

Yeni İmal Edilecek Uçak Motor Parçaları Üretim Sürecinin Tasarlanması için

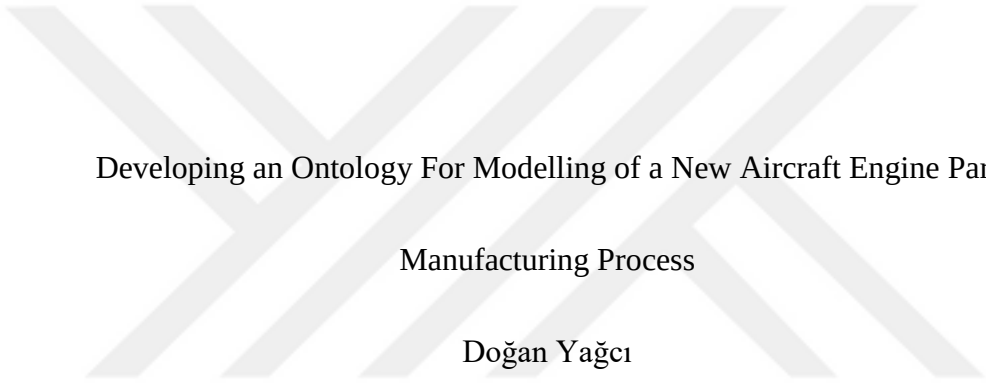
Ontoloji Geliştirilmesi

Doğan Yağcı

YÜKSEK LİSANS TEZİ

Havacılık Bilimi ve Teknolojileri Anabilim Dalı

Aralık 2019



Developing an Ontology For Modelling of a New Aircraft Engine Part

Manufacturing Process

Doğan Yağcı

MASTER OF SCIENCE THESIS

Aviation Science and Technology Department

December 2019

Yeni İmal Edilecek Uçak Motor Parçaları Üretim Sürecinin Tasarlanması için
Ontoloji Geliştirilmesi

Doğan Yağcı

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Havacılık Bilimi ve Teknolojileri Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Mustafa Ertunç TAT

Aralık 2019

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Mustafa Ertunç TAT danışmanlığında hazırlamış olduğum “Yeni İmal Edilecek Uçak Motor Parçaları Üretim Sürecinin Tasarlanması için Ontoloji Geliştirilmesi” başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim.

31/12/2019

Doğan Yağcı

ÖZET

Endüstride bilgi toplama, saklama ve paylaşma amacıyla kullanılan birçok sistem ve araç firmalara rekabette bir adım öne geçebilme adına hizmet vermektedir. Bu sistem ve araçlara teknoloji yardımı ile her geçen gün yenileri eklenmektedir. Bilginin niteliği de niceliği de ayrı ayrı önem kazanmaktadır. Havacılıkta motor parçası üretimi nicelikten çok nitelikli bilgi üretilen, saklanan ve paylaşılan genellikle atölye tipi bir üretim şeklidir. Bu tipteki üretimde üretilen bilginin kaybolmaması, nitelikli bir şekilde saklanması ve yerinde ve zamanında paylaşılması seri üretime göre daha zordur. Bu çalışma için de endüstride bilgi sistemlerinde ontolojilerin ve birbiri ile ilişkili bilginin kullanımı tartışılarak bir hava aracı motoru parçasının ilk kez imalat sürecine katılımı örnek bir uygulama olarak modellenmiştir.

Havacılık imalat süreçleri; ileri imalat teknolojileri, savunma teknolojileri, araştırma ve geliştirme faaliyetlerinin de dâhil olduğu karmaşık ve hassas imalat süreçlerini içeren geniş bir alandır ve gaz türbinleri gibi yüksek miktarda mühendislik çalışması yapılan motorların parçalarının ilk kez imalat sürecine katılması, nitelikli ve tecrübeye dayalı bilgi kullanımını gerektirir. Bu tezin uygulama kısmında motor parçalarının ilk kez imalat sürecine uyarlanması konusunda Amerikan Savunma Bakanlığı İmalat Teknolojileri Programı'nın endüstri firmaları ve akademik araştırmacıların işbirliği ile tanımlamış olduğu imalat olgunluk seviyeleri takip edilerek ontoloji temelli modelleme çalışması yapılmıştır. Hazırlanan model ile yeni parçaların imalat sistemine uyarlanması süreci boyunca süreç takipçisini yönlendirecek, gerçekleştirilen ve gerçekleştirilecek aktivitelerin kontrol edilmesini sağlayacak bir yazılım uygulaması hazırlanmıştır.

Anahtar Kelimeler: Havacılık, imalat, ontoloji, imalat olgunluk seviyeleri, yeni parça tanıtma

SUMMARY

Many systems and tools used in the industry to collect, store, and share information do serve companies to take a step forward in the competition. Newer ones are also added to these systems and tools with the help of technology every day. The quality and quantity of information gain importance separately. Production of aircraft engine parts in aviation is generally a shop type production that produces, stores, and shares high-quality information rather than quantity. It is more challenging to keep the knowledge furnished in this type of production, not to be lost, to be stored in a qualified manner, and to be shared on-site and on time. In this study, the use of ontology and interrelated information in the information systems in the industry are discussed. As an example, the introduction of a new aircraft engine part to the manufacturing process is modeled by the help of ontology.

