

3

Revisão Conceitual

3.1

Building Information Modeling - BIM

3.1.1

Definição

BIM é um processo baseado em um modelo inteligente para planejamento, projeto, construção e gerenciamento de obras que cria mais do que apenas modelos 2D ou 3D, utilizando-se de uma modelagem paramétrica, onde os objetos possuem inteligência, geometria e dados inerentes a eles. “Um modelo paramétrico é uma representação computacional de um projeto constituído de entidades geométricas que possuem atributos (propriedades) que são fixas e outras que podem variar.” (HERNANDEZ, 2006).

O termo BIM é utilizado muitas vezes de maneira equivocada por desenvolvedores de softwares para descrever o que o seu produto oferece. BIM é um processo e não deve ser confundido com modelagem 3D, modelos que não utilizem os parâmetros em seus dados, ou até mesmo softwares que utilizam múltiplos arquivos de referência em CAD 2D e que precisam ser combinados para definir o modelo final, sem a devida coordenação entre as vistas existentes no projeto. “Definir BIM como um tipo de software, porém, reduz muito o seu significado, que é derivado da longa tradição de pesquisas sobre a utilização do computador como suporte à produção de edifícios.” (SCHEER; FILHO, 2009).

Eastman et al. (2008) define BIM como uma tecnologia de modelagem e conjunto associado de processos para produzir, comunicar e analisar modelos construtivos. Estes modelos são caracterizados por conter componentes construtivos representados digitalmente (objetos), que carregam consigo gráficos computacionais, atributos e regras paramétricas, além de poder conter informações de como eles se comportam, sendo estas muitas vezes requisitadas para a execução de análises e processos de trabalho, como por exemplo estimativa de preços, especificações e análise energética.

Os modelos BIM possuem dados que são consistentes, não-redundantes e coordenados de maneira que, quando ocorre alguma mudança em um componente, esta é observada em todas as vistas do modelo, as quais são representadas de maneira também coordenada.

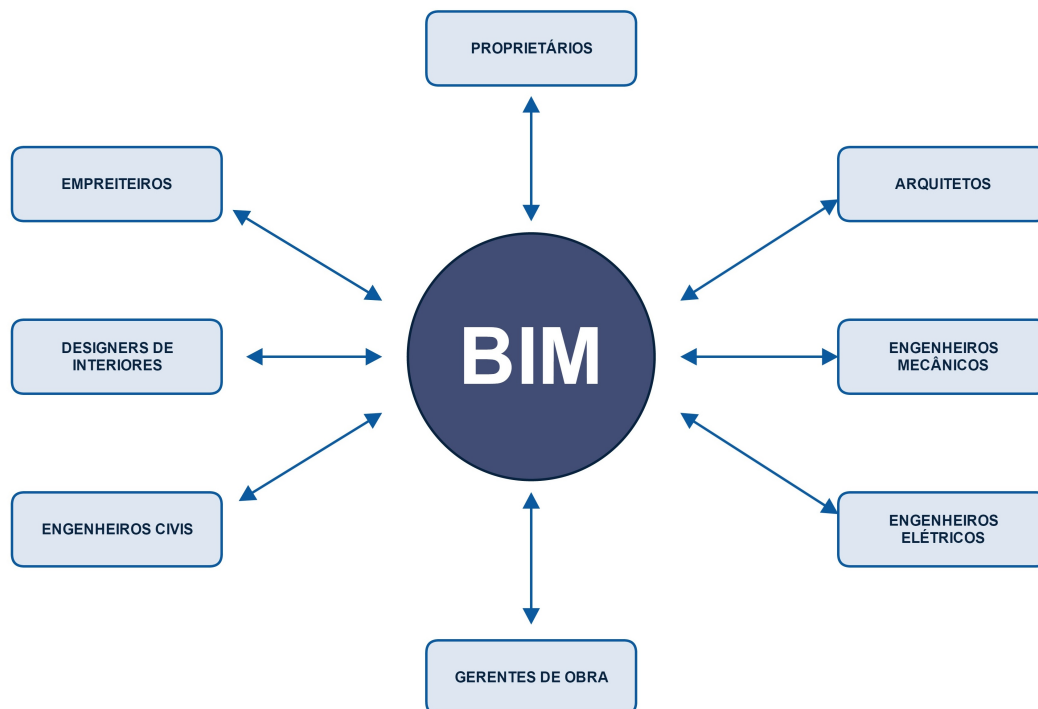


Figura 3.1: Trabalho colaborativo entre os atores em um processo integrado BIM.

Como é possível observar na Figura 3.1, arquitetos, engenheiros estruturais, empreiteiros e outros envolvidos demonstrados, possuem acesso às informações contidas no modelo BIM, extraindo-as quando lhes convier e colaborando com a inserção de novos dados ao longo de todo o ciclo de vida do empreendimento. O fluxo de trabalho, assim, é melhorado com a possibilidade da simultaneidade das ações sobre um único modelo consistente e rico em informações, enfatizando um processo de desenvolvimento de projeto baseado no trabalho colaborativo.

Segundo Holness e Gordon (2006), os benefícios da aplicação do BIM podem ser substanciais, com economia em potencial nos custos de construção variando de 15% a 40% com reduções paralelas nos cronogramas de obra e melhorias na qualidade.

O processo BIM ajuda todos os envolvidos em um projeto a coordenar os dados e ter comunicação direta com todos os integrantes da equipe, trabalhando no mesmo modelo e tendo o conhecimento das diferentes disciplinas envolvidas transferido com fluidez entre todos, o que contribui para aumentar a precisão das informações e reduzir o retrabalho. BIM tem o potencial de promover maior eficiência e harmonia inclusive entre os envolvidos que, anteriormente, se consideravam adversários, afirma Azhar et al. (2008 apud AZHAR,

2011).

Entretanto, enquanto BIM oferece novos métodos para se ter um trabalho colaborativo, têm-se também a introdução de problemas quanto à criação de equipes eficientes, já que haverá uma grande mudança no processo de desenvolvimento do projeto. Métodos devem ser determinados para que o modelo de informação seja devidamente compartilhado entre todas as partes envolvidas no projeto, além da necessidade de formalizar os requisitos a serem atendidos por cada integrante da equipe de trabalho.

No setor de AEC, aliada à aplicação do processo BIM, é crescente e cada vez mais comum a adoção de um novo método de desenvolvimento e entrega de projeto: o *Integrated Delivery Project* (IPD). A utilização dessa metodologia contribui para uma melhora significativa no trabalho colaborativo, criando um ambiente integrado e aberto (todos têm acesso às informações de projeto) entre todas as partes, incluindo o cliente ou um representante do mesmo. Utiliza-se, então, a *expertise* dos profissionais envolvidos para antecipar decisões importantes, diminuir custos, reduzir cronogramas, criar projetos baseados em conceitos de construtibilidade e sustentabilidade, por exemplo, obtendo assim, um produto final de maior qualidade.

3.1.2 Integrated Project Delivery - IPD

Integrated Project Delivery (IPD) ou, em português, Entrega de Projeto Integrado, consiste, de acordo com AIA (2007), em um método de relação contratual de trabalho que integra pessoas, sistemas, estruturas de negócios e práticas (Figura 3.2) em um processo que, colaborativamente, aproveita os talentos e visões de todos os participantes de maneira a otimizar os resultados do projeto, aumentar o valor para o proprietário, reduzir gastos, além de maximizar a eficiência ao longo de todas as fases de projeto, fabricação e construção.

Este novo método abordado é baseado na colaboração, que por sua vez, é baseada em confiança. AIA (2007) afirma que, um ambiente colaborativo de confiança e efetivamente estruturado estimula as equipes a focarem mais no resultado final do projeto do que em suas próprias metas. Sendo assim, os envolvidos precisam ser de certa forma re-educados e treinados, abraçando os princípios do *IPD* para que este gere resultados positivos.

