

# NECESSITY OF MONITORING VITAL BIM INFORMATION FOR OPTIMIZED DESIGN PROCESS MANAGEMENT

MOHAMED MOHSEN, SAMIR ALBITAR, MOHAMED ESSAM & ALAA RADWAN  
BIM Management Department, Engineering Consultants Group (ECGSA), Qatar

## ABSTRACT

The broad range of BIM processes and tools implementation has resulted in the generation of a huge amount of complex data in assets management life-cycle that includes the asset main information, authorities associated information to assets, and a number of organizational information related to buildings design. In the BIM process management, the virtual model's verification and validation are becoming too complex to be managed and controlled by normal procedures, which leads to the development of non-traditional strategies for reviewing the BIM database completeness and correctness by the automation of collecting and arranging these BIM Data to generate useful and simple information which helps in achieving the targeted outcomes. Monitoring BIM vital information for optimized design processes management is becoming more necessary and introduce the question of how big the information is related to the BIM process and projects development. The segregation of information according to process needs, real-time availability, and the accessibility of information is crucial for successful implementation and also helps in the analysis of information for BIM process time and wastes optimization. Decision-making based on reference analysed information is clearly affected and good results are noticed. This paper introduces types of BIM design process information and its segregation according to function, and how this data is controlled by an in-house developed Electronic Data Management System (EDMS) on Common Data Environment CDE. It also shows cases of how BIM information is reviewed, analysed, and corrected by automated solutions which help to enhance decision-making on different management levels.

*Keywords: BIM management, BIM automation, design information management, BIM vital information, BIM information analysis, design processes optimization, enhanced decision making, automated process monitoring.*

## 1 INTRODUCTION

BIM adoption has tremendously increased across the current decade in many countries [1], [2]. Numerous numbers of BIM models have been produced around the world for different projects accompanied by huge amount of information for different purposes [2]. This information can be classified and sorted with reference to the project stage, phase and use. During the project lifecycle many information exchanges issues related to quality and interoperability may appear. Manual monitoring, reviewing and analysing of this amount of information become more sophisticated, completeness and accuracy become the major issue of handing over models for each BIM use. An automated solution can facilitate the aggregation, extraction of important information, reviewing the correctness and completeness of information and helps with decision making on different management levels. The automation process can be classified into four phases: aggregation, extraction, reviewing and analysing, and optimization for better decision making (Fig. 1).

## 2 TYPES OF INFORMATION ASSOCIATED WITH BIM AND BIM SUPPORTING SYSTEMS

Each project stage has its own new inputs of related information and accumulative information from the previous stages. There are different types of information associated



with asset lifecycle which go beyond BIM model geometrical content from design to demolish, BIM Information is related to engineering analysis, design and construction progress, project schedule, stakeholders contacts list, design and construction budget, collaboration and communication protocol, information exchange matrix, hardware and software specifications, fabrication and manufacture details, disaster planning and management, space management, asset management, maintenance schedule and others. This information has to be managed on different levels by various support systems and software [3], which can be classified as follows: BIM modelling and data entry systems, BIM management Systems, BIM decision support systems and BIM executive systems (Fig. 2). Each system has many supporting software which help to get BIM use.

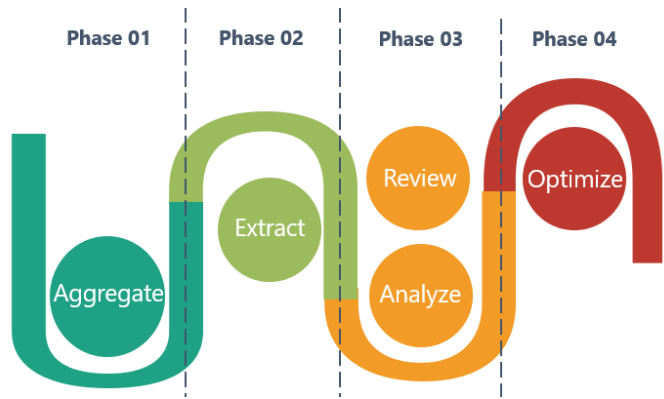


Figure 1: Phases of information automation for decision making.

