



Representação do Conhecimento de Projeto de Produto utilizando o Paradigma da Orientação a Objetos

Design Knowledge Representation using Object-Oriented Paradigm

BARROS, Alexandre Monteiro de; Dr.; PUCRS

alexandre.barros@pucrs.br

SILVA, Régio Pierre da; Dr.; UFRGS

regio@ufrgs.br

TEIXEIRA, Fábio Gonçalves; Dr.; UFRGS

fabioigt@ufrgs.br

Resumo

Elaborar soluções criativas e eficientes requer que conhecimento de projeto seja representado em um alto nível de abstração durante a fase conceitual. Nesta fase, o conhecimento de projeto ainda é pouco preciso e fragmentado, com poucas opções de ferramentas computacionais que ofereçam suporte para sua representação. A orientação a objetos é um paradigma baseado na abstração, em que os elementos do mundo real são representados através de uma linguagem simples, acessível e reutilizável, que pode ser aplicada para o projeto de produto. Este artigo é parte de uma tese de doutorado e apresenta um *framework* para representar o conhecimento de projeto de produto utilizando o paradigma da orientação a objetos. A aplicação do *framework* produz um modelo conceitual que concentra parte do conhecimento de projeto, servindo como uma estrutura geral que conecta e relaciona diferentes blocos de informação associados aos produtos e sistemas que estão sendo desenvolvidos, contribuindo para uma análise apurada do projeto.

Palavras Chave: conhecimento em design; representação do conhecimento; orientação a objetos.

Abstract

Design creative and efficient solutions require that the representation of project knowledge use a high level of abstraction in the conceptual phase. In the conceptual design project knowledge is unclear and fragmented, with a few options of computational tools that support its representation. The Object-orientated paradigm is based on abstraction, in which real-world elements are represented through a simple, accessible and reusable language that can be applied to product design. This article is part of a doctoral thesis and presents a framework to represent the design knowledge using the object-oriented paradigm. The application of the framework produces a conceptual model that concentrates part of the project knowledge, serving as a general structure that connects and relates different blocks of information associated with the products and systems that are being developed, contributing to an accurate analysis of the project.

Keywords: design knowledge; knowledge representation; object-oriented.



1 Contextualização

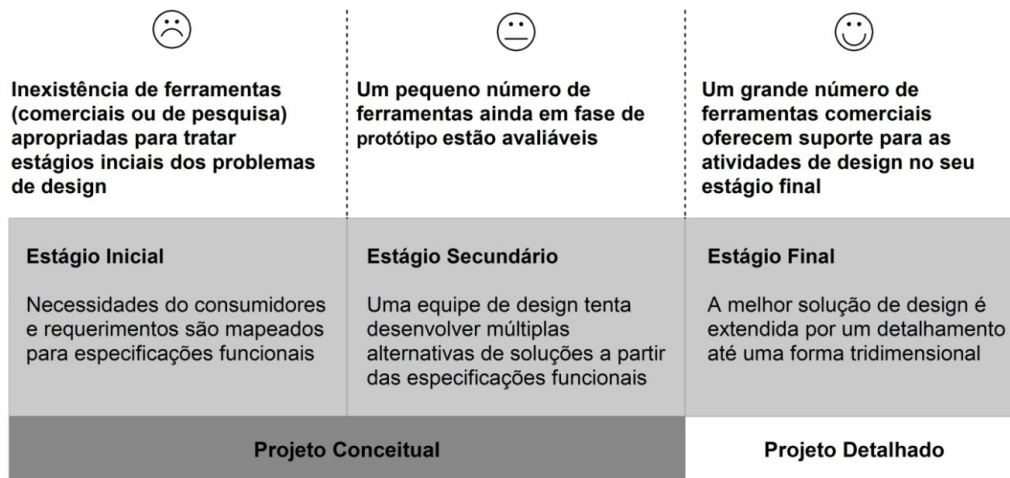
O conhecimento utilizado para o projeto de produtos e sistemas técnicos é em grande parte representado através de modelos geométricos tridimensionais, porém estes modelos transmitem pouca informação além da geometria e dos dados associados a ela. O domínio do problema de design, as características dos usuários e a tomada de decisão são elementos importantes do processo de projeto, porém por se tratar de informação não geométrica normalmente são pouco representados. Conforme Geyer (2012), os modelos digitais atuais são insuficientes em representar interdependências multidisciplinares complexas, tendo seu foco nas dependências geométricas. Para Ulrich (2011), o compartilhamento do conhecimento sobre artefatos físicos requer, além da comunicação de sua informação geométrica, a comunicação de muitos outros tipos de informação, como a especificação dos seus materiais e dos seus processos de produção.

Balmelli (2007) observa que nos sistemas computacionais atuais existe uma falta de suporte eficiente para a representação do conhecimento na fase de projeto conceitual, onde a arquitetura funcional e muitas vezes a arquitetura física de um produto ou sistema é decidida. Na fase de projeto conceitual as necessidades dos consumidores são transformadas em funções do produto e em casos de uso para, posteriormente, serem desenvolvidos por disciplinas específicas. A falta de representações formais para o projeto conceitual resulta em habilidades inadequadas para a tomada de decisão em nível de sistemas para o produto, como nos estudos de viabilidade. Igualmente, gera dificuldades de comunicação e interpretação para a equipe de projeto, o que não contribui para uma maior integração e colaboração no processo de design (BALMELLI, 2007).

O projeto conceitual é uma fase importante para o desenvolvimento de produtos e sistemas novos ou para o redesenho de produtos existentes. O adequado desenvolvimento conceitual para um produto ou sistema é importante porque as decisões tomadas nesta fase determinam as principais características de um produto ou sistema. O impacto das decisões realizadas nas fases iniciais de projeto é muito alto, gradualmente decaindo ao longo do processo de projeto, ao contrário do seu custo de implementação, que inicialmente é baixo, mas nas fases finais de projeto é elevado quando se é necessário inserir uma modificação no produto (WANG et al., 2002).