No IPD tem-se principalmente, segundo AIA (2007), três categorias de envolvidos: proprietários, construtores e projetistas. Além do benefício geral que este processo oferece em relação ao desenvolvimento sustentável, existe para os proprietários a vantagem de poder estar mais inserido no processo,

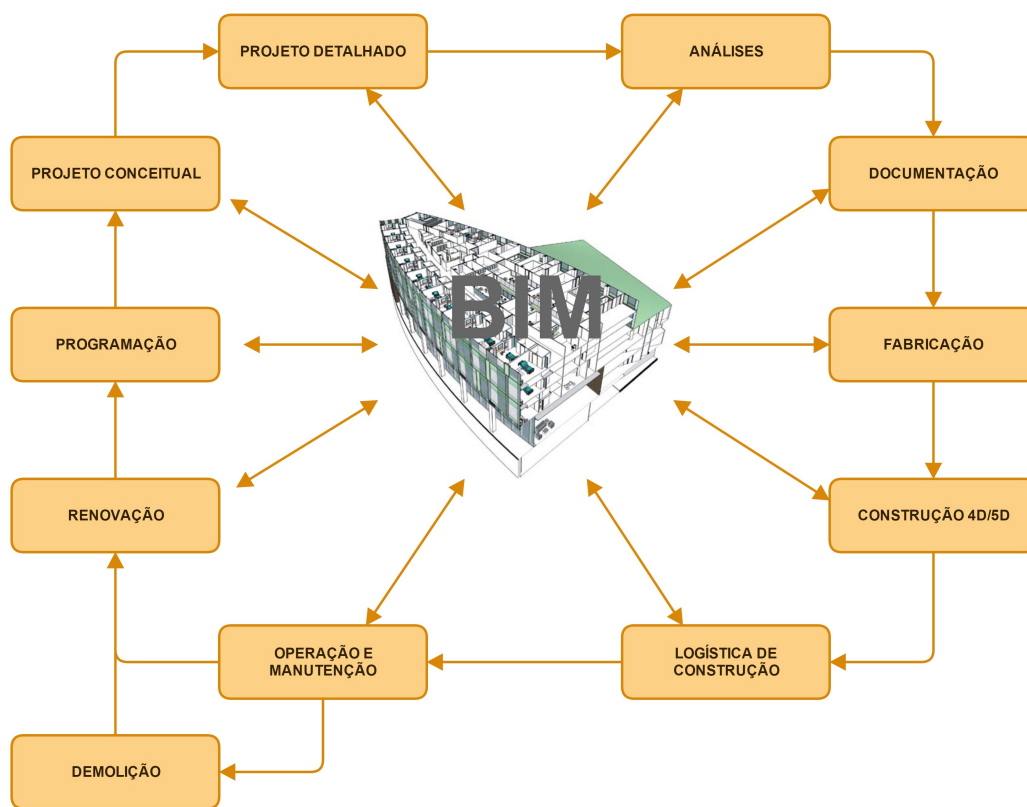


Figura 3.2: Esquema de IPD em processos BIM na indústria AEC, adaptado de Addor et al. (2010).

podendo antecipar decisões, prever e gerenciar melhor os seus custos e assim controlar melhor seu investimento.

Para os construtores, a prática do IPD permite que eles contribuam desde o princípio, com seu conhecimento, nas técnicas construtivas a serem adotadas, promovendo como resultado um projeto de maior qualidade e menos suscetível a erros durante a fase de construção, evitando, assim, gastos desnecessários que possam surgir.

Finalmente, para os projetistas, o método integrado de entrega de projeto permite o beneficiamento antecipado da contribuição da experiência dos construtores para a prevenção de futuros possíveis problemas e conflitos entre as disciplinas envolvidas no projeto, como Hidráulica, Elétrica, Estruturas, etc. Com o IPD, é possível criar projetos com maior qualidade, mais eficientes, sustentáveis, utilizando conceitos de construtibilidade em ambiente colaborativo.

Integrated Project Delivery provoca grandes mudanças no que diz respeito a quantidade de esforço aplicado e capacidade de alterações impactarem no projeto ao longo de todo o ciclo de vida do mesmo, desde a sua conceituação até a fase de construção. A denominada “Curva de Macleamy” (Figura 3.3),

demonstra essa mudança em relação ao método tradicional de entrega, onde é possível observar que a curva correspondente ao processo de projeto *IPD* concentra seus esforços nas fases onde a capacidade de alterações impactarem no custo geral do projeto é maior e os custos de mudança de projeto são baixos, o que demonstra uma maior eficiência do método integrado em relação ao processo tradicional.

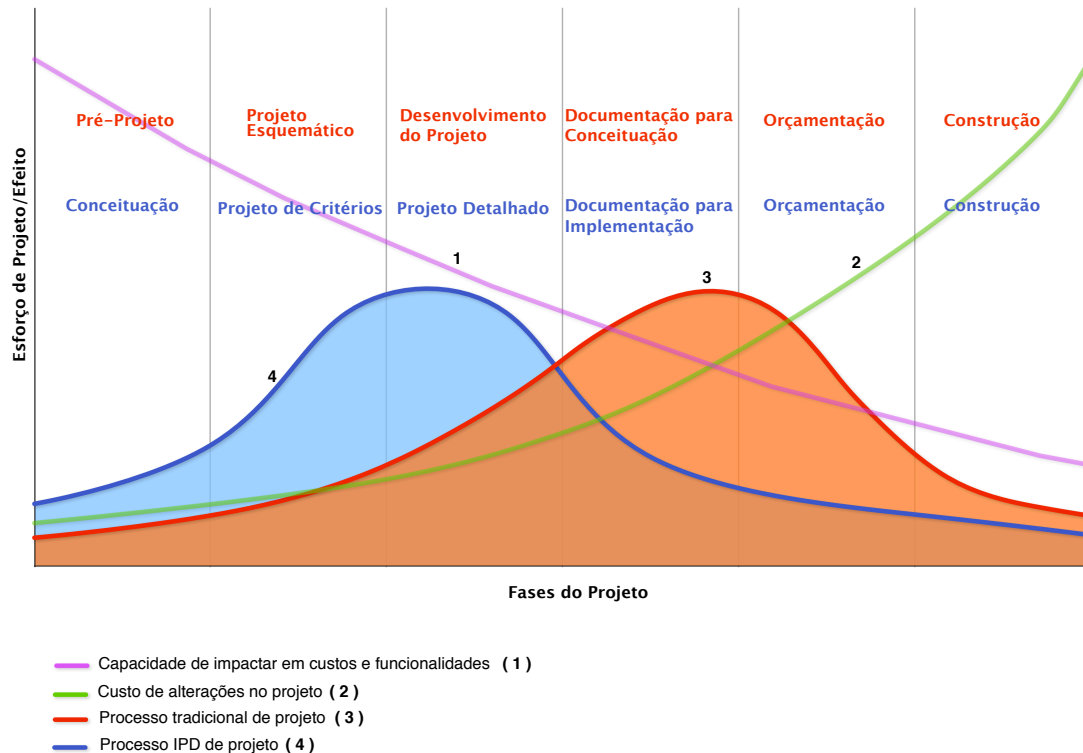


Figura 3.3: A “Curva de Macleamy”, adaptado de Eastman et al. (2008).

De uma maneira mais detalhada, o gráfico mostrado na Figura 3.3 ilustra a relação entre esforço de projeto e tempo, indicando como se dá a distribuição desse esforço no processo tradicional (linha 3) e no processo IPD. Quando se refere ao tempo, o mesmo é dividido em fases de projeto, as quais são diferenciadas para cada tipo de processo (em vermelho para o processo tradicional e em azul para o processo IPD). Além disso, as linhas 1 e 2 demonstram, respectivamente, a capacidade de alterar o projeto sem que haja grande impacto em custos, representando o valor das decisões tomadas em função do tempo, e o custo causado por alterações no projeto no decorrer das diferentes fases de projeto.

3.1.3

Tipos de aplicativos nos processos BIM

Eastman et al. (2008) fez uso explícito de alguns termos muito utilizados informalmente para designar a hierarquia existente nos aplicativos baseados em BIM e dividiu em três categorias: Ferramenta BIM, Plataforma BIM e Ambiente BIM.

Entende-se como Ferramenta BIM aquele aplicativo que foi designado para a realização específica de uma tarefa e que produz como documento de saída um resultado também específico, como por exemplo, ferramentas de análise estrutural. Os documentos produzidos pelas Ferramentas BIM geralmente são padronizados, como relatórios e desenhos, todavia em alguns casos é possível exportar os resultados para outras ferramentas para, por exemplo, levantamento de quantitativos para estimativa de custos.

As chamadas Plataformas BIM são softwares normalmente para projeto, que geram dados para múltiplos usuários. Geram um modelo de dados primário que armazena estas informações na plataforma. A grande maioria das plataformas BIM também internamente incorporam funcionalidades de ferramentas como, por exemplo, a produção de desenhos e *detecção de Clash*¹. Além disso, as plataformas comumente incorporam interfaces para múltiplas outras ferramentas com variados níveis de integração.