Aviation manufacturing processes are a large area that includes complex and exquisite manufacturing processes, including advanced manufacturing technologies, research, and development activities, and defense technologies. Introduction of a new gas turbine engine part, which includes a high level of engineering, to manufacturing process requires the use of qualified and experience-based knowledge. In the application section of this thesis, a new engine part introduction to the manufacturing process has been modeled with the help of an ontology-based approach according to the manufacturing readiness level documents, created by collective work of industry, academia, and American Department of Defence Manufacturing Technology Group. This ontology-based model has been used in developing a software application that can be used to orientate process managers to control manufacturing activities throughout the new part introduction process.

Keywords: Aviation, manufacturing, ontology, manufacturing readiness levels, new part introduction

TEŞEKKÜR

Tez çalışmalarımda bana yardımcı olan ve beni yönlendiren danışmanım Doç. Dr. Mustafa Ertunç TAT 'a, desteklerini hiçbir zaman esirgemeyen aileme ve tez süreci boyunca hep yanımda olan canım kardeşim Demet YAĞCI 'ya sonsuz teşekkürlerimi sunarım. Ayrıca akademik çalışmalarına başlamamı ve yaptığım çalışmaları destekleyen, karşılaştığım zorluklarda yardımlarıyla yanımda olan dostlarıma, tez konusu olarak seçtiğim süreçler hakkında düşünmeme vesile ve destek olan iş arkadaşlarım ve yöneticilerime, yabancısı olduğum disiplinlerdeki çalışmalarımda yardımcı ve yönlendirici olan arkadaşlarıma ve hocalarıma teşekkürü bir borç bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ VE AMAÇ	1
2. TEORİK BİLGİ VE LİTERATÜR ARAŞTIRMASI	3
2.1. Üretim Bilgi Sistemleri ve Ontolojiler.....	3
2.1.1. Üretim yönetimi sistemleri	3
2.1.2. Üretim yönetimi sistemlerinde modülerlik	4
2.1.3. Üretim yönetiminde bilgi sistemleri	4
2.1.4. Bilgi sistemlerinde birlikte çalışabilirlik	6
2.1.5. Birlikte çalışabilirlik için ontoloji.....	7
2.1.6. Ontoloji oluşturma	8
2.1.7. Ontolojilerin yapısı	9
2.1.8. İlişkisel ve grafik veri tabanları	10
2.1.9. Ontolojilerin uygulama alanları	12
2.1.10. İmalat alanı ile ilgili ontolojiler	13
2.1.11. İmalat alanı ile ilgili önemli bir ontoloji: ISA-95	14
2.2. Ontolojiler için Yazılım Altyapısı.....	15
2.2.1. Anlamsal ağ nedir?	15
2.2.2. Anlamsal ağ bileşenleri.....	17
2.2.3. Kısaltmalar.....	20
2.2.4. Web ontoloji dili	20
2.2.5. Ontoloji mühendisliği	21
2.2.6. Çıkarımlar	22
2.2.7. Açık ve kapalı dünya varsayımları	23
3. MATERYAL VE YÖNTEM.....	24
3.1. Kullanılan Yazılım Araçlar.....	24
3.1.1. Ontoloji editörü.....	24
3.1.2. Java geliştirme ortamı.....	24
3.1.3. Apache Jena kütüphanesi.....	24
3.2. Kurulan Ontoloji Bazlı Model	26
3.2.1. Modelin amaçları.....	26
3.2.2. Modelin kullanıcıları	27

	<u>Sayfa</u>
3.2.3. Fayda sağlayabileceği konular.....	27
3.2.4. Model kurulurken kullanılan prensipler	29
3.2.5. İmalat olgunluk seviyeleri	29
3.2.6. Yeni parçanın imalat sistemine tanıtılması.....	31
4. UYGULAMA.....	33
4.1. İmalat Aktiviteleri.....	35
4.2. İmalat Aktiviteleri için İlişkiler	36
4.3. İmalat Olgunluk Seviyesi Soruları.....	39
4.4. İmalat Olgunluk Seviyeleri İçin İlişkilendirmeler	40
4.5. Kavramların Kullanımı	42
4.6. Jena ile Kodlanan Dosyalar	43
4.7. Uygulama Programı	43
4.7.1. Proje arayüzü	44
4.7.2. Parça arayüzü.....	45
4.7.3. Aktiviteler arayüzü	46
4.7.4. İmalat olgunluk seviyeleri arayüzü.....	46
4.7.5. Kavramlar arayüzü.....	47
5. SONUÇ VE ÖNERİLER.....	49
KAYNAKLAR DİZİNİ	51
EK AÇIKLAMALAR.....	55
Ek Açıklama -A: Jena kütüphanesi kullanılarak hazırlanan kodlar	55
Ek Açıklama-A.1. “readyToDo” aktiviteleri görüntüleme için yazılan koddan bir kısım..	55
Ek Açıklama-A.2. Aktivitelerin Özelliklerini görüntülemek için yazılan koddan bir kısım	56
Ek Açıklama-A.3. İmalat olgunluk seviyesi sorularının özelliklerini görüntüleyen koddan bir kısım	57
Ek Açıklama-A.4. Kavramların Özelliklerini görüntülemek için metot.....	58
Ek Açıklama-A.5. Aktivite eklemek için yazılan kodların ana kısmı.....	59