Pipes (2007) argumenta que em muitos sistemas de desenho assistido por computador (Computer Aided Design- CAD) é necessário que os designers saibam exatamente o que querem fazer antes de utilizar estes softwares. Esta condição pode ser um problema para as fases iniciais de projeto, que envolvem um alto grau de abstração e quando não se possui informações precisas, como ângulos, dimensões e tolerâncias para modelar o produto. Conforme Pipes (2007), o paradoxo é que a maioria dos sistemas atuais precisa de toda a informação do produto acabado antes que o designer possa iniciar a modelagem geométrica. Wang (2002) observa que os computadores têm sido extensivamente utilizados para simulações, análise e otimização, porém para o projeto conceitual existem poucas opções de ferramentas computacionais, pois o conhecimento necessário nesta fase é pouco preciso e fragmentado, o que o torna difícil de ser representado, conforme apresentado na figura 01.

Figura 1: Disponibilidade de ferramentas computacionais nas fases de design.



Fonte: Adaptado de Wang et al. (2002).

Liverani, Amati e Pellicieri (2004) reforçam a falta de ferramentas de software para o design conceitual, sugerindo que novas abordagens sejam mais focadas em tratar outras formas de conhecimento, ao contrário do que ocorre atualmente, em que os softwares são excessivamente dedicados a geometria. Salustri, Eng e Weerasinghe (2008) sugerem que para o projeto conceitual são necessários tipos de representação que suportem informação não geométrica e qualitativa com alto grau de abstração.

O paradigma da orientação a objetos foi elaborado no domínio da engenharia de software, mas sua aplicação pode ser expandida para outras áreas, pois seus conceitos e suas técnicas permitem a modelagem tanto de soluções digitais como de soluções físicas. Um amplo potencial de aplicação do paradigma da orientação a objetos para o design de produto, a arquitetura e o urbanismo pode ser observado considerando sua capacidade de representar o conhecimento de forma abstrata (BARROS, SILVA e TEIXEIRA, 2014).

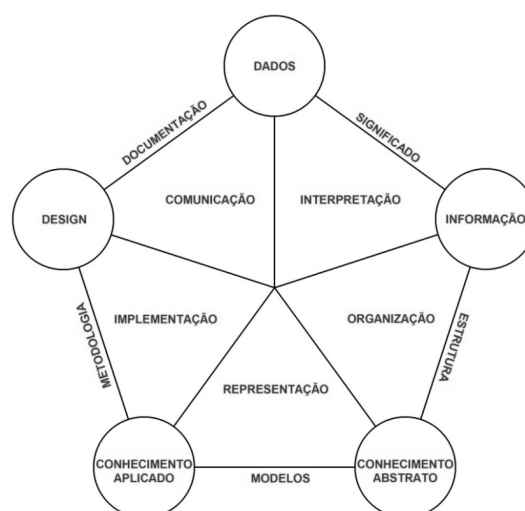
Para Schodek (2005) a modelagem orientada a objeto compromete-se a capturar não somente uma determinada geometria, mas também outros tipos de informação sobre o projeto por incluir conteúdo semântico além da representação digital de formas. Além disso, o paradigma da orientação a objetos está diretamente associado com padrões de projeto que permitem a reutilização do conhecimento anteriormente gerado (BOOCH, 2007). Apesar destas capacidades, o paradigma da orientação a objetos ainda é pouco aplicado nas fases iniciais de projeto de produtos e sistemas técnicos pelos profissionais de design. Entretanto, Salustri, Eng e Weerasinghe (2008) observam que existe uma grande oportunidade para ampliar a capacidade dos designers de pensarem tanto crítica como criativamente através de diagramas desde as fases iniciais de design e o paradigma da orientação a objetos pode ser apropriado para isto. Neste contexto, esta pesquisa propõe um *framework* para representar o conhecimento de projeto de produto aplicando o paradigma da orientação a objetos e assim oferecer suporte para a representação do conhecimento de design desde as fases iniciais de projeto.

2 Conhecimento em Design

A definição de conhecimento no design de produto depende do contexto em que for elaborada. Diferentes equipes de projeto podem fornecer definições distintas sobre o conhecimento para um mesmo produto, assim como uma mesma equipe pode fornecer definições diferenciadas dependendo do produto analisado. O conhecimento não está diretamente disponível, mas é obtido pela interpretação da informação deduzida através da análise de dados. Dados são disponibilizados de diferentes formas e sua interpretação, abstração ou associação conduz para a geração de informação. O conhecimento é obtido pela experimentação e aprendizado da informação que é colocada em ação. O conhecimento também pode ser descrito como a experiência, os conceitos, os valores e os modos de trabalho que podem ser compartilhados e comunicados. Em termos de design, o conhecimento é o entendimento de uma informação dada, como seu conteúdo, sua origem e a sua aplicabilidade, observando que não são apenas as regras que o designer utiliza durante o projeto, mas sim o histórico que possibilita entender e rever estas regras (CHANDRASEGARAM et al, 2013).

Para Raphael e Smith (2003) o conhecimento de projeto pode ser compreendido como um conjunto de dados que estão correlacionados entre si. Estes autores entendem que dados e conhecimento são subclasses da informação, pois o conhecimento é um tipo de informação em conjunto com os dados. Eilouti (2009) observa que o conhecimento em design pode ser compreendido como um conjunto de dados que evolui até se tornar o projeto de um produto ou sistema. Inicialmente, os dados devem ser processados e interpretados para formar peças de informação com significado. Estas peças de informação devem ser classificadas e organizadas por estruturas claras formando assim novas peças de conhecimento abstrato. O conhecimento abstrato necessita de representação na forma de protótipos estruturados ou modelos para então ser transformado em um conhecimento possível de ser aplicável. Através de uma metodologia apropriada, o conhecimento aplicável pode então ser utilizado para o projeto de novos artefatos (Figura 2).

Figura 2: Pentágono do conhecimento.



Fonte: Eilouti (2009).