O surgimento de ambientes BIM ocorre quando se começa a gerenciar um ou mais canais de informação que integram as aplicações (ferramentas e plataformas) dentro de uma empresa. O ambiente, então criado, suporta políticas e práticas de informação dentro de uma organização. Geralmente estes ambientes não são conceituados, crescem dentro da firma pelas necessidades que vão surgindo, e acabam sendo convencionados com a referida nomenclatura.

3.2

Interoperabilidade

Na seção anterior, conceituou-se BIM e foi abordado sobre o uso de novas metodologias de desenvolvimento de projeto baseado em um trabalho colaborativo, como por exemplo o IPD (*Integrated Project Delivery*). No entanto, não é possível abordar temas como IPD e trabalho colaborativo sem citar a interoperabilidade que deve existir para que o processo funcione.

Um estudo do NIST (sigla para *National Institute of Standards and Technology*) em 2004 aponta a falta de interoperabilidade entre os softwares AEC como um custo para a indústria de \$15,8 bilhões anualmente (AIA,

¹Segundo VicoSoftware (2015), é o processo de integrar vários modelos entre si e verificar possíveis conflitos a serem resolvidos.

2007, p. 3). Estatísticas como essa, demonstram a importância de se buscar a interoperabilidade e o seu aperfeiçoamento.

Múltiplas aplicações, com sobreposição de seus respectivos requisitos de dados, suportam várias atividades relacionadas ao projeto e construção. Interoperabilidade é a capacidade de trocar informação entre as aplicações, suavizando fluxos de trabalho e algumas vezes facilitando a automação (EASTMAN et al., 2008, p. 99).

De uma maneira mais geral, é a habilidade de dois ou mais sistemas ou componentes trocarem informação e usarem essa informação que foi trocada (COMMITTEE, 1990, p. 114). A interoperabilidade se tornou então sinônimo da capacidade de sistemas de múltiplas informações coexistirem, interagirem e compreenderem um ao outro durante funcionalidades de troca (CHEN; DAHLIN et al., 2006).

Ainda segundo Eastman et al. (2008), a interoperabilidade tem tradicionalmente confiado em formatos de troca baseados em arquivos limitados a geometria, como DXF (*Drawing eXchange F*ormat) e IGES (*Initial Graphic Exchange Specification*). Um problema de interoperabilidade surge quando dois ou mais sistemas incompatíveis são relacionados (NAUDET et al., 2010 apud POIRIER; FORGUES; STAUB-FRENCH, 2014). Trocas de informação (via direta) baseadas nas Interfaces de Programação de Aplicativos (*Application Programming Interfaces - APIs*) são as mais antigas e ainda importantes rotas para a interoperabilidade.

Iniciando no final da década de 80, modelos de dados foram desenvolvidos para dar suporte à troca de informação de produto e modelo de objeto dentro das diferentes indústrias, liderados pelo esforço aplicado na criação da padronização internacional *ISO-STEP*. Modelos de dados distinguem o *schema* utilizado para organizar os dados e a linguagem do *schema* para transportar os dados. Eastman et al. (2008) afirma ainda, que alguns tradutores são capazes de traduzir de uma linguagem de *schema* para outra, como por exemplo de *IFC* para *XML*.

Os dois principais modelos de dados de produtos da construção são as *Industry Foundation Classes* (IFC), para planejamento da construção, projeto, execução e gerenciamento, e o *CIMsteel Integration Standard Version 2* (CIS/2), para armazenar informações de projetos de aço estrutural. Um outro exemplo de padrão STEP é o ISO-15926, para automação de sistemas industriais e integração.

De acordo com Eastman et al. (2008), foram distinguidos três tipos de aplicativos BIM: ferramentas, plataformas e ambientes, como definidos no item 3.1.3. Interoperabilidade suporta diferentes capacidades e apresenta

diferentes problemas no intercâmbio de informações entre estes três tipos. A mais comum e fundamental ocorrência de troca de dados se dá entre plataforma e ferramenta BIM, como por exemplo, uma ferramenta de análise estrutural, podendo ocorrer de forma direta de um aplicativo para o outro, fazendo uso de uma API (normalmente ocorre entre softwares de um mesmo fabricante), ou utilizando os formatos neutros de arquivo, como o IFC, a fim de promover a interoperabilidade entre os softwares.

Nesta interação entre plataforma e ferramenta, Eastman et al. (2008) afirma que porções específicas do modelo de dados nativo da plataforma (a estrutura de dados que a plataforma utiliza internamente) são traduzidas, sendo que esta tradução é realizada definindo, na plataforma, os dados necessários (chamado de um *model view*) e colocando esses dados no formato exigido pela ferramenta, completando com informações que não estão contidas no modelo original.

A troca de dados entre plataforma e ferramenta pode ser complexa. Extrair o modelo analítico para análise estrutural e determinar cargas relevantes ainda não é uma tradução automatizada comum, já que esta requer conhecimento e julgamento humano (EMKIN, 1988 apud EASTMAN et al., 2008).

De uma maneira mais direta tem-se, entre ferramentas, as trocas de dados, que são restritas devido aos limitados dados disponíveis na aplicação que está exportando (EASTMAN et al., 2008).

O maior desafio de interoperabilidade se dá na comunicação entre plataformas. Isto inclui plataformas de modelagem como ArchiCAD, Revit e plataformas de projeto digital e fabricação como o Tekla, SDS/2 Structureworks, e StruCad, CADPipe, e CAMduct. Plataformas não somente incorporam uma grande gama de informações como também incorporam regras que administram a integridade dos objetos. Algumas vezes a troca de informações simplesmente não é possível, sendo que o maior desafio não é transferir objetos com formas fixas e algumas extrusões e, sim, a interpretação das regras inerentes aos objetos, como por exemplo a relação entre uma esquadria e uma parede (EASTMAN et al., 2008).

3.2.1 buildingSMART

Em 1995 Autodesk organizou uma aliança privada com a participação de mais onze empresas para provar os benefícios da interoperabilidade, ou do completo intercâmbio de informações entre os diferentes softwares usados na Indústria da Construção.

A Aliança foi formada pelas seguintes empresas: Autodesk, Archibus,

AT&T, Carrier Corporation, HOK Architects, Honeywell, Jaros Baum & Bolles, Lawrence Berkeley Laboratory, Primavera Software, Softdesk Software, Timberline Software e Tishman Construction.

Estas empresas estavam engajadas em áreas de projetos construtivos, engenharia, construção e desenvolvimento de software, reunidas por um interesse comum em investir no futuro da Indústria da Construção.

Após um ano de pesquisa, os membros da Aliança chegaram a conclusão de que a interoperabilidade não só era viável, como tinha um grande potencial comercial. Além disso, perceberam que os padrões tinham que ser internacionais, em formato aberto e que a Aliança, antes privada, agora deveria aceitar novos membros interessados.

Em 16 de Maio de 1996, a IAI (*International Alliance for Interoperability*) foi estabelecida em Londres durante uma reunião entre representantes da América do Norte, Europa e Ásia.

Finalmente, no dia 11 de janeiro de 2008, a IAI mudou seu nome para *buildingSMART* para melhor refletir a natureza e objetivos da organização.

A aliança *buildingSMART* é formada por Capítulos e Membros, sendo que os primeiros são organizações locais em países específicos que se preocupam em implementar o chamado *open BIM* no referido país.

Quanto à organização interna da instituição (Figura 3.4), esta possui um Conselho Internacional, o qual é o corpo governante legal da *buildingSMART* Internacional, e consiste em dois representantes para cada Capítulo. É constituída também de um quadro de diretores, denominado ExCom, eleitos pelo Conselho Internacional para liderar a *buildingSMART* e sua rede de Capítulos ao redor do mundo. Este grupo de liderança, por sua vez, é assistido por um grupo de representantes *seniors* das lideranças internacionais de negócios nas indústrias da construção e infraestrutura, formalizado como Conselho de Assessoramento Estratégico.

A nível executivo, a *BuildingSMART* possui três programas principais: Padrões, Adesão e Capítulos (Figura 3.4). Juntos, esses programas são destinados a impulsionar o desenvolvimento e uso ativo de padrões abertos internacionalmente reconhecidos, ferramentas, regimes de treinamento e certificação para dar suporte à utilização cada vez maior da metodologia BIM nas indústrias *AEC/FM*² pelos setores da Construção e Infraestrutura.

²Indústrias de Arquitetura, Engenharia e Construção (*AEC*) e Gerenciamento de Instalações Industriais (*FM*).

