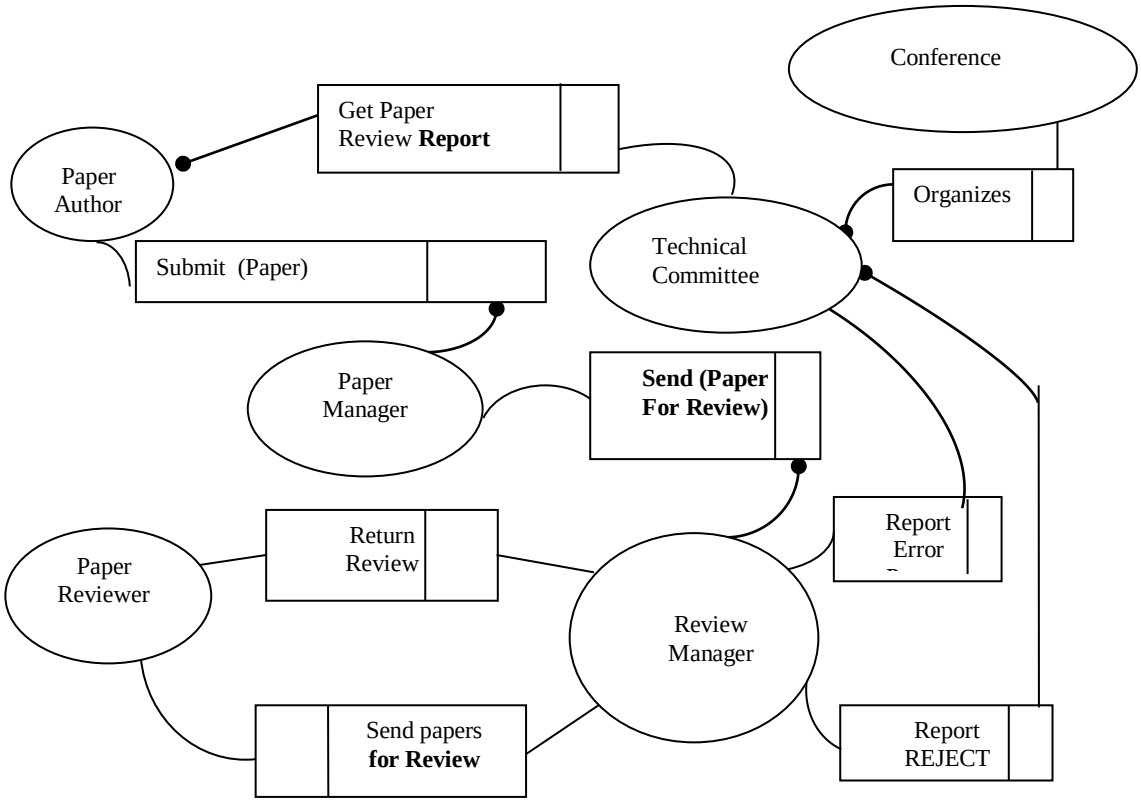

Fig. 8: Paper Management Activities section of the CCMSD

The Review Manager Send spapers for review to various Paper Reviewers and monitors the review process to make sure that the papers are reviewed at the appropriate time and that the Reviewers allocated papers are really experts in the area where they are saddled with the responsibility of reviews. The number of reviewer per paper is also determined by the review manager. Once the papers are reviewed they are returned to the Review Manager for the appropriate action. The Review manager will Report Error in paper and send to technical committee for onward transmission to the Authors for correction or report Rejection to the Author via the technical committee.

C. Section III: Paper publishing section of the CCMSD

The section on paper publishing still involves the Paper review manager who sends accepted papers to the Proceeding manager. The Proceeding manager in turn sends the papers in well-arranged proceeding format to the Publisher.