2.1 Conhecimento Precedente

Para cada artefato físico elaborado através de um processo de projeto está atribuído um conhecimento em design. Deste modo, soluções prévias de design podem ser exploradas e servir como fonte de conhecimento para o desenvolvimento de novos produtos e sistemas. Design precedente é um termo associado ao conhecimento que está vinculado a um design existente (PASMAN, 2003). O termo design precedente é empregado para descrever um design específico em que conceitos e ideias definem blocos de conhecimento. O conhecimento depende da estrutura destes blocos de informação e de sua representação. Um problema na representação de soluções prévias de design é a riqueza e a complexidade de seu conteúdo descritivo. Cada design contém muitas peças de informação que são difíceis de descrever ou decompor, e sua representação não engloba toda a informação embarcada. O conhecimento prévio pode ser considerado como uma fonte significativa de conhecimento para o processo criativo de design. Muitas ideias relevantes podem ser extraídas de projetos passados (OXMAN, 1994).

Oxman (1994) sugere que o desenvolvimento de sistemas computacionais que possam auxiliar o designer no processo de projeto utilizando precedentes seja baseado em uma memória organizada. Nestes sistemas, a habilidade de codificar, procurar e extrair um conhecimento relevante de design é extremamente importante. Tradicionalmente, a estrutura organizacional do conhecimento precedente é feita através de categorias como nome, período histórico, estilo ou localização que servem como um indexador da informação. Entretanto, Oxman (1994) observa a necessidade de a memória organizacional refletir um determinado modelo cognitivo de design de modo a possibilitar a associação entre os conceitos do problema abordado e os conceitos provenientes dos precedentes.

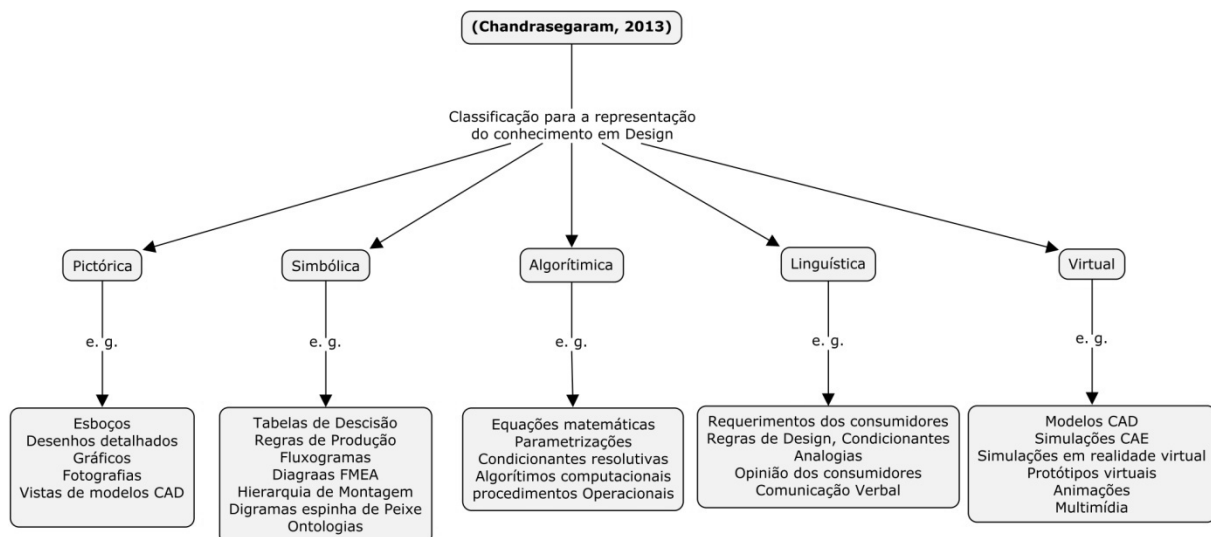
Bohm et al (2008) propõe um esquema de dados para capturar elementos fundamentais da informação de design em um repositório heterogêneo para então serem reutilizados em novos projetos. Estes elementos são divididos em sete categorias: (I) artefato, (II) função, (III) falha, (IV) configuração física, (V) desempenho, (VI) sensorial e (VII) mídia. Conforme estes autores, estas categorias podem resultar em uma descrição completa do conhecimento fundamental de um design. Estas categorias podem ser recriadas ou especializadas para outras aplicações mantendo o esquema de dados proposto.

O conhecimento da configuração física de um artefato ou sistema está relacionado com os aspectos de sua forma e é organizado em quatro categorias: informação de manufatura, material, dimensões geométricas delimitadoras e cor. O conhecimento sobre o artefato serve como um ponto central que engloba as demais categorias de dados. Um artefato pode ser desde um produto completo, uma montagem parcial ou apenas uma parte individual. Nesta categoria é possível estabelecer relações de pai e filho e assim obter uma hierarquia do produto, além de associá-lo um determinado sistema. O conhecimento sobre função é utilizado para fornecer porções de modelos funcionais associados a um artefato. A captura do fluxo de material, energia e sinal em um produto requerem a determinação de entradas e saídas entre fluxo e artefato para cada função. A informação das categorias é representada no formato de tabelas de banco de dados. Os dados são tabulados e indica relações entre os elementos das diferentes categorias, o que permite a visualização de maneira integrada de um subconjunto da informação relativa a um aspecto do artefato. As tabelas da base de dados podem armazenar informação sobre o produto de maneira direta ou referenciar informação de maneira indireta proveniente de outras tabelas (BOHM, 2008).

3 Representação do Conhecimento

Para Chandrasegaram et al. (2013) o conhecimento pode ser visto como uma informação em um determinado contexto e sua representação pode ser classificada como: (I) Pictórica e.g. esboços, desenhos detalhados, fotografias, gráficos e vistas de modelos CAD; (II) Simbólica e.g. quadros de decisão, regras de produção, diagramas, árvores de montagem, ontologias; (III) Linguística, e.g. requisitos dos consumidores, regras de design, condicionantes, analogias, comunicação verbal; (IV) Virtual e.g. modelos CAD, simulações CAE, simulações de realidade virtual, protótipos virtuais, animações e multimídia; e (V) Algorítmica e.g. equações matemáticas, parametrizações, algoritmos computacionais e procedimentos operacionais (Figura 3). Várias representações podem ser enquadradas em mais de uma categoria, como no caso dos gráficos de montagem, que podem ser considerados representações algorítmicas ou pictóricas (CHANDRASEGARAM et al, 2013).

Figura 3: Tipos de Representação.