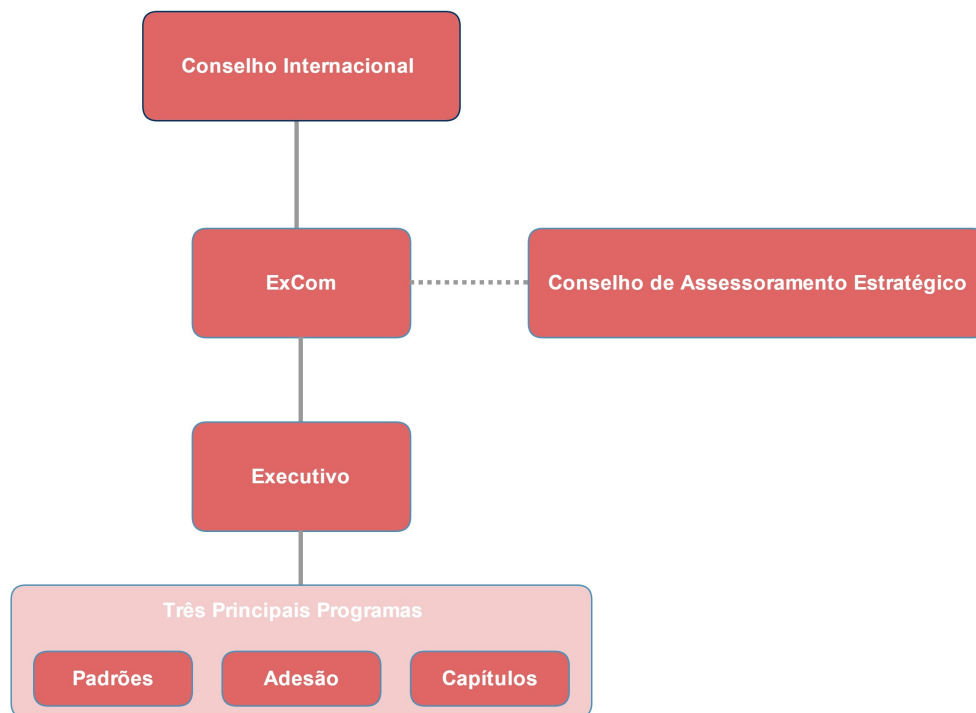


Figura 3.4: A organização da buildingSMART

3.2.2

Industry Foundation Classes - IFC

O *Industry Foundation Classes* (IFC) é um *schema* desenvolvido para definir um extenso conjunto consistente de representações de dados de informação da construção para intercâmbio entre aplicações de software da indústria AEC. O formato IFC tem como base a linguagem e conceitos ISO-STEP³ EXPRESS para a sua definição, com algumas pequenas restrições na linguagem EXPRESS.

Enquanto a maioria das outras tentativas baseadas no padrão ISO-STEP focaram na troca detalhada de informações entre os softwares dentro dos domínios específicos da engenharia, pensava-se que na indústria da construção isso levaria a resultados fragmentados e um conjunto de padrões incompatíveis. Ao invés disso, o IFC foi projetado como um modelo de *framework*⁴ extensível.

³STEP (STandard for the EXchange of Product model data) é o nome informal que se dá a norma ISO 10303, a qual consiste em um padrão para a representação e troca de informações computacionais da fabricação de produtos.

⁴Em desenvolvimento de software, é um ambiente universal e reutilizável que oferece funcionalidade específica como parte de uma ferramenta computacional maior visando facilitar o desenvolvimento de softwares, produtos e soluções.

Isto é, os desenvolvedores do *schema* projetaram para que o IFC oferecesse definições amplas e gerais dos objetos e dados a partir dos quais modelos mais detalhados e específicos para a realização de uma tarefa poderiam ser definidos.

Sendo assim, o IFC foi projetado para lidar com quase todas as informações da construção ao longo do ciclo de vida do empreendimento, desde a viabilidade e planejamento, passando pelo projeto, incluindo a análise e simulação, construção, ocupação e, até mesmo operação do empreendimento (KHEMLANI, 2004).

O modelo de dados baseado em IFC pode ser definido por múltiplos formatos, suportando diferentes codificações para a mesma base de dados. Estes formatos são: **IFC-SPF**, **IFC-XML** e **IFC-ZIP**.

O IFC-SPF, segundo buildingSMART (2015b), é o formato mais utilizado e consiste em um formato de texto definido pela ISO 10303-21 (“*STEP-File*”) onde cada linha normalmente consiste em um registro de objeto, tendo a extensão “.ifc” e a vantagem de ser um formato compacto e de código legível. O IFC-ZIP é um formato comprimido do arquivo IFC-SPF e possui a extensão “.ifcZIP”.

Já o IFC-XML, como o próprio nome sugere, é um formato XML definido pela ISO 10303-28 (“*STEP-XML*”), cuja extensão é denominada “.ifcXML” e é apropriado para interoperabilidade com ferramentas XML envolvidas no processo.

Em 2010, uma nova versão do IFC foi lançada: a versão IFC2x4. Este lançamento possui aproximadamente 800 entidades (objetos de dados), 358 conjuntos de propriedades e 121 tipos de dados. Enquanto esses números indicam a complexidade do IFC, eles também refletem a riqueza semântica da informação da construção, atendendo a múltiplos sistemas diferentes, às diferentes necessidades das aplicações, abrangendo desde a análise energética e estimativa de custos até rastreamento de materiais e planejamento de cronograma, por exemplo. Interfaces baseadas no novo formato estão atualmente sendo implementadas pela maioria dos fabricantes de ferramentas e plataformas BIM, substituindo a antiga e ainda mais usual versão IFC2x3.

Enquanto o IFC é capaz de representar uma grande gama de informações de projeto, engenharia e produção, a possibilidade de dados que podem ser trocados na indústria AEC é enorme. O IFC passa a atender cada vez mais as necessidades dos usuários e desenvolvedores conforme elas vão surgindo e assim novas edições são lançadas.

Tratando um pouco da estrutura de dados do arquivo IFC, segundo Liebich et al. (2004), existem três tipos de entidade fundamentais no modelo IFC: objetos, relações e propriedades. Derivadas do *IfcRoot* (Figura 3.5), estas

entidades formam o primeiro nível de especialização dentro da hierarquia de classes IFC.

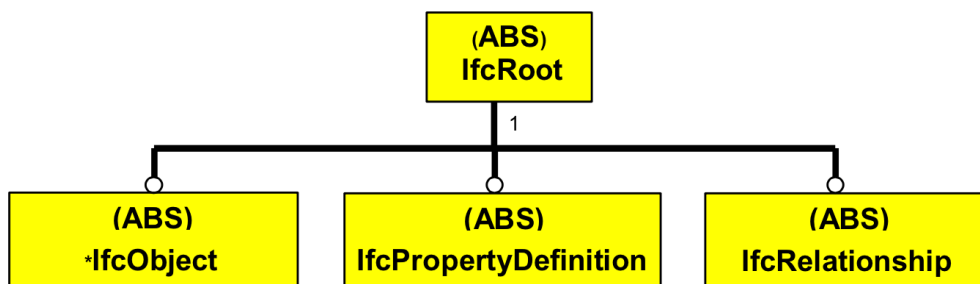


Figura 3.5: Definição da árvore de subtipos de IfcRoot. Retirada de Liebich et al. (2004).

Um objeto é o *supertipo*⁵ abstrato, *IfcObject* (Figura 3.6), e representa todos os itens tangíveis, como paredes, vigas ou cobertura, itens fisicamente existentes, como espaços, e itens conceituais como *grids*⁶ ou contornos virtuais. Este também incorpora processos, tais como tarefas de trabalho; controles, como itens de custo; recursos, como recurso de trabalho; e atores, como pessoas envolvidas no projeto ou processo construtivo, etc (LIEBICH et al., 2004).

Um objeto extrai sua informação contextual das relações nas quais está envolvido. A informação de propriedade e, se disponível, sobre o tipo específico de objeto subjacente. Pode estar associado a um descritor informal de tipo, que, por sua vez, denota um tipo particular para posteriormente especificar o objeto. As relações nas quais o objeto pode estar envolvido estão descritas na Tabela 3.1.