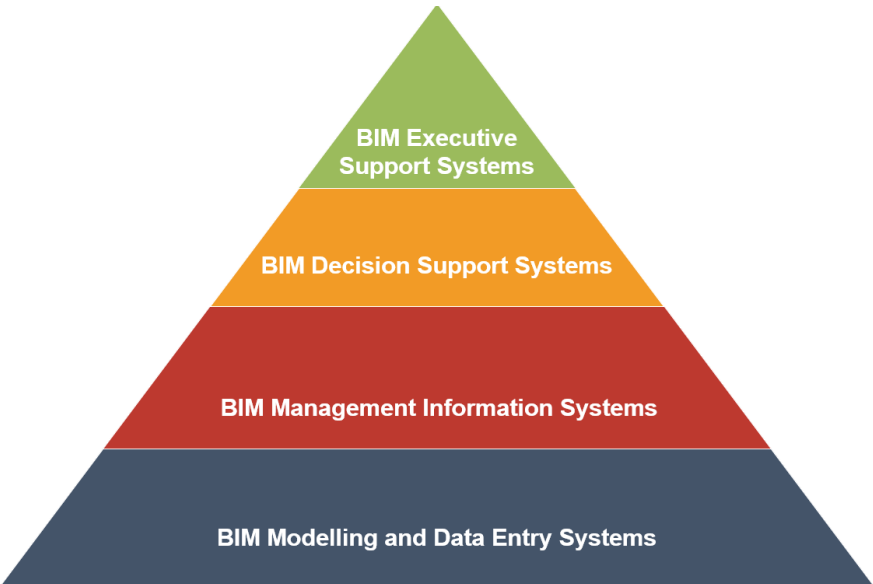


Figure 2: Levels of BIM supporting systems.

### 3 AGGREGATION OF BIM INFORMATION

While implementing BIM for any project, Common Data Environment usage is mandatory and gives a superb opportunity to control and manage the information of assets' volumes [4]. CDEs software are various and providing different solutions based on project's needs. ECG has introduced different way of managing projects information based on windows environment using PEER solution which is managed by inhouse EDMS software complying with international BIM standards, regulation and projects satisfaction [5]. The components of the solution are: File Servers, Data server, PEER Software, ECG-EDMS. (Fig. 3 shows an overall diagram for ECG CDE and EDMS.)

The shared environment is the only source of valid design information shared between design team. Changes or modifications of files are not permitted for users on share folder, only updating files by sharing newer version is allowed only if the WIP files along with its meta data (size, date modified, sharing status which used for log usages and registers) are synchronized on peer server [6] across different countries using EDMS application (Fig. 4). File watcher application is used for this purpose to keep an eye on the files meta-data and guarantee identity of files. Sharing process is accompanied by auto email notification including comments and attachments which are used to inform all concerned users about all changes have been made to the files also fixed list of emails can be set by the manager to keep all team leaders and management automatically informed about the sharing and submission process.

Each shared version of files is saved on archive server to track changes (Fig. 5) and to verify the reasons whenever problems or conflict occurs and to determine the reasons and plan for actions. Flexible numbering system for versions ([1A,1B,1C ....etc.] or [1.001, 1.002, 1.003 ...etc.]) is used to control the versions based on project's needs.