Fonte: Baseado em Chandrasegaram (2013).

Observa-se que não existe um limite claro e rígido que demarque os diferentes tipos de representação, pois múltiplas dimensões devem ser consideradas para sua classificação. Uma representação pode utilizar diferentes níveis de abstração, variando desde uma representação abstrata até uma representação concreta da informação. Cada representação está associada a uma linguagem para descrever um determinado modelo da realidade. Diferentes modelos expressam visões complementares sobre um produto ou sistema de acordo com a necessidade do projeto (Booch, Rumbaugh e Jacobson, 2006).



3.1 Diagramas

Diagramas são representações externas próximas da estrutura cognitiva utilizada no raciocínio humano para a resolução de problemas. Conforme Larkin e Simon (1987) representações diagramáticas preservam explicitamente a informação sobre as relações geométricas e topológicas entre os componentes do problema, diferentemente das sentenciasais, que preservam relações temporais ou lógicas por exemplo. Para compreender as vantagens e a adequada aplicação do uso de diagramas, Larkin e Simon (1987) investigaram representações diagramáticas e representações não diagramáticas em um sistema de processamento da informação. Uma estrutura de dados que se apresenta em uma única sequência é uma representação sentencial. Uma estrutura de dados em que a informação é indexada em duas dimensões é uma representação diagramática. Na representação sentencial, não diagramática, expressões descrevem um problema através de uma linguagem natural que pode ser transferida para uma linguagem formal simples, formada por sentenças e orientada uma a uma. Na representação diagramática, as expressões correspondem aos componentes que descrevem um problema, igualmente orientados um a um. Cada expressão contém sua informação armazenada em um particular ponto do diagrama, incluindo a informação sobre suas relações com outros pontos adjacentes.

Diagramas do tipo nó e ligação utilizam figuras geométricas para itens, atividades e conceitos e ligações através de linhas que estabelecem as relações entre estas entidades. Mapas cognitivos, mapas mentais, mapas conceituais, modelos de relações entre entidades, fluxogramas, mapas argumentativos, redes semânticas, mapas de conhecimento e diagramas UML são exemplos de métodos sistemáticos de representação diagramática que relacionam diferentes itens enquadrados por figuras através de ligações que podem ou não ser rotuladas com regras sequenciais explícitas (EPPLER, 2006).

Kokotovich (2008) demonstra que os mapas mentais não hierárquicos também são ferramentas úteis para o design, especialmente para as fases iniciais de design, quando é necessário analisar o problema de projeto. Este método contribui significativamente para a compreensão da informação de projeto por parte de estudantes e profissionais, pois permite documentar conhecimento sobre um determinado domínio de maneira visual. Kokotovich (2008) sugere que existe uma clara relação entre o uso de mapas mentais não hierárquicos e a habilidade dos designers de produzirem projetos criativos e bem fundamentados. Isto ocorre porque eles possibilitam uma clara articulação de complexas inter-relações dinâmicas entre as questões de projeto nas fases iniciais do processo. Ademais, a natureza gráfica dos mapas mentais permite a compreensão da natureza e estrutura do problema de projeto e possibilita uma abordagem holística para sua solução. Também é possível integrar informação, ideias e questões novas e antigas, assim como construir um corpo de conhecimento relatado a um determinado domínio, funcionando com um auxílio da memória para os designers (KOKOTOVICH, 2008).



4 Paradigma da Orientação a Objetos

Gabbar, Suzuki e Shimada (2003) observam que uma abordagem orientada a objetos permite a representação do mundo real de maneira natural, onde os sistemas são representados e analisados a partir de modelos estruturais, comportamentais e funcionais. O paradigma da orientação a objetos consiste em distribuir informação, conhecimento e relações sobre objetos como parte de uma estrutura modular. Design orientado a objetos é um método de projeto que aplica a decomposição do conhecimento sobre um determinado domínio ou sistema utilizando representação de forma abstrata (VAN DER PLOEG, 2013). La Rocca (2012) expõe que nas engenharias os conceitos da orientação a objetos já são amplamente aplicados em abordagens consolidadas, como a engenharia de sistemas baseada em modelos e a engenharia baseada em conhecimento.

Schodek (2005) indica que o design orientado a objetos utiliza um modelo de dados para descrever informação complexa através de componentes discretos, porém inter-relacionados. Estas unidades de dados, os objetos, contêm tanto atributos como métodos. Atributos são propriedades, como por exemplo, a largura de uma determinada porta, e métodos são partes de um código de programação que contém descrições operativas ou comportamentais. Objetos também podem conter um conjunto de mensagens e podem responder a elas. Mensagens são requisições entre objetos para executar um método em particular, como ajustar a espessura de um objeto parede para coincidir com a espessura de um objeto porta. Os objetos que compartilham os mesmos métodos, respondem as mesmas mensagens e possuem os mesmos atributos são agrupados em classes. Um objeto “porta de correr” pode ser uma instância de uma classe denominada “porta”, que por sua vez pode conter outras instâncias, como “porta de abrir”. Estes objetos podem compartilhar certos atributos como “espessura” e “altura” e pode ser diferenciado pelos valores destes atributos. O mecanismo de herança consiste na criação de novas classes contendo atributos e métodos de outras classes, o que permite o compartilhamento de informação e a mesma resposta a algoritmos particulares de projeto. Esta estrutura permite que os objetos sejam utilizados de diferentes maneiras pelo sistema ao mesmo tempo em que podem ser modificados ou ampliados sem afetar o modo como os objetos são usados. Os objetos podem ser construídos, reutilizados e refinados ao longo do tempo (SCHODEK, 2005).

Um modelo de classes é uma estrutura estática de representação da realidade visando formalizá-la. Na modelagem orientada a objetos, Starr (2001) utiliza a denominação modelo de classes ao invés de diagrama de classes (que normalmente é empregada na literatura sobre OO), pois um diagrama de classes é apenas uma representação diagramática de um conjunto de classes. Já um modelo de classes é uma estrutura independente de dados que pode ser representada de diferentes formas. Embora seja uma estrutura estática, um eficiente modelo de classes deve permitir a compreensão da dinâmica do sistema. O comportamento, a inteligência e a confiabilidade de um sistema devem responder a uma estrutura clara e não ambígua. Para um modelo de classes ser eficiente, ele deve expressar uma série de condições de um determinado sistema de maneira transparente e precisa, restringindo a uma interpretação unificada. Para iniciar a modelagem de um sistema, Starr (2001) sugere a definição de classes observando as propriedades dos objetos e sua interação com outros objetos do mundo real. Para identificar as propriedades relevantes de uma classe é necessário avaliar um determinado contexto. As propriedades relevantes são determinadas quando se analisa o contexto.