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. Veri tabanları karşılaştırması a) grafik veri tabanları b) ilişkisel veri tabanları	11
2.2. Anlamsal ağ bileşenleri	18
2.3. İki kaynaklı bir RDF şeması	18
3.1. Apache jena yapısı (Apache Jena, 2011)	25
4.1. “ManufacturingFacility” sınıfları	34
4.2. “StandartPartOWL”	35
4.3. Aktivite sınıfı bireyleri	36
4.4. Aktivite ilişkileri	37
4.5. Aktiviteler için veri tipi özellikler	38
4.6. Aktiviteler için obje tipi özellikler	38
4.7. MRL adımlarındaki soruların bireyler olarak gösterim örneği	40
4.8 “checkForMRL” ilişkisi ile gösterilmiş üçlüler	41
4.9. “rdfs:comment” ilişkisi	41
4.10. MRL Veri ilişkileri	42
4.11. “DocumentStatusProperty”	43
4.12. Proje arayüzü	44
4.13. Parça arayüzü	45
4.14. Aktiviteler arayüzü	46
4.15. İmalat olgunluk seviyeleri arayüzü	47
4.16. Kavramlar arayüzü	48

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1. Kullanılan Standart Kısaltmalar	18
2.2. Açık ve Kapalı Dünya Varsayımları Arasındaki Farklar	22



KISALTMALAR DİZİNİ

Kısaltmalar Açıklama

API	Uygulama Programlama Arayüzü (Application Programming Interface)
ADACOR	Dağıtılmış Üretim Sistemleri için Uyarlanabilir Holonik Kontrol Mimarisi (Adaptive holonic Control architecture for distributed manufacturing systems)
CWA	Kapalı Dünya Varsayımı (Close World Assumption)
DOD	Amerikan Savunma Bakanlığı (Department Of Defence)
ERP	Kurum Kaynak Planlama (Enterprise Resource Planning)
GRDDL	Kaynak Tanımlarını Dil Lehçelerinden Ayıklama (Gleaning Resource Directory Description Language)
IDE	Bütünleşik Geliştirme Ortamı (Integrated Development Environment)
IDMU	Sistemin Dijital Kopyası (Integral Digital Mock-Up)
IEEE	Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
IIC	Endüstriyel İnternet Konsorsiyumu (Industrial Internet Consortium)
IIRA	Endüstriyel İnternet Referans Mimarisi (Industrial Internet Reference Architecture)
IOT	Nesnelerin İnterneti (Internet of Things)
IRI	Uluslararası Kaynak Tanımlayıcısı (Internationalized Resource Identifier)
LOTA	Uzun Süreli Arşivleme (Long Term Archiving)
MASON	İmalat Semantiğinin Ontolojisi (Manufacturing's Semantics Ontology)
MES	İmalat Uygulama Sistemleri (Manufacturing Execution Systems)
NIST	Ulusal Teknoloji ve Standartlar Enstitüsü (National Institute of Standards and Technology)
MRL	İmalat Olgunluk Seviyesi (Manufacturing Readiness Level)
NPI	Yeni Parça Tanıtımı (New Part Introduction)
NSF	Ulusal Bilim Vakfı (National Science Foundation)
OWA	Açık Dünya Varsayımı (Open World Assumption)
OWL	Web Ontoloji Dili (Web Ontology Language)
PLM	Ürün Yaşam Çevrimi Yönetimi (Product Lifecycle Management)

KISALTMALAR DİZİNİ (devam)

Kısaltmalar Açıklama

PSL	Süreç Tanımlama Dili (Process Specification Language)
RAMI	Endüstri için Referans Mimari Model (Reference Architecture Model for Industry)
RDF	Kaynak Tanımlama Çerçevesi (Resource Description Framework)
RDFS	Kaynak Tanımlama Çerçevesi Şeması (RDF Schema)
RDMBS	İlişkisel Veritabanı Yönetim Sistemleri (Relational Database Management Systems)
RDS	İlişkisel Veri Tabanı (Relational Database)
RIF	Kural Değişim Biçimi (Rule Interchange Format)
SaaS	Hizmet Olarak Yazılım (Software as a Service)
SPARQL	SPARQL Protokolü ve RDF Sorgu Dili (SPARQL Protocol and RDF Query Language)
STEP	Ürün Veri Modeli Alışverişi Standardı (Standard for The Exchange of Product)
SWT	Standart Bileşen Araç Kiti (Standard Widget Toolkit)
SKOS	Basit Bilgi Organizasyonu Sistemi (Simple Knowledge Organization System)
SQL	Yapılandırılmış Sorgu Dili (Structured Query Language)
TOVE	Toronto Sanal Organizasyon Girişimi (Toronto Virtual Enterprise)
TRL	Teknoloji Hazırlık Seviyesi (Technology Readiness Level)
UML	Birleştirilmiş Modelleme Dili (Unified Modeling Language)
XML	Genişletilebilir İşaretleme Dili (eXtensible Markup Language)