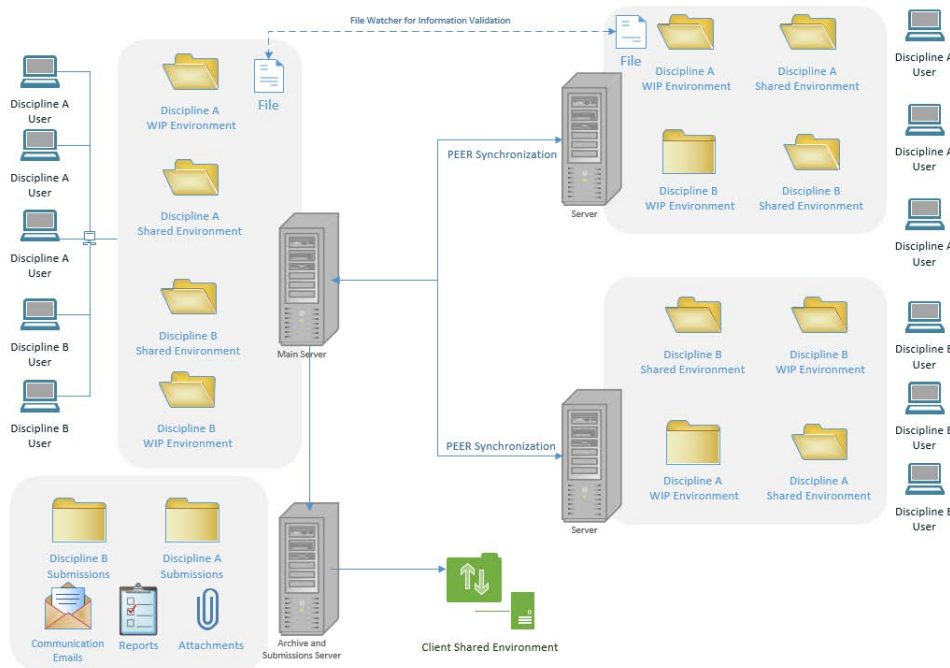


Figure 3: An overall diagram for ECG CDE and EDMS.

*\\Get list of files with older dates:*

```
List<string> Files_with_Lower_Dates = new List<string>();
Files_with_Lower_Dates = EDMS_Functions.Sync.Get_Files_Lower_Date(Leader_XML, Data_Exchange.Leader_name, "WIP");
```

*\\Get dates for all files:*

```
XmlNodeList Qatar_Files_Nodes = Qatar_Dates_XML.SelectNodes("//Files/F");
foreach (XmlNode Single_File in Qatar_Files_Nodes)
{
    if (Single_File.Attributes["Name"].Value.ToUpper() == File_Name)
    {
        Qatar_WIP = Single_File.SelectSingleNode("WIP").Attributes["DT"].Value;
        Qatar_Share = Single_File.SelectSingleNode("Share").Attributes["DT"].Value;
    }
}

XmlNodeList Egypt_Files_Nodes = Egypt_Dates_XML.SelectNodes("//Files/F");
foreach (XmlNode Single_File in Egypt_Files_Nodes)
{
    if (Single_File.Attributes["Name"].Value.ToUpper() == File_Name)
    {
        Egypt_WIP = Single_File.SelectSingleNode("WIP").Attributes["DT"].Value;
        Egypt_Share = Single_File.SelectSingleNode("Share").Attributes["DT"].Value;
    }
}
```

*\\Validate if the file is the latest with warning message:*

```
if (Files_with_Lower_Dates.Contains(Path.GetFileName(File_Path).ToUpper())) {
    MessageBox.Show("ECG says: " + "\n" + "\n" + "File in not synchronized, please submit later."); return; }
```

Figure 4: Part of the code for validating file synchronization across different countries.

*\\Save a Copy of files next versions in submission and archive server:*

```
string S_next_ver = EDMS_Functions.General.Get_next_version(S_ver);
string New_Version_FilePath_in_Store_Qatar =
    Path.Combine(S1_path, Path.GetFileNameWithoutExtension(File_path)
        + "-"
        + S_next_ver
        + "-" + SC + Path.GetExtension(File_path));
string New_Version_FilePath_in_Store_Egypt =
    Path.Combine(Data_Exchange.XML_Pathes, S1_path.Substring(7), Path.GetFileNameWithoutExtension(File_path)
        + "-"
        + S_next_ver + "-"
        + SC + Path.GetExtension(File_path));

if (!EDMS_Functions.File_Actions.Copy_File(File_path, New_Version_FilePath_in_Store_Qatar, true))
{ Errors_Dictionary[File_Name] = Data_Exchange.Messages_Dictionary[6] + Data_Exchange.Catch_Error; return false; }
```

Figure 5: Part of the code for saving file versions.