Por exemplo: no contexto de tráfego aéreo, todo avião possui um número de cauda; uma altitude, uma velocidade e uma posição. Aplicando a abstração, é possível definir uma classe para a categoria avião, conforme figura 4. Conforme Starr (2001) classes também podem ser representadas através de uma estrutura de dados independente em formato de tabela, sendo possível realizar operações de ordem matemática e lógica. Uma estrutura de dados tangível permite operações para manipulação dos dados através de relações álgebras bem definidas e que são suportadas por ações da UML.

Figura 4: Abstração de uma classe a partir de dados do mundo real.



Fonte: Adaptado de Starr (2001).

Conforme Schodek (2005) o que pode ser feito ou não por um sistema de design digital é definido pelas capacidades orientadas a objetos que ele possui. A capacidade depende de como o software está escrito e estruturado, qual linguagem de programação é usada, e qual o tipo de aplicação de uma base de dados armazena a informação que os usuários inserem durante o processo de design. Estes fatores estão relacionados com o grau das capacidades orientadas ao objeto os softwares possuem ou não. A habilidade para definir classes, criar instâncias de objetos e atribuir comportamentos somente é possível utilizando sofisticadas linguagens de programação orientadas a objetos. Softwares como CATIA, Pro/ENGINEER, SolidWorks, MicroStation e Revit possuem capacidades de orientação a objetos, embora não necessariamente são referidas nestes termos (SCHODEK, 2005).

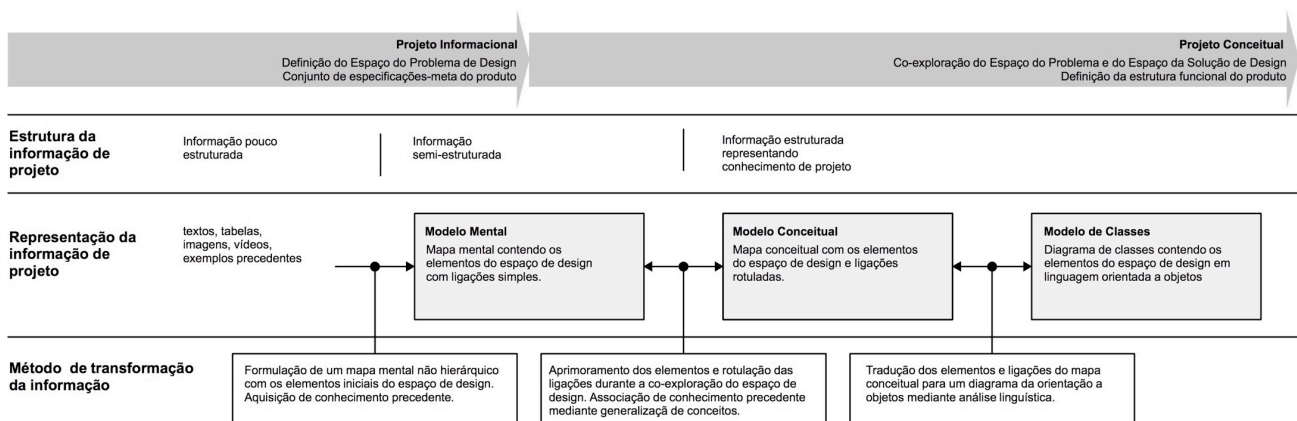
5 Framework para Representação do Conhecimento de Projeto de Produto

Para representar o conhecimento de design, é proposto um *framework* composto por um modelo de tratamento da informação e de um método de aplicação. O modelo consiste em estruturar a informação de projeto, desde elementos pouco ordenados e aprimorá-los até torná-los elementos formalmente ordenados que documentem adequadamente o conhecimento de projeto. A informação fornecida pela etapa de projeto informacional para a etapa de projeto conceitual é pouco precisa e fragmentada. Esta informação pode estar representada através de diferentes linguagens, como textos, tabelas e imagens, e ainda não possui uma estrutura clara e precisa, portanto, não formal. Assim, para o projeto conceitual, sugere-se partir de uma linguagem não formal e gradativamente estruturar a informação, manipulando os elementos constituintes do espaço de design até efetivamente produzir conhecimento de projeto. Para estruturar a informação, transformando-a desde uma linguagem não formal até uma linguagem formal, são sugeridos três modelos de representação que permitem manter uma correspondência entre os elementos de cada modelo de projeto: um modelo mental, um modelo conceitual e um modelo de classes.

O modelo mental é produzido a partir de uma linguagem não formal e constitui uma representação externa mais próxima do processo cognitivo empregado pelos designers durante o mapeamento do espaço de design, acomodando uma maior imprecisão dos elementos sem comprometer sua utilidade. Já um modelo conceitual é mais preciso que um modelo mental, restringindo sua interpretação e armazenando de maneira mais clara o conhecimento, pois possui conteúdo semântico. O modelo conceitual é uma linguagem semiformal, definindo uma ontologia para um determinado domínio a partir do contexto do projeto, porém ainda possui pouca capacidade computacional, ou seja, não é possível utilizar o modelo para fins de simulação, por exemplo. Por fim, o modelo de classes é uma estrutura estática e independente de dados com uma interpretação unificada e formal, com alta capacidade computacional, sendo possível implementar simulações e até mesmo gerar código dependendo do nível de detalhamento alcançado.