No modelo IFC existem sete tipos de entidade fundamentais derivadas do *IfcObject* e estes formam o segundo nível de especialização dentro da hierarquia de classes existente no IFC, sob a ramificação de objeto. Na Figura 3.7 é possível observar a hierarquia que se segue abaixo do *IfcObject* com as classes *IfcProduct*, *IfcProcess*, *IfcControl*, *IfcResource*, *IfcActor*, *IfcGroup* e *IfcProject*. Na Tabela 3.2 são conceituadas as terminologias utilizadas nestas classes.

Quando se trata de geometria, o IFC tem meios para representar uma grande quantidade delas, incluindo extrusões, sólidos definidos por um conjunto conectado fechado de faces que delimitam o volume (B-REPS), e formas definidas por uma árvore de formas e operações de união e interseção (utilizando adição e subtração e/ou Geometria Sólida Construtiva).

⁵Tipo de entidade que possui relação com um ou mais subtipos e possui atributos em comum com estes.

⁶Em arquitetura, consiste em uma rede de linhas horizontais e verticais sobrepostas em um mapa, planta de construção, entre outros, utilizada como referência para localizar pontos.

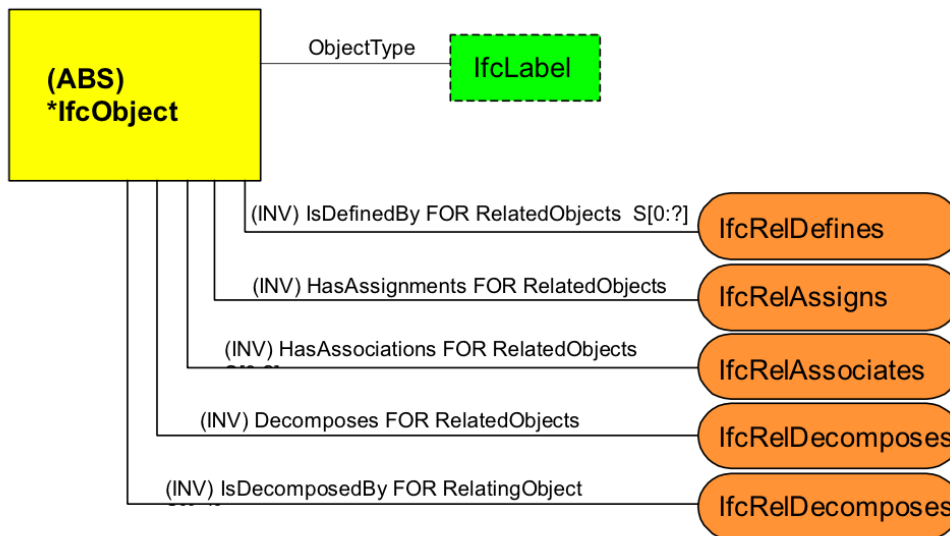


Figura 3.6: Definição de IfcObject. Retirada de Liebich et al. (2004).

Tabela 3.1: Relações às quais o objeto pode estar associado.

IsDefinedBy	Associação de um tipo de objeto (ou estilo de objeto), ou conjunto de propriedade compartilhável a um objeto, que aplica a mesma informação de propriedade a muitas instâncias de objeto.
HasAssignments	Relações com outros objetos da árvore. O significado dessas relações é posteriormente especializado em um nível mais baixo da hierarquia de classes do modelo IFC.
HasAssociations	Relações com conceitos definidos pelos recursos IFC, tais como classificação, biblioteca ou referências de documento, ou definições de material.
Decomposes/IsDecomposedBy	Relações que permitem qualquer objeto a ser decomposto em partes.

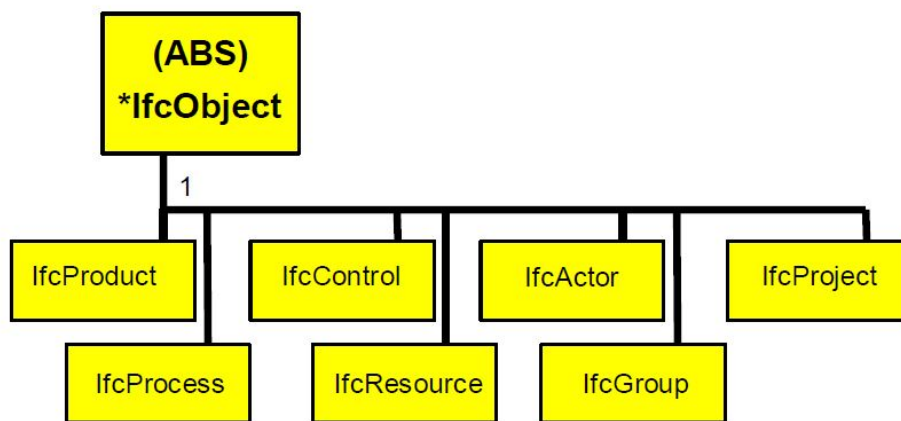


Figura 3.7: Tipos de entidade derivados de *IfcObject*. Retirada de Liebich et al. (2004).

Tabela 3.2: Conceituação das terminologias utilizadas nos subtipos de *IfcObject*

Produtos (<i>IfcProduct</i>)	São objetos físicos (manufaturados, fornecidos ou criados) para incorporação em um projeto. Eles podem ser fisicamente existentes ou tangíveis. Os produtos podem ser definidos por representações da forma e ter uma localização em um espaço de coordenadas.
Processos (<i>IfcProcess</i>)	São ações que ocorrem em um projeto com o propósito, por exemplo, de aquisição, construção ou manutenção de objetos. Os processos ocorrem em seqüência no tempo.
Controles (<i>IfcControl</i>)	São conceitos que controlam ou restringem outros objetos. Controles podem ser vistos como guia, especificação, regulação, restrição ou outra requisição aplicada a um objeto que precisa ser atendida.
Recursos (<i>IfcResource</i>)	São conceitos que descrevem o uso de um objeto principalmente dentro de um processo.
Atores (<i>IfcActor</i>)	São agentes humanos que estão envolvidos em um projeto ao longo do ciclo de vida do mesmo.
Projeto (<i>IfcProject</i>)	É a realização de algumas atividades de engenharia direcionadas ao produto.
Grupo (<i>IfcGroup</i>)	É uma coleção arbitrária de objetos.

Por padrão, a maioria das formas são exportados como B-Reps. Com o lançamento da edição 2X4, as superfícies podem ser aquelas definidas pelas formas extrudadas (incluindo os extrudados ao longo de uma curva), Bezier, e agora superfícies *Non Uniform Rational Basis Spline* (NURBS). Partes das formas podem ser distinguidas como recursos de formas, os quais suprem quase todas as necessidades da construção e a maioria das necessidades de projeto.

No que tange às relações associadas a um objeto em modelo IFC, estas são tipificadas e ligam um objeto a outro, sendo tratadas dentro do tipo de entidade fundamental do IFC chamado *IfcRelationship* (EASTMAN et al., 2008). Esta classe é a maneira desejável de lidar com os relacionamentos entre objetos e permite que propriedades específicas de relacionamento possam ser mantidas diretamente no objeto de relacionamento, tornando, ainda, possível que as semânticas de relacionamento sejam separadas dos atributos de objeto (LIEBICH et al., 2004).

Derivadas da classe *IfcRelationship* existem cinco tipos de relacionamento fundamentais no modelo IFC, formando um segundo nível de especialização dentro da hierarquia de classes IFC na ramificação correspondente a Relacionamento. A árvore da hierarquia de classes correspondente aos subtipos imediatos da classe *IfcRelationship* estão representados na Figura 3.8, enquanto que a Tabela 3.3 conceitua as terminologias.

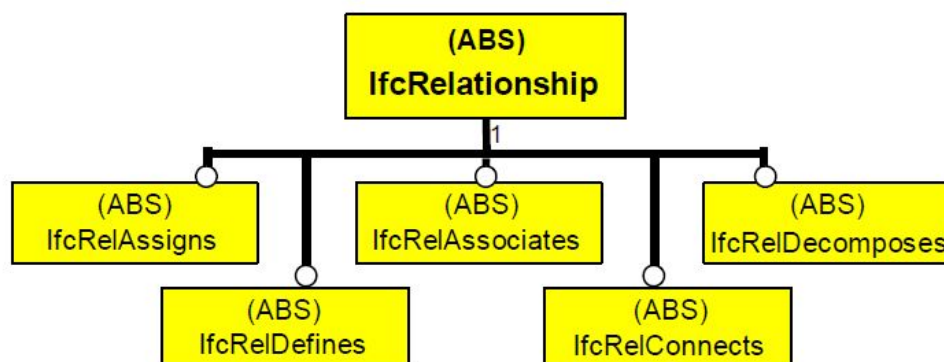


Figura 3.8: Definição dos subtipos de *IfcRelationship*. Retirada de Liebich et al. (2004).