At any projects stage submission, a proper handover and quality control should be performed as part of the BIM process to verify and validate BIM content. The verification and validation process are reflected automatically on the ECG-EDMS application by a controlled process. Discipline leader have to submit files for approval or rejection by manger and technical reviewers. Submitting the files is not allowed until the WIP files are synchronized across all countries same as sharing process, Check list must be filled along with the submission, Every item on the check list has levels of compliance (0%, 25%, 50%, 75%, 100%) with the ability to add comments shows the reason behind the selection of compliance level (Fig. 6), Every item has 3 levels of acceptance (reject, approve with comments, approve) with an overall acceptance value for all submission (reject, approve with comments, approve). QA/QC report (in excel format) must be attached with the check list (Fig. 7), reviewers can change the compliance level of check list item and adding comments on this item to show the reason behind the change. If the reviewer approve/approve with comment the submission, it will be forwarded to the manager for final decision but if he or the manager rejected the submission, it will return for resubmission action. When the manger approves the submission, all files are collected and saved on predefined stage folder on the submission and archive server (Fig. 8).

*\\ Declare buttons with compliance values and Comments*

```
TextBlock CheckList_Text_box = (TextBlock)main_Grid.FindName("CheckList_Box" + i.ToString());
StackPanel Item_Stackpanel = (StackPanel)main_Grid.FindName("Item" + i.ToString());

Button Item_0 = (Button)main_Grid.FindName("Item" + i + "_0");
Button Item_25 = (Button)main_Grid.FindName("Item" + i + "_25");
Button Item_50 = (Button)main_Grid.FindName("Item" + i + "_50");
Button Item_75 = (Button)main_Grid.FindName("Item" + i + "_75");
Button Item_100 = (Button)main_Grid.FindName("Item" + i + "_100");

Button Comment = (Button)main_Grid.FindName("Comment" + i);
CheckBox NA_CheckBox = (CheckBox)main_Grid.FindName("NA" + i);
```

Figure 6: Code for the interaction panel with user for model's checklist as part of QA/QC.

*\\ confirm the attachment of QA/QC lists as part of the design workflow:*

```
private void send_Click(object sender, RoutedEventArgs e)
{
    if (Data_Exchange.Project_WorkFlow == "DESIGN")
    {
        if (Accepted_QAQC_Data_Grid_Selected_Files.Count != 0)
        { if (Selected_Excel_Pathes.Count == 0) {
            MessageBox.Show("ECG says: " + "\n" + "\n" + "You must attach QA/QC excel files?"
                , "Missing Items? "
                , MessageBoxButton.OK
                , MessageBoxImage.Exclamation); return; } }
    }
}
```

Figure 7: Part of the verification code for the QA/QC process of design workflow.

*\\ saving checklists*

```
if (!EDMS_Functions.File_Actions.Save_XML(Temp_Leader_Checklist, Temp_CheckList_Path))
{ Errors_Dictionary[File_Name] = Data_Exchange.Messages_Dictionary[15] + Data_Exchange.Catch_Error; return false; }
```

*\\ saving QA/QC files in design workflow:*

```
if (Data_Exchange.Project_WorkFlow == "DESIGN")
{
    if (!EDMS_Functions.File_Actions.Copy_File(Excel_File_Path
        , QAQC_Server_Folder_Path
        + Path.GetFileNameWithoutExtension(Excel_File_Path)
        + "[Ver " + S1_ver + " -M]"
        + Path.GetExtension(Excel_File_Path), true))
    { Errors_Dictionary[File_Name] = Data_Exchange.Messages_Dictionary[10] + Data_Exchange.Catch_Error; return false; }
```