1. GİRİŞ VE AMAÇ

Havacılık sektöründe motor parçaları üretimi kendine has özelliklere sahiptir. Bunlardan öncelikli olanları üretilen parçaların sayısının genellikle az olması ve üretilen parçaların seçilen malzemeler ve kullanılan imalat yöntemleri açısından nitelikli parçalar olmasıdır. Tasarım ekipleri tarafından yüksek sıcaklıkta çalışma şartlarına uygun seçilen malzemeler genellikle termal özellikleriyle öne çıkarlar. Süper alaşımlar gibi malzemelerin sıklıkla tercih sebebi budur ve bu tür malzemeler maliyeti yüksek malzemelerdir. Sunulan ürünlerin yapımında maliyeti arttıran etmenler arasında bunların yanı sıra birbirinden farklı gereksinimleri olan imalat yöntemlerinin tek parçada çok sayıda bir araya gelmesi ve üretim zamanlarının da yüksek olması da öne çıkar. Ayrıca hava taşıtları için motor parçaları üretimi sektörü can güvenliğine doğrudan etkileri sebebi ile kalite gereksinimleri açısından da sıkı kontrollü yönergelerle takip edilen süreçler, her karakterin yüzde yüz kontrolü ve yapılan işlerin uzun süreli dokümantasyonu gibi kendine özgü uygulamaları olan standartları oldukça yüksek bir sektördür.

İmalat teknikleri her sektöre göre çeşitlilik gösterir. Geleneksel yöntemler olarak bilinen genelde mekanik ve ısı enerjisi kullanan yöntemler olan torna, freze, taşlama gibi yöntemler, ileri teknikler olarak sayılabilecek elektrik enerjisi kullanan elektro erozyon yöntemleri ve kimyasal enerji kullanan elektrokimyasal işleme gibi yöntemler, lazer ışını ile kaynak, gaz altı kaynak ve lehimleme gibi birleştirme teknikleri, çeşitli kaplama teknikleri ve yüzey özelliklerini değiştiren teknikler ve bunların yanı sıra son yıllarda gelişim gösteren eklemeli üretim teknikleri havacılıkta kullanılan tekniklerden örnekler olarak sayılabilir. Atölye tipi geleneksel imalat süreçleri ve yeni teknoloji imalat süreçlerine sahip havacılık imalat süreçleri ve bu imalat süreçlerinin bir araya gelmesiyle üretilen parçalarda bilgi toplanması, depolanması ve tekrar kullanılması süreçlerinin standart bir model yardımı ile yürütülmesi oldukça zordur. Ayrıca bu bilgi yönetimi sürecinin paydaşları da işletmelerin farklı bölümlerinde çalışan farklı tecrübe ve uzmanlığa sahip kişilerdir. Bu kişiler arasındaki bilgi paylaşımı zaman zaman yukarıda sayılanlardan daha fazla önem arz eden bir probleme dönüşebilmektedir. İşletmelerde genellikle birçok süreç için standart veya değiştirilebilir modeller uygulanmaktadır. Standart olanlarda toplanan bilginin niteliği düşük iken değiştirilebilir olanlarda ise üretilen bilgiye ulaşım oldukça zordur. Bu zorluğun

bir sebebi de üretilen bilginin öznel olması ve yazılımlar tarafından okunabilecek formatta olmamasından kaynaklanır. Literatürde birçok makalede verinin farklı uzmanlıklara sahip çeşitli alanlardan gelmesi, bunların eş zamanlı olarak paylaşılıp kullanılması ve değişikliklerinde çıkacak sorunların çözümünde ortak dil ve veri yapıları kullanılması gerekliliklerinden bahsedilmiştir. (Ray ve Jones, 2006; Matsokis ve Kiritsis, 2010)

Bu sebeplerle üretim tesislerinde aynı kavramsal dil ile iletişim kurulabilmesi ve üretilen bilginin kaybolmaması ve tekrar erişilebilmesi için depolanabilmesi gerekir. Karmaşık imalat süreçlerinin tekrarlanabilmesi, güncellenebilmesi ve en iyi haline getirilebilmesi için bu süreçlerin yazılımlar tarafından okunabilen, özelleştirilebilen, dinamik modeller halinde tanımlanması, günümüz şartlarında gereklilik haline gelmiştir.

Amerika kıtasında “Smart Manufacturing”, Avrupa kıtasında ise “Industry 4.0” olarak adlandırılan güncel endüstriyel yenilenme, imalat sürecini takip eden internete bağlı makineleri içermektedir ve veri analiz teknikleriyle imalat performansını artırmayı hedeflemektedir. Bunun sonucunda yeni imalat süreçleri ve modelleri gündeme gelmiştir. Thoben vd.nin (2017), Endüstri 4.0 ve akıllı imalat konusunu incelediği yazısında bahsettiği gibi arayüzler ve veri modellerinin standartlaştırılması akıllı imalat sistemlerinde en gözde konulardan biridir.