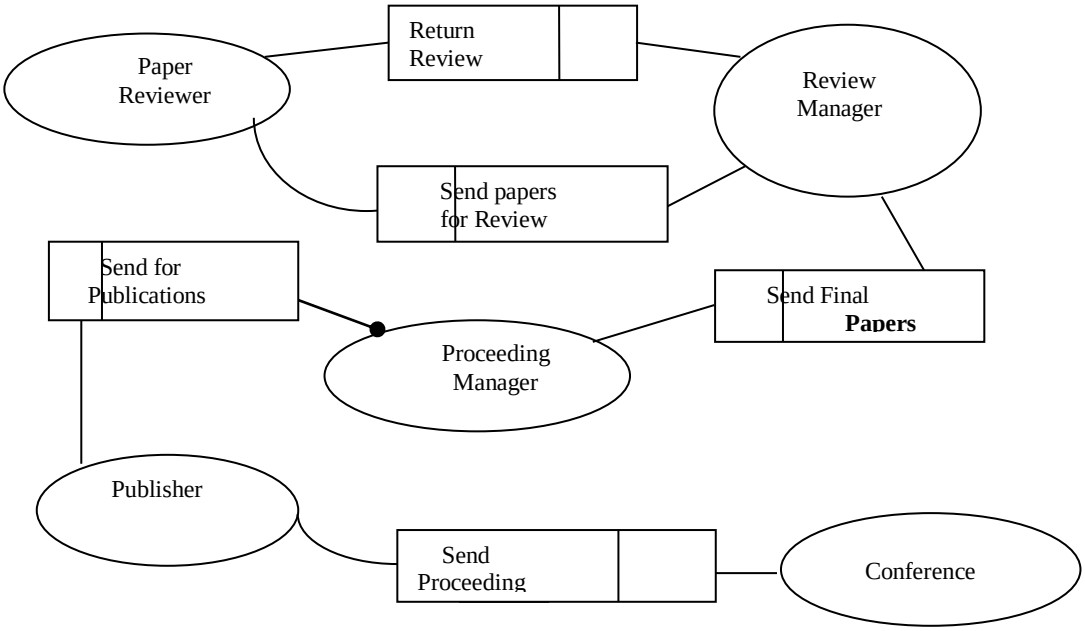

Fig. 9: Paper publishing section of the CCMSD

Once the activity reaches the publisher, the proceeding will be prepared and published. The last phase involves sending of the proceeding to the conference.

V. SUMMARY

In this paper the ability of object databases to communicate with object-oriented applications class-to-class, object-to-object and programming construct to construct using Goal dependency architecture in role-relationship are desired. The research looked at how the features could be incorporated into modern system development. This concept was tested in the research using a conference management system as a use case with application objects and all the user objects within the application usage environment.

VI. CONCLUSION

In this paper, a goal dependency architecture for the development of object database systems using Conference Management System (CMS) as a use case was developed. This model will serve as a guide for database management vendors and researchers in development of real life object database systems for use in the businesses and organization. We also designed object-database architecture for the CMS using Object-Role Modeling. Object-Role modeling is tailored towards developing object based representation of database entities.

It has been proven that it is easier for third parties to understand and represent entities attributes and their interactive behaviors when compared to entity relationship models (ER model). ER models in design are less effective at creating, transforming, or evolving a design. ER diagrams are further removed from natural language, cannot be populated with fact instances, require complex design choices about attributes, lack the expressibility and simplicity of a role-based notation for constraints, hide information about the semantic domains that glue the model together, and lack adequate support for formal transformations (Terry, 2012). This is why we have used the Object Role Models in the design though it seems to be relatively new to academic designs.

The research work was finally implemented using the CMS design produced with object data and associated database. This project has made some contributions which include:

- Object Role models of the system was equally developed which is modern tool for representing object database entity relationships in an easy to understand model.

VII. RECOMMENDATION

We also recommend the research in this work to academics that have the need of simplifying their development process as well as implementation of systems that need complex data. The conference management system developed in the work can also benefit academic conferences and the development process can also be used in the development of more complex systems using AMP technology and other facilities used in this research. The approach created via this research can help improve how efficiently and effectively academic institutions and other organizations manage conferences based on paper submissions. We advise academic conference planners to consider this study, as well as professional organizations like Computer Society of Nigeria (NCS) and others that host conferences on a variety of topics for practical application, so that our society can benefit from its inherent advantages.

REFERENCES

[1.] Stanley D. B. (2012) A Primer on Object Role Modeling, University of California Press, Berkeley.
[2.] Terry Halpin (2012) Object-Role Modeling (ORM/NIAM), Microsoft Corporation, USA reproduced by permission from Handbook on Architectures of Information Systems, eds P. Bernus, K. Mertins & G. Schmidt, Springer-Verlag, Berlin, 1998, www.springer.de/cgi-bin/search_book.pl?isbn=3- 540-64453-9.
[3.] Codd E. F. (1970) A relational model of data for large shared data bank, Communications of the ACM 13 (6), 377-387.
[4.] Falkenbeg E. (1976) Concepts for modeling information, in: G.M.Nijssen (Ed.), Proceeding of the IFIP Conference on Modeling in DataBase Management System, North-Holland, Amsterdam, pp. 95-109.
[5.] Blasius K. H., Hedtstuck U., Rollinger C. R.(Eds.) (1989). Sorts and types in artificial intelligence, Lecture Notes in Artificial Intelligence, Springer, Berlin, 418.
[6.] Carpenter B. (1992). The Logic of Typed Feature Structures: With Applications to Unification Grammars Logic, Program and Constraint Resolution, Cambridge University Press, Cambridge.
[7.] Halpin T. A. (1995). Conceptual Schema and Relational Database Design, Prentice-Hall, Sidney
[8.] Halpin T. A., Proper H. A. (1995). Subtyping and polymorphism in object-role modeling, Data & knowledge Engineering 15, 251 – 281.
[9.] IEEE (2000). IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.
[10.] Scott R. (1997). Organizations: Rational, Natural, and Open Systems, New Jersey: Prentice Hall.