*\\ saving approved files for stage submission:*

```
if (Approval_Value == "A" || Approval_Value == "A*")
{
    if (!EDMS_Functions.File_Actions.Copy_File(S1_New_Version_FilePath_in_Store_Qatar
        , Path.Combine(P_path, Stage_and_Date, Path.GetFileName(File_path)), true))
    { Errors_Dictionary[File_Name] = Data_Exchange.Messages_Dictionary[19] + Data_Exchange.Catch_Error; return false; }
```

*\\ saving communication emails as attachment for submission:*

```
EDMS_Functions.File_Actions.Copy_File(Temp_message_path,
    Data_Exchange.XML_Pathes
    + @"ECG-VC\Projects\"
    + Data_Exchange.ProjectNumber_ProjectWorkFlow
    + @"\Leaders\" + Data_Exchange.Leader_name
    + @"\Qatar_Emails\Pub_Emails\"
    + EDMS_Functions.General.Now_Date_Time("YMD_HM")
    + "-" + Data_Exchange.Leader_name
    + ".msg", true);
```

Figure 8: Part of the code for submission processes.

#### 4 AUTOMATING DESIGN INFORMATION EXCTRACTION FROM BIM MODELS FOR REVIEW AND VALIDATION

In Middle East ME region, most of BIM projects are developed on Autodesk platform. Many programming languages can be used (such as: C-Sharp, VB.NET, Ruby, Python, and others) for integrating an automated solution with Autodesk Revit. ECG has developed an interface for automated application for models review and data extraction using C-Sharp and Revit macros were information can be automatically extracted from linked models on demand or on a schedule basis, this depends on the process needs. Whenever any new model or a newer version of an existing model is added to the working servers, the application watcher detects the addition, then Revit macros run to extract the information related to project progress and quality of BIM models content (Fig. 9), then automatically save the extracted information by adding new records to the files inside data folder. Automatic review and analyse of this information are performed by running checks based on projects manuals and criteria. These checks give an indication about level of completeness and compliance of information which the BIM manager can use to ask the working teams for corrective actions if needed (Fig. 10).

*\\ Data extraction from linked models*

```
using (Transaction = new Transaction(doc, "Link File"))
{
    transaction.Start();
    ModelPath linkpath = ModelPathUtils.ConvertUserVisiblePathToModelPath(file.FullName);
    RevitLinkOptions options = new RevitLinkOptions(false);
    RevitLinkLoadResult result = RevitLinkType.Create(doc, linkpath, options);
    RevitLinkInstance.Create(doc, result.ElementId);
    RevitLinkInstance instance2 = RevitLinkInstance.Create(doc, result.ElementId);
    Location = instance2.Location;
}

\\ extraction of numbers of elements per category for progress monitoring
{
    for(int x=0; x<GN_Categories.Count; x++)
    {
        string CategoryName = Category.GetCategory(instance2.GetLinkDocument(), GN_Categories[x].Id).Name;
        string File_BIM = CategoryName.Replace("/", "-") + "_N.csv";
        string First_Line = null;
        if(File.Exists(Path.Combine(BIM_Path, File_BIM)))
        {
            { First_Line = "File Name, Count of " + CategoryName; }
            using (StreamWriter My_Report = new StreamWriter(Path.Combine(BIM_Path, File_BIM), true))
            {
                if(First_Line != null)
                { My_Report.WriteLine(First_Line); }
                int No = 0;
                var L_El = new FilteredElementCollector(instance2.GetLinkDocument()).OfClass(typeof(GN_BuiltInCategories[x]));
                foreach(Element BIM_Element in L_El)
                {
                    if(BIM_Element.GetType().Name == "FamilySymbol" || BIM_Element.GetType().Name.Contains("Type")) {continue;}
                    else {No++;}
                }
                My_Report.WriteLine(file.Name + ", " + No);
            }
            My_Report.Close();
        }
    }
}
```

Figure 9: Part of the code that shows auto-extracting of information related to progress.