Havacılık ve motor parçaları kapsamında yeni imalat adaptasyonu süreci, imalatı yapılacak bir parçanın başka parçaların üretiminde kullanılan bir üretim ortamına uyarlanması, parça adetlerinin arttırıldığında takip edilecek süreç planının oluşturulması ve bu planda gereksinim duyulacak her kaynağın belirlenip gerekli uyarlamalarının yapılması süreçlerini içerir. Süreç her bir adım detaylandırıldığında karmaşık bir hal almakta ve takibinin yapılması zor bir sürece dönüşmektedir ve süreç yöneticisinin yönlendirilme ihtiyacı doğmaktadır. Birçok tesiste bu yönlendirme ihtiyacı tecrübeye ve kurum kültürüne dayalı olarak giderilmektedir. Bu adaptasyon süreci Amerikan Savunma Bakanlığı İmalat Teknolojileri Programı’nın endüstri temsilcileri ve akademisyenler yardımıyla hazırlayıp tavsiye ettiği ve yayınladığı İmalat Olgunluk Seviyeleri (Manufacturing Readiness Level, MRL) kullanılarak gerçekleştirilebilir. Bu tezin amacı yeni parça imalat adaptasyon sürecinin imalat olgunluk seviyeleri yol gösterimiyle modellenmektir. Bu amaçla bir ontoloji oluşturulmuş ve bu ontoloji ile çalışan bir uygulama hazırlanmıştır.

2. TEORİK BİLGİ VE LİTERATÜR ARAŞTIRMASI

Bu çalışmanın teorik bilgi ve literatür taraması kısmında üretim bilgi sistemleri, ontolojiler ve anlamsal ağ konuları incelenmiştir. Üretim bilgi sistemleri kısmında üretim yönetimi için bilgi sistemlerinin kullanımı ve bu sistemlerin sahip olması gereken önemli özellikler incelenmiştir. Anlamsal ağ ve ontolojiler kısmında ise anlamsal ağın gelişimi ve onu oluşturan yapı taşları incelenmiştir. İlk kısımda mevcut durumlar ile ilgili bilgiler verilmiş ve ontolojilerin faydalı olabileceği alanlar gösterilmiş, ikinci kısımda ise ontolojilerin nasıl yapılandırılabilceği gösterilmiştir.

2.1. Üretim Bilgi Sistemleri ve Ontolojiler

Üretim bilgi sistemleri ve ontolojiler başlığı altında üretim yönetimi sistemleri, bu sistemlerdeki modülerlik gereksinimi ve bilgi sistemlerinin mevcut durumu, üretim yönetimi bilgi sistemlerinde birlikte çalışabilirlik kavramları incelenecektir. Ontolojilerin birlikte çalışabilirlik için önemi tartışılacak, ontolojilerin yapısı ve kullanımı hakkında bilgiler verilecektir. Ontolojilerin uygulama alanları hakkındaki bir kısım ardından imalatta ontolojilerin kullanımı konuları incelenecektir.

2.1.1. Üretim yönetimi sistemleri

Üretim sistemleri üretim gereklilikleri, üretim kavramları, sistem parametreleri, malzemeler ve diğer istekleri ile çok büyük sistemlerdir. Birçok farklı bakış açısı ile bu sistemleri yönetmek gerekliliği vardır. Kurumlarda genellikle çeşitli birimler sisteme kendi bakış açıları ile bakarlar ve kendi hedefleri doğrultusunda kararlar almaya çalışırlar. Birbiri ile çelişen farklı birimlerin hedefleri şirketler içerisinde karar almayı zorlaştıran ve yavaşlatan en önemli problem kaynaklarından biridir. Üretim yapan şirketlerin yönetim sistemlerinde en genel beş bakış açısı tasarlama, planlama, simülasyon, kontrol ve uygulama olarak sayılabilir ve yönetim sistemleri bunlarla analiz edilebilir. Bu aşamalar üretim sistemlerinin yönetiminde temel kavramlardır. (Cutting-Decelle vd., 2007)

Üretim sistemlerinde çıkan problemlere yaklaşım genellikle rijit, sıralamalı ve sonunda da sistemin sonuna doğru tek yönlü olarak yönlenmiş şekilde olmaktadır. Genellikle de sistem tamamlandığında başlarken yaratılan gereklilikler sona gelindiğinde değişmiş olarak görülür. Daha gelişmiş sistemler ise sürekli geri besleme veren ve hızlı hareket eden sistemlerdir. Bu yaklaşımların çalışabilmesi için ise hızlı koordine olma, bilginin hızlı elde edilmesi ve işlenmesi ve hızlı iletişim gerekliliklerinin sağlanması zorunludur. (Frankovič vd., 2002)

2.1.2. Üretim yönetimi sistemlerinde modülerlik

Modülerlik ihtiyacı aynı kurum içinde ürün bazında ve üretim bazında, ayrıca da diğer firmalar ile kurulan sistemlerde kendini gösterebilir. Modülerliği yönetmekte kritik bir problem de verinin, iletişimin ve üretimde dolanan bilginin yönetimidir. Veri genellikle daha önceden modellenmiş yapılarda bulunmaktadır ve bu formatlar genellikle özelleşmiş formatlardır. Dahası bu yapı üretim süreci boyunca tutarlı halde bulunmamaktadır, üzerindeki anlam da genellikle zayıf ve farklı şekillerde anlaşılabilir niteliktedir. Bunun yanında farklı araçlar arasında iletimi ya doğrudan değildir ya da otomasyonu kolay değildir. Veri iletişimindeki yaklaşımlar da genellikle anlamsal (semantic) olmayıp sadece format tutarlılığına dayalıdır. Bilgi paylaşımı ve bilginin tekrar kullanımı bu sebeple bu tarz yazılımlarda büyük bir birlikte çalışabilirlik problemi yaratmaktadır. (Cutting-Decelle vd., 2007)