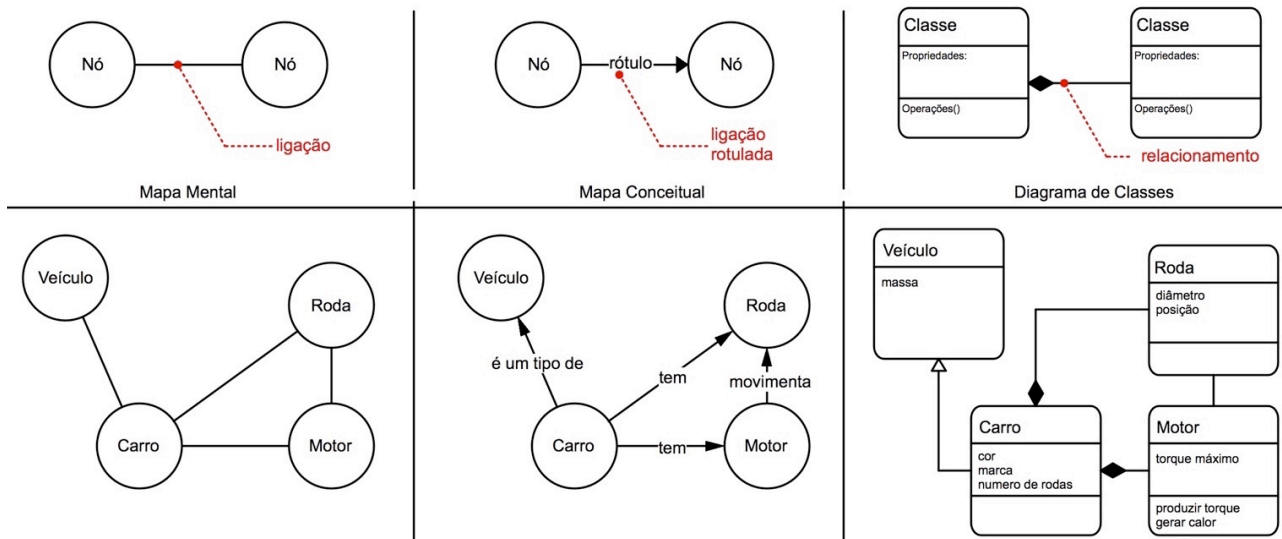
O método do *framework* estabelece os procedimentos (formulação de conceitos e ligações) e os mecanismos (rotulação e análise linguística) para operacionalizar os modelos de projeto. Para a formulação e a transformação destes modelos é necessário a: (I) elaboração de um modelo mental, no início do projeto conceitual, para capturar e documentar de maneira livre e não formal os elementos do espaço de design. Posteriormente, é necessário o (II) aprimoramento e o refinamento do modelo mental em um modelo conceitual, transformando, acrescentando, suprimindo ou reformulando os conceitos e rotulando suas ligações. Finalmente, é feita a (III) transformação dos elementos do modelo conceitual para um modelo de classes. O mecanismo que possibilita esta transformação é a análise linguística, de acordo com García Peñalvo e Pardo Aguilar (1998). A estruturação da informação corresponde ao gradual aperfeiçoamento dos elementos do espaço de design, mediante inserção, exclusão, modificação e rotulação das ligações, utilizando representação diagramática para produzir modelos adequados aos níveis de abstração requeridos pelo projeto durante a exploração do espaço de design. Modelos conceituais e modelos de classes estruturam e representam parte do conhecimento de projeto associado a um domínio em particular em um alto nível de abstração e assim são apropriados para a etapa de projeto conceitual. A figura 5 apresenta o desenho do *framework*. Já a figura 6 apresenta a linguagem de representação utilizada no *framework*.

Figura 5: Desenho do framework proposto.



Fonte: Elaborado pelo autor.

Figura 6: Linguagem de representação do framework proposto e exemplo de aplicação.



Fonte: Elaborado pelo autor.

6 Avaliação do Framework

Conforme Dresch, Lacerda e Junior (2015) a avaliação de um framework pode ser realizada através da instanciação, que é a aplicação dos artefatos que o compõe (constructos, modelos e métodos) em um ambiente real de uso. Nesta pesquisa, um estudo observacional foi conduzido mediante a realização de *workshops* de síntese (Figura X). Os *workshops* visavam simular uma situação real de projeto de produto para permitir a avaliação inicial do *framework* proposto por especialistas da área de design. Foram convidadas pessoas com experiência profissional e acadêmica, atuantes há mais de três anos na área de design de produto, residentes na região sul do Brasil a fim de viabilizar o encontro presencial. Participaram de sete voluntários. Todos os participantes fizeram parte do *workshop* de síntese e de sua posterior avaliação mediante o preenchimento de um questionário. Foram realizadas duas sessões de projeto com duração de duas horas e trinta minutos para facilitar a reunião dos participantes (Figura 7). O convite para a participação da pesquisa foi enviado via correio eletrônico com o encaminhamento do Termo de Consentimento Livre e Esclarecido e respectiva aprovação pelo Comitê de Ética em Pesquisa (CEP).

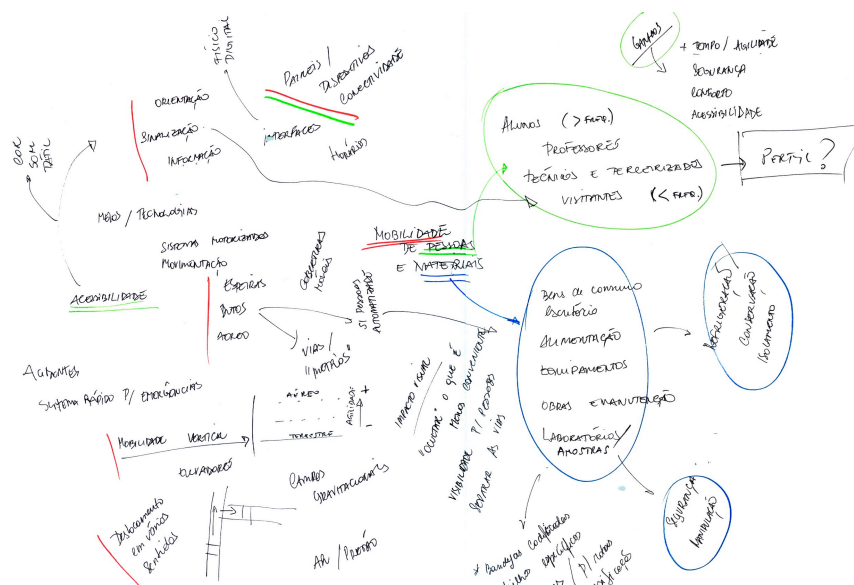
Figura 7: Workshop de projeto.