*\\ check for family naming*

```
public void Family_Name(Autodesk.Revit.DB.Document doc, string path, string CSV_Path)
{
    foreach(Family F in new FilteredElementCollector(doc).OfClass(typeof(Family)))
    {
        FamilyName.Add(F.Name);
        FamilyCategory.Add(F.FamilyCategory.Name);
    }
    #region Excel Package

    using (ExcelPackage Package = new ExcelPackage(new FileInfo(path)))
    {
        for(int f=0; f<FamilyName.Count(); f++)
        {
            {
                if(FamilyName[f].Split('_').Length < 2)
                {
                    Worksheet1.Cells[f+2,3].Value = "Family name is not Correct";
                }
                else {Worksheet1.Cells[f+2,3].Value = "Family name is Correct";}
            }
        }
    }

    #endregion
}
```

Figure 10: Part of the code that shows auto-reviewing of family naming vs. project standards.

## 5 REAL TIME AVILABILITY OF EXTRACTED INFORMATION FOR MONITOR, REVIEW AND ANALYZE

All the aggregated and extracted information previously illustrated are saved on dedicated server. An application on backend server hosted online is responsible to process the requests coming from frontend application such as add/modify project, add/modify user, get data of charts or tables....etc. Frontend application is responsible to render the data in the form of readable charts and tables with the ability of filtering and sorting (Figs 11, 12 and 13).



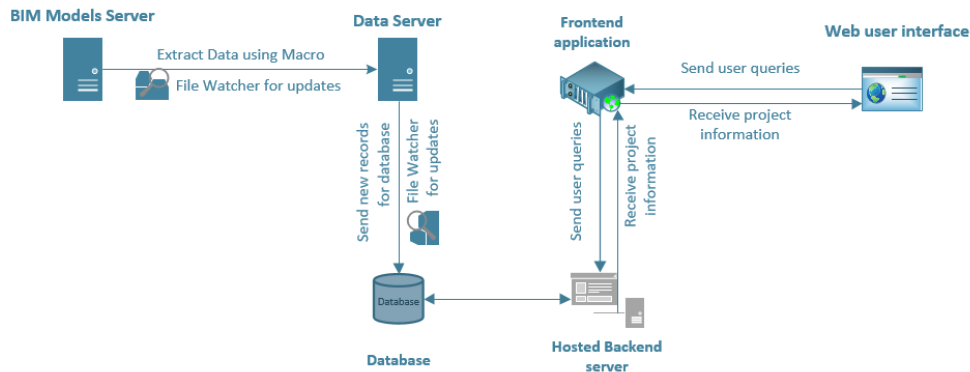


Figure 11: Diagram for the overall data extraction and communication.

*\\ the code shows the sequence of adding project (get request data [name, modules, colour, status], insert the data into the database, create project directory, create modules directories, return response to the frontend)*

```

My_Server10.post('/Add_Project', (sam_req,sam_res) => {

    const {P_Name,P_Module,P_Color,P_Statures} = sam_req.body;
    var Project_Data_directory = './Sync_Data/Data/Current/' + P_Name;

    Sam_knex_DB('projects').returning('*')
        .insert({p_name: P_Name,p_module: P_Module,p_color: P_Color,p_statures: P_Statures})
        .then(Sam1 => {{Sam_Make_Directory(Project_Data_directory)}})
        .then(Sam2 => {
            P_Module.forEach(single_module =>{
                var Module_directory = Project_Data_directory + "/" + single_module
                Sam_Make_Directory(Module_directory)}))
        .then(Sam_Response => {sam_res.json(Sam_Response)});
    .catch(err =>{sam_res.status(400).json("Failed")})

});
    
```

Figure 12: Part of the code for submission processes.