2.1.3. Üretim yönetiminde bilgi sistemleri

Bilgi yönetimi üretim sistemlerinde yıllardır en önemli meselelerden biri olarak kabul edilmektedir (Gains vd., 1995). Bilgi yönetiminin amacı, Despres ve Chauvel (2001) tarafından karar verme mekanizmalarında kritik olan farklı şekillerdeki bilgiyi yaratmak, paylaşmak ve kullanmak için geliştirilen ve uygulanan araçlar, süreçler, sistemler, yapılar ve kültürlerin aracılığıyla organizasyonel yapının iyileştirilmesi olarak tanımlanmıştır.

Genel olarak bakıldığında dünyada üretimin çoğunluğunun seri imalat tipinde olduğu bir dönem yaşanmaktadır. Buna istinaden üretim aktiviteleri de birbirini tekrar eden görevlerden ibarettir. Öte yandan ürün çeşitliliğindeki artış da geniş boyutlara ulaşmaktadır.

Buna rağmen, aynı tasarım ve planlama görevlerinin tekrar etmesi günümüz endüstrisinde günlük görevleri şekillendiren olgudur. Üretilen büyük miktarda veri ise çoğu zaman işlenmemiş bir halde kalmaktadır. (Efstratios, 2015)

Havacılıkta motor parçaları üretimi gibi bilgi yoğun ve atölye tipi üretim ortamlarında ise veri çok yüksek miktarlara ulaşmamakla birlikte bilginin niteliğinin yüksek olmasından bahsetmek mümkündür. Bu durum, bilgi yönetiminin seri üretime nazaran daha da önem kazanmasını sağlar. Havacılık endüstrisi için bu durum Mas vd. (2015) tarafından özetlenmiştir. Buna göre sonraki yıllarda mühendisler ve sahip olmaları gereken nitelikler açısından eşzamanlı mühendislik (concurrent engineering) trendinin ardından işbirlikçi yaklaşımla mühendislik trendinin (collaborative engineering) yerleşeceği öngörülmektedir. Mühendislik tanımlamaları açısından IDMU (Integral Digital Mock-Up) yani bütün sistemin dijital kopyası, üretim açısından dijital fabrika (digital factory), PLM açısından SaaS (Software as a Service) yani hizmet olarak yazılım, fiziksel altyapı olarak telefonlar, bilgisayarlar, tabletler ve sanallaştırma eğilimi, bilgi modelleri olarak ise model bazlı tanımlamalar (Model Based Definition) ardından bilgi tabanlı mühendislik (Knowledge Based Engineering) trendleri öngörülmüştür. Veri formatları, veri değişim formatları ve uzun süreli arşivleme (Long Term Archiving, LOTA) için de uluslararası standartların işlerlik kazanacağı düşünülmektedir.

Bilgi sistemleri ve teknolojileri üretim sistemlerinde en önemli konulardan ikisi olarak öne çıkıyor. Yıllar içinde üretim sistemleri ve üreticiler elle çizilmiş resimleri çizip yorumlamaktan bilgisayar destekli teknolojilere evrildiler. Günümüzde bir ürünün yaşam döngüsünün neredeyse tamamı dijital benzetimle gösterilebilmektedir. Kurum Kaynak Planlama (Enterprise Resource Planning, ERP), Ürün Yaşam Çevrimi Yönetimi (Product Lifecycle Management, PLM), İmalat Uygulama Sistemleri (Manufacturing Execution Systems, MES) gibi yazılımlar da çok hızlı bir şekilde gelişme gösterdiler ve artan rekabet koşulları ile göstermeye de devam edeceklerdir. ERP, PLM, MES gibi sistemler oldukça efektif bir şekilde firmalar içerisinde iletişimi ve bunun yanında firmalar arası iletişimi sağlayabilmektedir. Fakat verinin yazılımlar tarafından yorumlanabilmesi uygulamaları, çok fazla üzerinde durulmamış bir olgudur. Tüm sistemler yorumlama kısmını, bu konuda yetenek elde etmiş insan kaynaklarına bırakmaktadırlar.

2.1.4. Bilgi sistemlerinde birlikte çalışabilirlik

İmalat teknolojilerinde son zamanlarda birçok gelişme yaşanmaktadır ve bu gelişmeleri tetikleyen farklı fikirler ortaya atılmaktadır. Kaliteyi ve çeşitliliğe uyumu arttırma, maliyetleri düşürme amacı ile ortaya çıkan yeni teknoloji ve sistemler beraberinde bilgi ve iletişim sistemlerinde de yeniliklere ihtiyaç duyulmasına sebep olmaktadır. İnsan, makine ve yazılım kaynaklarının kendi içlerinde ve birbirleri ile kurdukları ara yüzler vasıtası ile kurdukları karmaşık sistemlerin bu paydaşlarla ve birbirleri ile kurdukları iletişim, üretim şirketlerinin başarısında oldukça belirleyici olacaktır.