Fonte: Elaborado pelo autor.

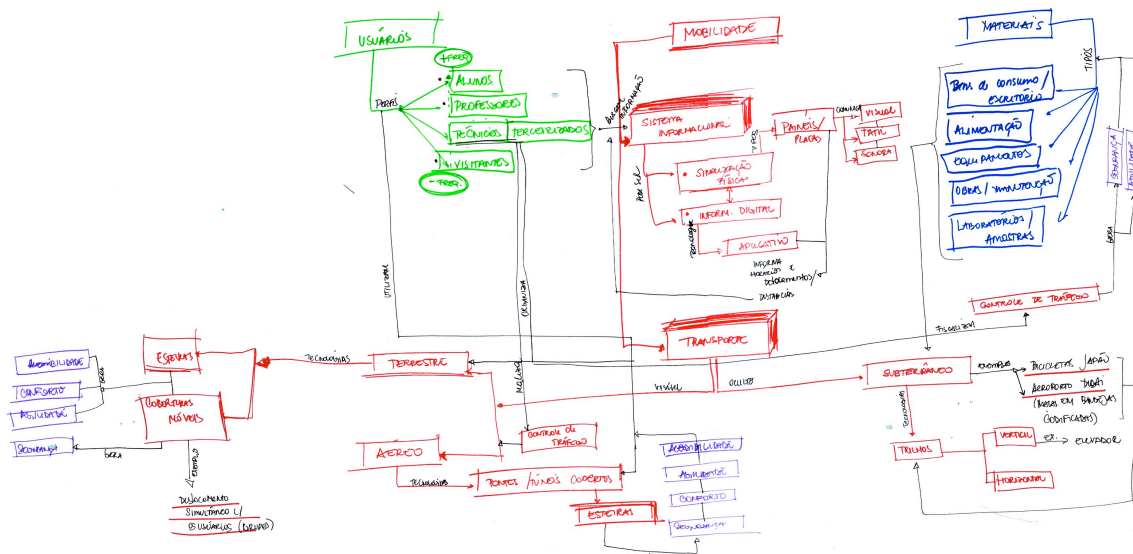
Os participantes foram solicitados a desenvolver um projeto de um produto/sistema técnico, simulando uma atividade comum na prática do design, para posteriormente responder um questionário de avaliação sobre o *framework* empregado. Em um primeiro momento o *framework* foi apresentado e instruções sobre sua aplicação foram dadas. Após isto, foi apresentado o tema de projeto e os participantes iniciaram o mapeamento dos elementos do espaço do problema de design (Figura 8). Posteriormente ao mapeamento dos elementos do espaço do problema de design, os especialistas iniciaram a geração de ideias para propor as soluções adequadas ao contexto por eles estabelecido. Finalmente, os especialistas formularam os mapas conceituais que representavam o modelo de classes da sua solução proposta (Figura 9).

Figura 8: Modelo mental elaborado pelos especialistas #4 e #5.



Fonte: Workshop de Síntese.

Figura 9: Modelo conceitual elaborado pelos especialistas #4 e #5.



Fonte: Workshop de Síntese.



Mediante o acompanhamento dos *workshops*, foi possível observar e constatar pontos positivos e pontos negativos que ocorreram ao longo do processo. Como as sessões foram registradas em vídeo, também foi possível revisar o material a fim de sanar eventuais dúvidas a respeito de determinadas questões. O conjunto das repostas das questões permitiram verificar a aplicabilidade do *framework* em relação a seus objetivos primários, e isto foi comprovado pela média dos resultados, com a maior parte dos especialistas considerando uma contribuição efetiva para oferecer suporte ao processo de projeto de produtos e sistemas.

7 Discussão

Atualmente, com o aumento da complexidade dos produtos e sistemas técnicos, o conhecimento sobre o domínio de projeto pode ser bastante amplo e diversificado, o que dificulta a tarefa de estruturar e representar a informação. As ferramentas computacionais disponíveis para o projeto de produto são altamente especializadas e eficazes para as fases finais do processo de projeto, porém as fases iniciais ainda carecem de suporte para auxiliar os designers neste sentido. Assim, identificou-se a possibilidade de sistematizar a representação do conhecimento de projeto de produto em um alto nível de abstração utilizando o paradigma da orientação a objetos.

Com a inserção do paradigma da orientação a objetos nas fases iniciais do desenvolvimento de produtos, espera-se estruturar e representar a informação para que ela possa evoluir de maneira contínua e armazenar conhecimento reutilizável sobre o projeto. A estrutura de dados do *framework* é tangível e permite a articulação de complexas inter-relações entre os elementos, contribuindo para uma análise apurada do projeto. Além disso, esta pesquisa unifica um conjunto de múltiplas representações que descrevem, sob diferentes aspectos, tanto os problemas de projeto como possíveis soluções para o mesmo, complementando a representação geométrica normalmente utilizada em um projeto de produto.

O *framework* adota uma linguagem de representação diagramática que pode evoluir desde um modelo mental, com elementos diversificados e pouco ordenados, até uma rede estruturada de classes e relacionamentos em um modelo de classes. Os mapas mentais são apropriados para representar informação fragmentada e pouco precisa, estabelecendo os elementos e as relações iniciais do espaço de design através de conceitos e ligações simples. Por sua vez, os modelos conceituais podem ser representados por meio dos mapas conceituais. Um mapa conceitual é um diagrama contendo elementos nodais e múltiplas ligações rotuladas que possuem direção. Neste caso, a ligação entre os conceitos pode ser lida em sequência e formar uma oração completa e facilmente interpretável. Cada nó em um mapa conceitual corresponde a um conceito. Múltiplos conceitos inter-relacionados definem uma ontologia para um domínio ou projeto em particular. A partir de uma ontologia pode ser produzido um modelo de classes. O modelo de classes concentra conhecimento sobre o projeto, servindo como uma estrutura geral que conecta e relaciona diferentes blocos de informação associados aos produtos e sistemas que estão sendo elaborados.