Além de *IfcObject* e *IfcRelationship* como subtipos fundamentais em um modelo IFC, há também o *IfcPropertyDefinition*, que como o próprio nome sugere, é a generalização de todas as características dos objetos. Este subtipo reflete a informação específica de um tipo de objeto, perante a informação de ocorrência da instância de objeto atual no contexto de projeto.

Segundo Liebich et al. (2004), existem dois conceitos fundamentais de definição de tipos de propriedade no modelo IFC, derivados da classe

Tabela 3.3: Conceituação das terminologias utilizadas nos subtipos de *IfcRelationship*

Atribuição (<i>IfcRelAssigns</i>)	É a generalização dos relacionamentos de “link” entre instâncias de objetos e seus variados subtipos
Associação (<i>IfcRelAssociates</i>)	Refere-se a fontes externas de informação (grande maioria notavelmente uma classificação, biblioteca ou documento) e associa estas a objetos ou definições de objeto.
Decomposição (<i>IfcRelDecomposes</i>)	Define um conceito geral de elementos sendo compostos ou decompostos. O relacionamento de decomposição simboliza uma hierarquia total ou parcial com a habilidade de navegar do todo (a composição) para as partes e vice versa.
Definição (<i>IfcRelDefines</i>)	Utiliza uma definição de tipo ou de conjunto de propriedade (visto como informação de tipo parcial) para definir as propriedades da instância de objeto.
Conectividade (<i>IfcRelConnects</i>)	Define a conexão entre dois ou mais objetos. A semântica específica da conexão é declarada dentro dos subtipos, definidas em qualquer lugar no modelo IFC. Um exemplo é a <i>IfcRelSequence</i> , que trata da concatenação dos processos no tempo.

IfcPropertyDefinition, formando um segundo nível de especialização dentro da hierarquia de classes IFC, na ramificação da definição de propriedade, mais especificamente, o *IfcTypeObject* e o *IfcPropertySetDefinition* (Figura 3.9).

A classe *IfcTypeObject* define uma informação específica sobre um tipo, sendo que este último, por sua vez, define todas as informações de propriedade e também, para o produto do tipo, todas as informações da geometria. Já a classe *IfcPropertySetDefinition* define conjuntos extensíveis e compartilháveis de propriedades podendo estar ligados a ocorrências de objetos. O conjunto de propriedades é considerado como uma informação parcial de tipo, já que esta forma um conjunto de informações comuns de propriedades compartilhadas entre as ocorrências de objeto.

O IFC, desenvolvido para responder a diferentes necessidades dos projetistas, empreiteiros, fornecedores, fabricantes, e outros, é de certa forma rico em informações e ao mesmo tempo redundante. Seus múltiplos tipos de geometria, propriedades e relações, são necessários para identificar a informação

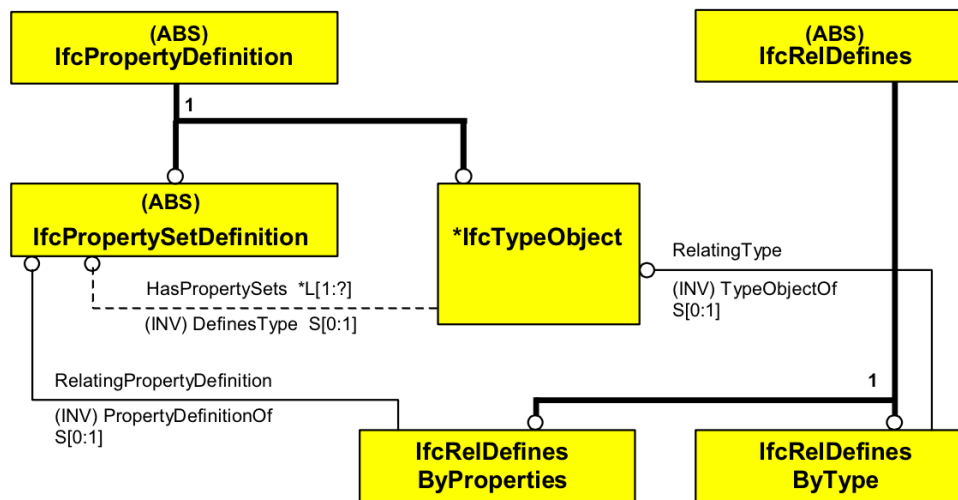


Figura 3.9: Definição dos subtipos do *IfcPropertyDefinition*. Retirada de Liebich et al. (2004).

necessária para realizar alguma determinada troca ou tarefa, o que leva a constatar a redundância presente no modelo de dados (EASTMAN et al., 2008, p. 120).

Para intercâmbio de informação entre os diferentes envolvidos no projeto, o conhecimento sobre os requisitos do fluxo de trabalho e da tarefa em questão vem sendo apontado como fator crítico para o sucesso. Não basta simplesmente importar ou exportar o arquivo IFC, deve-se determinar subconjuntos dentro do *schema* IFC com o intuito de realizar troca de informação associada à tarefa. Por exemplo, o fluxo de trabalho que envolve intercâmbio de dados entre arquiteto e engenheiro estrutural, onde o engenheiro não necessita receber todos os dados da arquitetura e sim, apenas a parte que lhe interessa para análise da estrutura.

Este nível de especificidade, que envolve a identificação das informações necessárias para suportar uma determinada troca é denominado como *Model View Definition* (MVD) e será abordado com mais detalhe no item 3.2.4.

3.2.3 Model View Definition - MVD

Segundo buildingSMART (2015d), um *Model View Definition* (MVD), ou *IFC View Definition*, define um subconjunto do *schema* IFC, o qual é necessário para satisfazer um ou mais requerimentos de troca de informação na indústria AEC. O método usado e propagado pela buildingSMART para definir tais requerimentos é o *Information Delivery Manual*, IDM, que será tratado mais adiante neste trabalho.

Enquanto que o requerimento de troca de informações independe da versão do IFC, o MVD é dependente de cada versão de IFC lançada.

Atualmente, a versão mais usual e suportada pelas ferramentas computacionais envolvidas é a *IFC2x3 Coordination View Version 2.0* e, como indicado na nomenclatura, foi desenvolvido para a versão IFC2x3 exclusivamente.

O *Coordination View* foi a primeira *Model View Definition* (MVD) desenvolvida pela *buildingSMART International* e seu propósito principal é permitir o compartilhamento de modelos de informação na construção entre as disciplinas de arquitetura, engenharia estrutural e serviços da construção (mecânica). Esta MVD contém definições da estrutura espacial, edificação e elementos relacionados aos serviços da construção, as quais são necessárias para coordenar informações de projeto entre as disciplinas (GRAPHISOFT, 2015).

Além da *Coordination View* outras MVDs podem ser citadas brevemente, como por exemplo a *Basic FM Handover View* (versão 1.1 desenvolvida dentro do projeto aquário da *buildingSMART* chamado de “FM Handover View”) e a *Structural Analysis View* (versão 1.0 finalizada em 2008). A primeira tem como objetivo melhorar a interoperabilidade da informação do ciclo de vida da construção utilizando softwares disponíveis de planejamento BIM, projeto, construção e comissionamento, além de aplicativos de *Computer Aided Facility Management* (CAFM) e *Computerized Maintenance Management System* (CMMS) utilizados em gerenciamento de instalações para fase operacional. Já o *Structural Analysis View* define o modelo estrutural que pode ser trocado entre aplicativos de modelagem estrutural e de análise.

Model View Definitions (MVDs) são codificadas em um formato chamado MVDXML e definem valores permitidos para atributos particulares de alguns tipos de dados. Um exemplo fornecido por *buildingSMART* (2015d) é que a MVD pode exigir que uma parede forneça sua taxa de proteção ao fogo, que possua classificação de acordo com a tabela 22 de *OmniClass*⁷, além de conter informações necessárias para se realizar a análise estrutural, como o módulo de elasticidade dos materiais envolvidos.

O formato de *Model View Definition* para o IFC faz uso extensivo de basicamente duas idéias gerais: definições e configurações. Definições capturam uma gama de possibilidades, enquanto que as configurações capturam como estas possibilidades são utilizadas em um caso específico. Podem existir muitas diferentes configurações para qualquer definição, pois as mesmas possibilidades podem ser usadas de diferentes formas em diferentes situações (HIETANEN; FINAL, 2006).