Bahsedilen teknolojik gelişmelerden en önemlilerinden biri siber fiziksel sistemlerdir (Cyber Physical Systems). NSF'nin (National Science Foundation) tanımına göre; her biri farklı yer ve zaman skalasında yer alan, çoklu ve ayırık modellerde davranış gösteren, konu ile alakalı olarak birçok farklı yolla birbiri ile haberleşen ve birbirleri ile iç içe geçmiş fiziksel ve yazılımsal parçalardan oluşan sistemlerdir (Chungoora ve Young, 2011). Bu tanım ile bu sistemlerin birbiri ile iletişim ihtiyacı çok açık bir şekilde ortaya konulmaktadır. Bu iletişimi mümkün kılacak yapı ise nesnelerin interneti (Internet of Things, IOT) adı ile anılmaktadır. Bu yapı başka bir kaynakta akıllı cihazların, sistemlerin, ürünlerin, bulut temelli servislerin, insanların ve şirketlerin oluşturduğu tam teşekküllü bir ağa erişimi mümkün kılan teknoloji olarak anılmıştır (Nolin ve Olson, 2016). Bilgiye ulaşım kolaylığı, depolanmış ama yapılandırılmamış bilginin kullanıma alınma şansı ve siber fiziksel sistemlerin yardımı ile büyük boyutlarda bilgi üretimi yapılabilmektedir. Üretilen büyük boyutlarda yeni bilgiler ile de işlenmeye ihtiyaç duyan bir bilgi yığını oluşmaktadır. Oluşan bu ve diğer kaynakları organize etmek ve barındırmak için ise yeni gelişen bulut tabanlı servisler tercih edilmektedir. Böylelikle bilgi ve tecrübenin saklanması, gerekli zamanlarda kullanılması ile süreçlerin tekrar keşfedilmesinden ziyade bunların kullanımı ve geliştirilmesine yoğunlaşılması mümkün kılınmıştır (Zeid vd., 2019). Çeşitli gelişmiş analiz yöntemleri ve makine öğrenmesi uygulamaları ise bu bilgi yığınlarını kullanmaya ve çıkarımlarda bulunmaya yardım edecek araçlar olarak imalat sektöründe gelişim kat etmektedirler. Bu tipteki araçlar planlama, karar destek sistemleri, kalite sistemleri, öngörü sistemleri gibi birçok sistemi besleyebilmektedirler.

Tüm bahsedilen gelişmeler imalat alanında üretilen ve kullanılan bilginin paydaşlar arasında sorunsuz değişimine büyük ölçüde bağlıdır. Bu olgu ISO 16100'de (2011) birlikte çalışabilirlik (interoperability) adı altında belirtilmiş ve ortak bir arayüzde uygulamaya özel fonksiyonların birbirleri ile ilişki kurması için ortak bir semantik ve sözdizimi kullanılarak bilgi paylaşma ve değişimi yeteneği şeklinde tanımlanmıştır. Karşılıklı işlerlik de diyebileceğimiz “interoperability” IEEE'nin “Enterprise IT Body Of Knowledge” kaynağında sentetik ve semantik olmak üzere iki çeşitte kategorize edilmiştir. İlki verinin formatına daha çoklukla odaklanırken ikincisi verinin anlamlandırılmasına odaklanır.

Karşılıklı işlerlik kavramını imalat sektörüne ve diğer sektörlerede adapte edebilmek amacı ile birçok model geliştirilmiştir. Bunların arasında öne çıkan iki mimari vardır. Birincisi Platform Industrie 4.0 adıyla bilinen organizasyon tarafından geliştirilen “Reference Architecture Model for Industry 4.0”tür ve RAMI 4.0 adı ile bilinmektedir (DIN, 2016). Diğeri ise bir diğeri büyük organizasyon olan IIC (Industrial Internet Consortium) tarafından geliştirilen “Industrial Internet Reference Architecture”dır (IIC, 2019) ve IIRA kısaltması ile ulaşılabilir. İki yapı da karşılıklı işlerlik yakalamaya çalışarak endüstri şirketleri içerisindeki farklı katmanları birbirine bağlamaya çalışır.

2.1.5. Birlikte çalışabilirlik için ontoloji

Farklı kaynaklardan gelebilecek farklı formlarda tasarlanmış bilginin toplanması ve bilgisayarlar tarafından anlamsal olarak yorumlanıp sistem paydaşlarına sistemin tasarlanışına uygun bir şekilde ulaştırılması bilgi sistemlerinde en önemli konulardan biridir. Sistemlerin birlikte çalışabilirliğini sağlayan bu olguyu ontolojilerin sağladığı mantıksal kurallar işletme ve sonuç çıkarma becerileri ile başarmak mümkündür.