*\\ the code shows the sequence of adding project (get request data [name, modules, colour, status], insert the data into the database, create project directory, create modules directories, return response to the frontend)*

```

if (File_NT[index - 1] === '_n_') {

    if (Show_Hide_N(Data_N)) {

        if(this.state.arrayValue_names.includes(File_Name[index - 1])){

            All_Charts_Tables.push(
                <div key={index}>
                    <br/>
                    <div className='tc pa2 ma2 bw2 shadow-5' style={{backgroundColor: "rgba(255,255,255,0.9)}}>
                        <h2>{File_Name[index - 1]}</h2>
                        <h3>{"Date Modified: " + File_Date[index - 1]}</h3><br/>
                        <SamChart Sam_Hieight={1000} Sam_categories={Xaxis_Array} Sam_Data_YAxis={Bit99_N}></SamChart>
                    </div><br/>
                </div>
            );
        }
    }
}
    
```

Figure 13: Part of the code for submission processes.



## 6 DESIGN PROCESS MANAGEMENT BASED ON COLLECTED INFORMATION

BIM Design processes can be optimized if available information is reviewed and analysed for better decision making and lesson learning from projects. As example on a district project in Doha done using BIM, the models' complexity increases for architectural models which led to lower production efficiency. As justification for this change in performance, the BIM information is analysed and following results was found related to models' content for number of model elements at each model share: Numbers of model elements increased from 72 thousand elements (10 months Progress) to 208 thousand elements (total project duration-one year). Which means an increase of 136 thousand elements in 2-months of progress.

When the latest version of architectural model is analysed, more than 76% of the model content was related to 7 categories of 69 used categories. These 7 categories are related to graphical representation of drawing such as annotation and drafting (wall tags, door tags, room tags, spot elevations, lines and dimensions) (Figs 14 and 15).

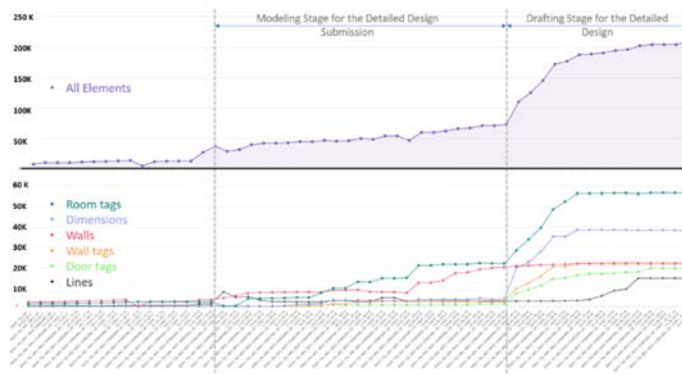


Figure 14: Analysis of BIM modelling and drafting progress across a one-year duration for project detail design stage.

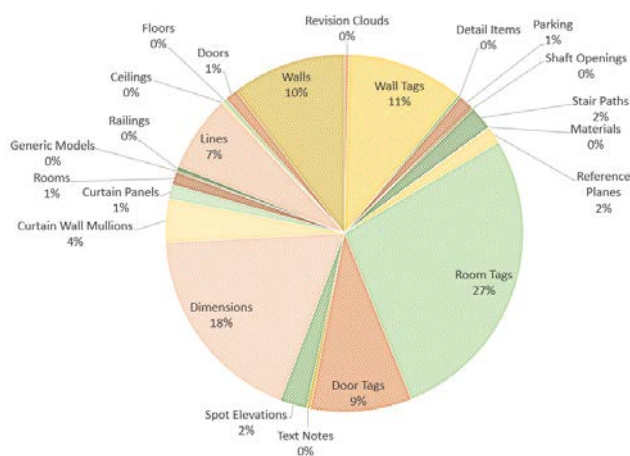


Figure 15: Chart showing the percentage of model elements at the last version of Architectural BIM model.

Duplicated information is considered as one of the most critical issues for BIM production and analysis. At same project, design scope of work of architectural elements was split between the architectural team and the ID team. When the progress of the architectural and interior design models is analysed, design progress issues are clarified. One of the major issues was related to ceiling progress, a duration of 27 days of duplicated ceiling elements was found in both models which at the time of coordination caused confusion related to hosting of ceiling items (lighting fixtures, diffusers, detectors and others) on architectural ceiling elements which was deleted at later stage of project (Fig. 16).