A avaliação por especialistas em design verificou que o *framework* possui a capacidade de representar o conhecimento útil para o projeto, mas não foi possível determinar com precisão se ele realmente auxilia a geração de soluções criativas, pois comprovou-se que ele é uma forma complementar de representação e precisa estar associado a outras técnicas e estratégias de design para assim auxiliar os designers a explorar soluções criativas no processo de projeto.



Ressalta-se que esta pesquisa concentra sua atenção entre a fase final do projeto informacional e o início da fase de projeto conceitual. Ela está baseada na representação simbólica diagramática dos elementos de projeto, porém outros tipos de representação são necessários para o desenvolvimento de produtos e sistemas técnicos, conforme já mencionado anteriormente. Não é proposto que a representação aqui utilizada sobreponha ou substitua os outros tipos de representação associados ao processo de design, mas sim que ela complemente e se some a eles para assim descrever e especificar uma solução de design de maneira mais completa e precisa, podendo ser utilizada em conjunto com os métodos tradicionais de design. Como proposta para continuidade desta pesquisa, espera-se verificar a aplicabilidade do *framework* em projetos mais amplos, tanto por estudantes e professores em ateliês de projeto de produto, como por profissionais de design.

8 Referências

- BALMELLI, Laurent et al. An overview of the systems modeling language for products and systems development. **Journal of Object Technology**, v. 6, n. 6, p. 149-177, 2007.
- BARROS, Alexandre; SILVA, Régio; TEIXEIRA, Fábio. Processo de Projeto de Produto a partir do Paradigma da Orientação a Objetos. **Estudos em Design**, v. 22, n. 1, 2014.
- BOHM, Matt R. et al. Introduction of a data schema to support a design repository. **Computer-Aided Design**, v. 40, n. 7, p. 801-811, 2008.
- BOOCH, Grady et al. **Object-Oriented Analysis and Design with Applications**. 2007.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Elsevier Brasil, 2006.
- BÜRDEK, Bernhard E.; VAN CAMP, Freddy. **Design: história, teoria e prática do design de produtos**. Editora Edgard Blücher, 2006.
- CHANDRASEGARAN, Senthil K. et al. The evolution, challenges, and future of knowledge representation in product design systems. **Computer-aided design**, v. 45, n. 2, p. 204-228, 2013.
- DRESCH, Aline; LACERDA, Daniel Pacheco; JÚNIOR, José Antonio Valle Antunes. **Design science research: método de pesquisa para avanço da ciência e tecnologia**. Bookman Editora, 2015.
- EILOUTI, Buthayna Hasan. Design knowledge recycling using precedent-based analysis and synthesis models. **Design Studies**, v. 30, n. 4, p. 340-368, 2009.
- EPPLER, Martin J. Making knowledge visible through intranet knowledge maps: concepts, elements, cases. **In: System Sciences, 2001. Proceedings... 34th Annual Hawaii International Conference on. IEEE, 2001. p. 9 pp.**
- GABBAR, Hossam A.; SUZUKI, Kazuhiko; SHIMADA, Yukiyasu. Plant object-orientated model formalization—case study: HDS plant design. **Design Studies**, v. 24, n. 1, p. 101-108, 2003.
- GARCÍA-PEÑALVO, Francisco José; PARDO AGUILAR, Carlos. **Diagramas de Classe en UML 1.1**. 1998.
- GEYER, Philipp. Systems Modeling for Building Design: A Method based on the Systems Modeling Language. **In: Proceedings of the 2011 eg-ice Workshop, University of Twente, The Netherlands. 2011.**
- KOKOTOVICH, Vasilije. Problem analysis and thinking tools: an empirical study of non-hierarchical mind mapping. **Design Studies**, v. 29, n. 1, p. 49-69, 2008.



- LA ROCCA, Gianfranco. Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. **Advanced engineering informatics**, v. 26, n. 2, p. 159-179, 2012.
- LARKIN, J. H; SIMON, H. A. Why a Diagram is (Sometimes) Worth Ten Thousand Words. 1987. **Cognitive Science**. V. 11. Pg. 65-99.
- LIVERANI, A .; AMATI, G .; PELLICCIARI, M. "SYMBOLIC DESIGN": A UML BASED ENVIRONMENT FOR INTEGRATED PRODUCT DEVELOPMENT. In: DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia. 2004.
- OXMAN, Rivka E. Precedents in design: a computational model for the organization of precedent knowledge. **Design studies**, v. 15, n. 2, p. 141-157, 1994.
- OXMAN, Rivka E. Think-maps: teaching design thinking in design education. **Design studies**, v. 25, n. 1, p. 63-91, 2004.
- PASMAN, G. **Designing with precedents**. 2003. Tese de Doutorado. TU Delft, Delft University of Technology.
- PIPES, Alan. **Drawing for Designers**: drawing skills, concept sketches, computer systems, illustrations, tools and materials, presentations, production techniques. Great Britain, 2007.
- RAPHAEL, Benny; SMITH, Ian FC. Fundamentals of computer aided engineering. John Wiley & Sons, 2003.
- SALUSTRI, Filippo A.; ENG, Nathan L.; WEERASINGHE, Janaka S. Visualizing information in the early stages of engineering design. **Computer-Aided Design and Applications**, v. 5, n. 5, p. 697-714, 2008.
- SCHODEK, D. et al. **Digital Design and Manufacturing**: CAD/CAM Technologies in architecture and Design. 2005. ISBN: 0-471-45636-5.
- STARR, Leon. **Executable UML**: How to Build Class Models. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- ULRICH, K. T. **Design**: Creation of artifacts in society. University of Pennsylvania. ISBN 978-0-9836487-0-3, 2011.
- VAN DER PLOEG, J. C. **A knowledge-based framework for structural optimization**: An object-oriented approach for reuse of explicit knowledge in computational optimisation of steel structures. 2013. Tese de Doutorado. TU Delft, Delft University of Technology.
- WANG, Lihui et al. Collaborative conceptual design—state of the art and future trends. **Computer-Aided Design**, v. 34, n. 13, p. 981-996, 2002.