⁷Sistema de classificação para a indústria da construção.

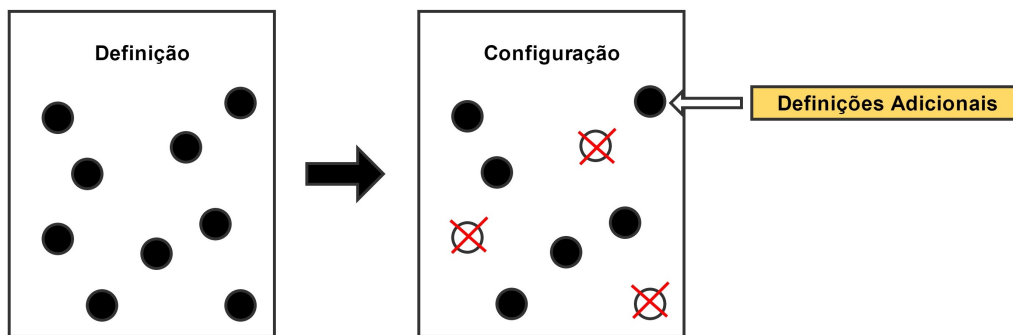


Figura 3.10: Definições e configurações

Uma configuração é utilizada para definir um subconjunto de possibilidades disponíveis e definir de maneira mais detalhada como este subconjunto é utilizado. Ou seja, as configurações de uma maneira geral reduzem o escopo, no entanto são mais específicas quanto a forma como o restante do escopo vai ser utilizado. Definições adicionais não devem entrar em conflito com a definição original e não é possível estender o escopo da definição original com o uso de configurações (HIETANEN; FINAL, 2006).

De acordo com Eastman et al. (2008), o MVD identifica o que deve ser esperado de uma troca de informações para ser efetiva, ajudando os usuários de ambos os lados da troca. O exportador sabe o que é requerido e também o que não é requerido, assim como o receptor tem o conhecimento do que vai ser recebido e com o que vai trabalhar.

Além de beneficiar os usuários, é importante citar que, do ponto de vista dos implementadores do IFC, o MVD é algo de extrema importância, já que define o que deve ser de fato implementado para que tanto a exportação quanto a importação estejam sincronizadas. A adoção de MVDs, então, reduz significativamente, se não completamente, o possível conflito de informações ocasionado pela redundância do conteúdo de um modelo IFC.

3.2.4

Information Delivery Manual - IDM

O *schema* IFC completo é desenvolvido como um conjunto de *schemas* de tópicos individuais. Cada *schema* de tópico representa uma idéia geral consistente (análise estrutural, por exemplo). Então, todos os *schemas* de tópicos são colocados juntos em um *schema* simples, o qual é a versão autorizada. Este contém centenas de entidades (classes) tipos de dados e conjuntos de propriedades (os componentes do IFC) (WIX; KARLSHOEJ, 2010, p. 11).

Esta grande quantidade de entidades presentes no schema IFC, as quais estão relacionadas às diferentes áreas dentro da indústria AEC, demonstram como o IFC foi projetado para suportar todos os requerimentos que possam existir dentro desse setor, em todos os estágios do projeto, ao longo de todo o ciclo de vida do empreendimento.

A completude do modelo IFC se torna relevante a partir do momento em que o projeto inteiro precisa ser compartilhado ou trocado entre organizações ou empresas. No entanto, esta não é a forma com que as informações de projeto normalmente são trocadas.

De uma maneira geral, as indústrias AEC possuem como característica a capacidade de juntar diferentes empresas e autoridades governamentais em uma organização específica para um projeto. Sendo assim, para que o trabalho seja eficiente, é necessário que todos os participantes da organização saibam quais os diferentes tipos de informação que devem trocar entre si e quando deve ocorrer esta troca.

As implementações IFC permitem aos usuários que acessem um modelo de dados BIM altamente estruturado e com objetos atribuídos a ele. Para que haja acesso ao modelo, uma interface de programação de aplicativos (API) é utilizada de acordo com seus direitos predefinidos. No entanto, o *schema IFC*, segundo Eastman e Sacks (2010, p. 02), não define os requisitos de intercâmbio de informação específicos para os diferentes estágios do projeto e, ainda, entre os diferentes atores envolvidos e aplicativos de software, o que dificulta o desenvolvimento de interfaces de software IFC úteis.

Ainda de acordo com Eastman e Sacks (2010), os desenvolvedores de software e usuários demandam guias de implementação de IFC que permitam focar em usar casos de interesse e garantir compatibilidade com outras implementações de software, garantindo assim uma melhor interoperabilidade.

O padrão ISO 29481-1:2010 “Building information modeling - Information delivery manual - Part 1: Methodology and format” foi desenvolvido pela *buildingSMART* com o intuito de criar uma metodologia para capturar e especificar processos e fluxo de informação durante o ciclo de vida do empreendimento (BUILDINGSMART, 2015c).

A metodologia pode ser usada para documentar processos já existentes ou novos, e descrever a informação associada que deve ser compartilhada entre as equipes de trabalho (BUILDINGSMART, 2015c).

Assim, o desenvolvimento de IDMs e MVDs tem sido uma iniciativa significativa para resolver o problema da interoperabilidade entre os diferentes softwares das mais variadas disciplinas inseridas em um projeto, identificando o subconjunto do modelo de dados IFC necessário para suportar os processos

de negócio definidos pelo usuário.

Quanto à arquitetura técnica básica do IDM, esta pode ser representada como na Figura 3.11, a qual contém três camadas: mapeamento de processos, requisitos para intercâmbio de informações e partes funcionais.

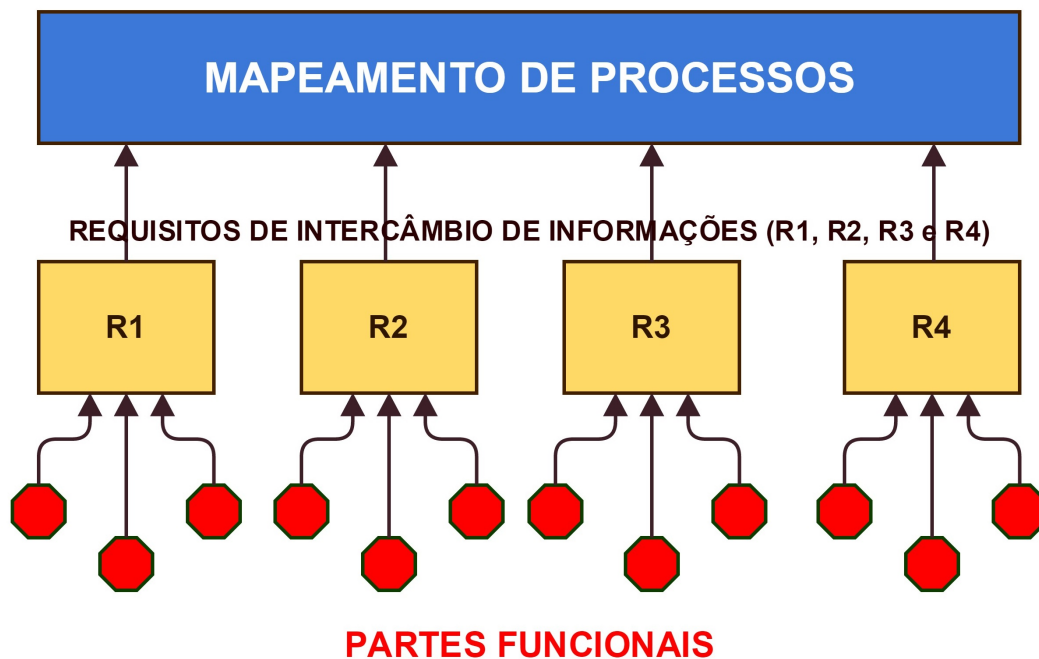


Figura 3.11: Arquitetura técnica básica do IDM, adaptado de Wix e Karlshoej (2010)

O mapeamento de processos descreve o fluxo de atividades dentro dos limites de um tópico particular. O propósito desse mapeamento é adquirir conhecimento da configuração das atividades a serem desenvolvidas, dos atores envolvidos, da informação requerida, consumida e produzida (WIX; KARLSHOEJ, 2010).

Já os requisitos para intercâmbio fazem parte de um conjunto de informações que necessitam ser trocadas para dar suporte aos requerimentos de um empreendimento em particular e do estágio de um projeto.