On the same submission when the fire and life safety latest model is analysed, more than 89% of model content was related to 4 categories. These categories are related to model elements such as pipes, pipe fittings, sprinklers and mechanical equipment (Fig. 17). At an intermediate progress share the file size suddenly increase from 150 mb to 233 mb with no justification. When progress graphs reviewed, text notes and detail items were having the same change. After analysing the case graphs deeply, the results were related to exploding CAD details inside Revit, which is not the best practice, corrective action is taken by purging and removing unwanted details components which lead to decrease the file size again to 169 mb (Fig. 18).

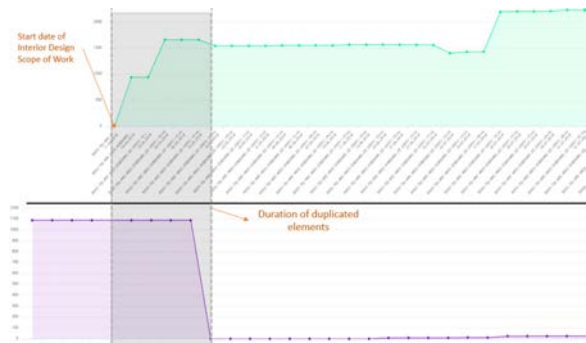


Figure 16: Graph showing the duration of the existence of duplicated ceiling elements.

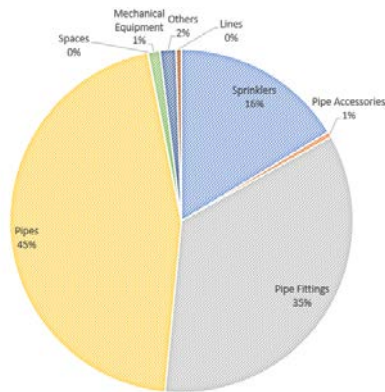


Figure 17: Chart showing the percentage of model elements in the last version of mechanical fire and life safety BIM model.



Figure 18: Analysis showing the change in file size when detail items and text notes are added by exploding cad details inside the model.

## 7 CONCLUSION

The availability of vital BIM information gives reliability for better decision making when processed, analysed and viewed properly using BIM models content and associated information which can be automatically reviewed and corrected. Optimized design process management can be achieved by applying automated solution for information aggregation, extraction and review which plays a very important role in design processes management optimization. In this paper, the overall automated solution is introduced with sample codes for each information management stage and same cases are analysed with encouraging results for wide use of the solution for more applications.

## ACKNOWLEDGEMENTS

The work of these authors was fully funded and supported by Engineering Consultants Group ECGSA under the direct supervision of Dr Abbas Eladawy, Eng. Omar Bahgat and Eng. Emad Kadry.

## REFERENCES

- [1] NBS. The National BIM report, 2019. [www.thenbs.com/knowledge/national-bim-report-2019](http://www.thenbs.com/knowledge/national-bim-report-2019). Accessed on: 15 Jul. 2019.
- [2] Qehaja, A.B., Improving decision making with information systems technology – A theoretical approach. *LIRIA International Review*, **1**, pp. 49–62, 2013.
- [3] McGraw Hill Construction, *The Business Value of BIM for Construction in Major Global Markets: How Contractors Around the World Are Driving Innovation with Building Information Modelling*, 2014.
- [4] NBS, BS 1192:2007, Collaborative production of architectural, engineering and construction information – Code of practice, 2007.
- [5] ISO 19650 Series of International Standards, 19650-1:2018, Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) – information management using building information modelling – Part 1: Concepts and principles, 2018.
- [6] peersoftware, PeerSync File Synchronization for LAN & WAN Environments. [www.peersoftware.com/peersync/](http://www.peersoftware.com/peersync/). Accessed on: 30 Jul. 2019.