Na camada mais baixa do esquema apresentado na Figura 3.11 há a presença das partes funcionais, que nada mais são do que uma unidade de informação, ou uma simples idéia de informação, usada por desenvolvedores de soluções computacionais para dar suporte aos requisitos para intercâmbio de informação. Uma parte funcional descreve a informação em relação ao que

o modelo padrão de informação, no qual esta é baseada, requer (ver Figura 3.12).

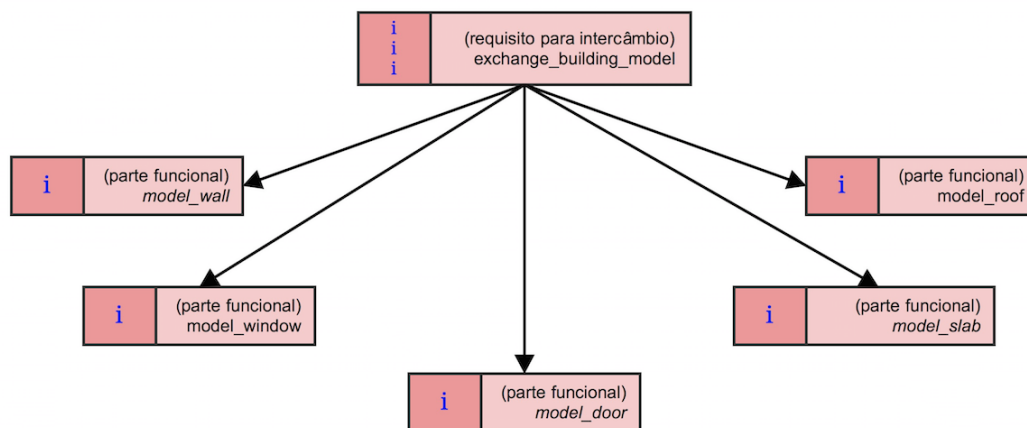


Figura 3.12: Partes funcionais em um requisito de intercâmbio de informações, adaptado de Wix e Karlsheoj (2010)

É importante frisar que para tornar uma IDM operacional esta deve ser devidamente suportada pelo software. O propósito principal do IDM é assegurar que os dados relevantes estão sendo passados de forma precisa e suficiente para as atividades a serem realizadas.

Um projeto de IDM deve, em princípio, desenvolver um MVD antes de continuar com a fase de implementação de software. Se o desenvolvimento do software é acompanhado de certificação, é possível controlar quando o software está atendendo o conjuntos de requisitos no IDM. O passo final é colocar o IDM em prática e validar os dados, como mostrado na Figura 3.13(KARLSHØJ, 2012).



Figura 3.13: Fluxo de registro de necessidades para implementação em ferramentas de software e certificação das ferramentas, adaptado de Buildingsmart (2011).

De um modo geral, o que se espera do IDM é a descrição dos processos do empreendimento com a especificação das informações dentro do ciclo de vida do projeto AEC para tornar possível se beneficiar completamente de uma significativa melhora no processo e do compartilhamento de informações.

3.2.5 International Framework for Dictionaries - IFD

Com o objetivo de melhorar cada vez mais o fluxo de trabalho, buscando aprimorar a capacidade das ferramentas computacionais interoperarem, três fatores básicos devem ser levados em consideração (Figura 3.14): o formato para a troca de informação (IFC), uma especificação de que informação deve ser trocada e quando deve ser realizada essa troca (IDM), e um entendimento padronizado do que a informação que está sendo passada realmente representa (IFD).

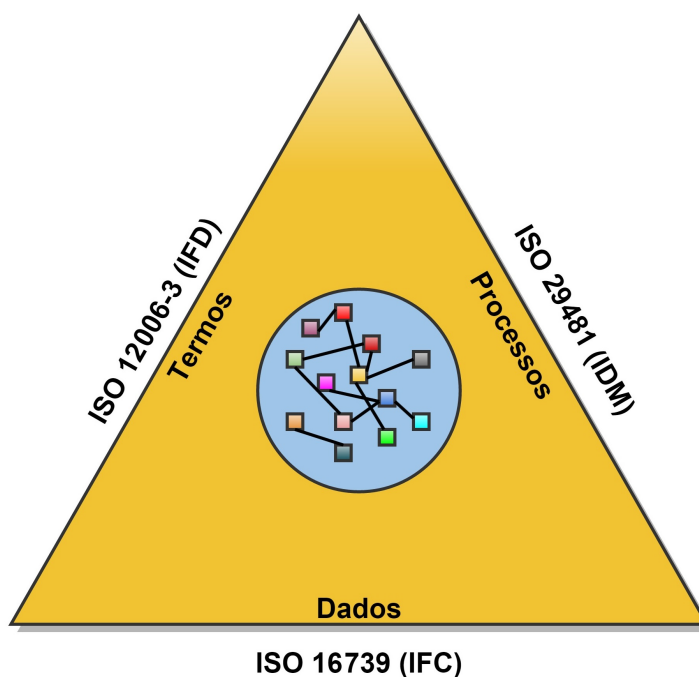


Figura 3.14: Pirâmide com os três conceitos básicos do OPEN BIM, adaptado de Buildingsmart (2015a).

Constatando que o IFC e o IDM já foram devidamente abordados neste trabalho, neste subcapítulo será tratado um pouco sobre o IFD.

O *International Framework for Dictionaries* (IFD), segundo Bell e Bjørkhaug (2006), é uma biblioteca aberta, onde conceitos e termos são semanticamente descritos e lhes é fornecido um número único de identificação. Isto permite que toda a informação contida no formato IFC seja rotulada com um *Globally Unique ID* (GUID).

Um exemplo que pode ser fornecido para o melhor entendimento do leitor é imaginar uma situação em que um arquiteto, quando fornece informações sobre o tipo dos materiais das vigas e colunas, utiliza uma *string* de texto simples. No entanto, mesmo que este profissional escreva de maneira correta,

não há garantia de que o aplicativo que receberá a informação em sequência no fluxo de trabalho entenderá o que exatamente este *string* de texto significa (BELL; BJØRKHAUG, 2006). E se, por ventura, for utilizada uma outra linguagem, dialeto ou o plural de alguma palavra? É neste momento que se faz necessário o uso do IFD.

Enquanto o padrão IFC descreve objetos, como eles estão relacionados e como a informação deve ser trocada e armazenada, o padrão IFD se preocupa unicamente em descrever o que os objetos são e que partes, propriedades, unidades e valores eles podem ter. O padrão IFD (ISO 12006-3) fornece dicionários, as definições dos conceitos e, assim, o entendimento comum necessário para a comunicação fluir da melhor forma possível (BJØRKHAUG, 2003 apud BELL; BJØRKHAUG, 2006).

O IFD, além de atribuir a cada conceito um GUID, também atribui a estes um conjunto de nomes e definições em diferentes idiomas. Entretanto, um nome pode também estar referido a múltiplos conceitos dentro da biblioteca. Enquanto o identificador é utilizado para coletar informações, os nomes e definições são usados meramente para comunicar a idéia para o usuário final. O IFD também permite classificar e estruturar informações de múltiplas maneiras. Na teoria todos os sistemas de classificação existentes podem ser recriados e interconectados dentro de uma biblioteca comum com o uso do IFD.

As versões mais recentes do IFC contém meios de trocar referências com uma biblioteca externa baseada em IFD para todas as informações relevantes (BELL; BJØRKHAUG, 2006).

Mas afinal como ocorre a integração entre o formato IFC e o IFD? Para que haja conexão entre estes, a informação formatada para o IFC normalmente possui uma instância da entidade *IfcLibraryInformation* visando identificar a biblioteca externa utilizada para referência. Sendo assim, cada entidade IFC contendo informação relevante, tipicamente *strings* de textos livres como o nome do material, a qual precisaria ser reconhecida por um computador, na verdade vai conter o GUID atual, identificando esta na biblioteca, além de conter uma referência para a *IfcLibraryInformation* para que a biblioteca, da qual o GUID é originado, possa ser identificada.

A adoção do IFD como biblioteca de referência para a informação contida no IFC, baseado no IDM, permite, assim, uma padronização da informação a ser trocada, já que todos os *strings* passam a ser etiquetados com um ID referenciando um conceito em uma “framework” de ontologias. Isso facilita substancialmente a implementação do IFC em softwares e melhora consideravelmente a comunicação e interoperabilidade entre os diferentes aplicativos.