İmalat odaklı kurumlarda bilgi toplama ve paylaşma işlemleri ERP ve PLM gibi yazılım sistemleri tarafından yapılmaktadır. Bu sistemlerin son yıllardaki gelişimi oldukça güçlü olsa da bu sistemler veri tabanı odaklı sistemler olmaya devam etmektedirler. Bu yüzden arkalarında çok gelişmemiş ontolojiler vardır. Birçok kurumda bilgi sistemlerinde birlikte çalışabilirlik gerekliliği uluslararası standartları insan zekâsı işgücü kaynağı kullanılarak sağlanmaktadır. Uluslararası standartlar her ne kadar tüm alanları

kapsayamasalar da sistemlerin birlikte çalışabilirliği için önemli ontolojiler sayılırlar. Üretim alanında aşağıdaki standartlar bunlara örnek olarak sayılabilir:

- ISO 10303- Ürün Veri Modeli İçin Veri Alışverişi Standartları (Standart for The Exchange of Product Data Model, STEP)
- ISO 13584- Parça Kütüphanesi (Part Library)
- ISO 15531- Endüstriyel İmalat Yönetim Veri Standardı (Industrial Manufacturing Management Data)
- ISO 13399- Kesici Takım Standartları (Cutting Tool Standart)
- ISO 18629- Süreç Modelleme Dili (Process Specification Language, PSL)
- ISA95- Kurum- Kontrol Sistem Entegrasyonu (Enterprise- Control System Integration)

Çok özelleşmiş ve dar alanlarda o alan ile ilgili standartların çok faydalı olduğu söylenebilir. Bir alan ile ilgili terimlerin anlamları o alan sahipleri tarafından zaten anlaşılmış olduğu için sıkı sıkıya tanımlanmış olmaması mümkün olabilir. Fakat PLM gibi birden çok standardın geçerli olduğu daha geniş bir alanda sistemlerin birlikte çalışabilirlik kavramı bir soruna dönüşür. Böyle sistemlerde bilginin ortak bir standarda bağlı halde paylaşılması bir gereklilik olmuş olur. Ayrıca birçok standart da yapılandırılmamış halde metin bazlı olarak kaydedilmiş ve paylaşılmaktadır. Bazen aynı standart içinde bile tutarlılık sorunları yaşanmaktadır. (Usman vd., 2011)

Üretim sektöründe ürün tasarımı, imalatı ve servis ve bakımı ve bunların onlarca alt dalı farklı standartlar ile yürütülmektedir. Zaman zaman her birinin arasında bilgi paylaşılması gerekliliği doğabilmekte ve ara yüzler oluşmaktadır. Bu bilgi paylaşımının sorunsuz ve hızlı olması kurumları birlikte çalışabilirlik kavramına yönlendirmektedir.

2.1.6. Ontoloji oluşturma

Ontoloji felsefeden alınmış bir terimdir ve bu terim var olanı sistematik olarak ifade eder. Bilgi tabanlı sistemler için ise var olan ifade edilebilendir (Gruber, 1993). Ontoloji de bağlamda belirli bir alandaki bilgi parçalarının sınıflandırılıp ilişkilendirilmesinin bir metodu olarak tanımlanabilir.

Ontoloji öncelikli olarak semantik web, yapay zekâ ve enformatik gibi alanlarda çalışma konusu olmuştur. Büyük veri depolarında ve internet sitelerinde saklanan ve bu veriyi manipüle eden yazılımlar için veri yapısının gösterimi amaçlanarak geliştirilmeye başlanmıştır. (Gruber, 1993; Berners-Lee vd., 2001)

Uygulama alanları Noy ve McGuinness (2001) tarafından aşağıdaki gibi tanımlanmıştır;

- Belirli bir alandaki bilgiyi analiz etmek.
- İnsanlar ve yazılımlar arasında bilginin yapısına ilişkin ortak bir anlayış oluşturmak.
- Belirli bir alandaki bilgiyi tekrar kullanabilmek.
- Belirli bir alandaki bilgi değişimlerini kolay kılmak amacıyla o alana ilişkin varsayımları kesinleştirmek.
- Belirlenen alana ilişkin bilgiyi o alanın operasyonel bilgisinden ayırmak.

2.1.7. Ontolojilerin yapısı

Ontolojilerin insan ve makine arasında iletişimi, ve hatta makineden makineye ve insandan insana iletişimi, ontolojide belirtilen terminoloji çerçevesinde sağlıyor olabilmesi gerekmektedir (Guarino, 2009). Bunu, içindeki sınıf (class) yapısı ile sağlar. Sınıflar bazen konsept olarak da ifade edilirler. Alt ve üst sınıflara sahip olabilecek şekilde birinin bir diğerini kapsayabilme özelliği vardır.

Bu sınıflar bireylere sahip olabilmektedir. Her bir birey kendi sınıfının özelliklerini taşır ya da belirlenen konsepte göre özellikler alabilmektedir. Bir girdinin sınıf veya bir başka sınıfın bireyi olması ontoloji hazırlanırken çok ince bir çizgi ile belirlenir. Aslında bu karar ontolojinin kullanıma nasıl hizmet edeceğine göre verilen bir karardır. Bireyleri olmaya başladığı anda bir ontoloji veri tabanına dönüşmeye başlar.

Özellikler (“properties”, “roles” veya “slots” diye anılabilirler), ontolojinin bir sınıfı ya da bireyi tanımlamak için kullanılan yapıtaşlarından. Bu özellikleri sınırlandırmak için (restrictions) çeşitli yapılar da kullanılabilmektedir.

Bir ontoloji tasarlarırken birçok farklı metot ve araç kullanılabilir fakat hepsinin takip ettiği belli başlı prensipler vardır:

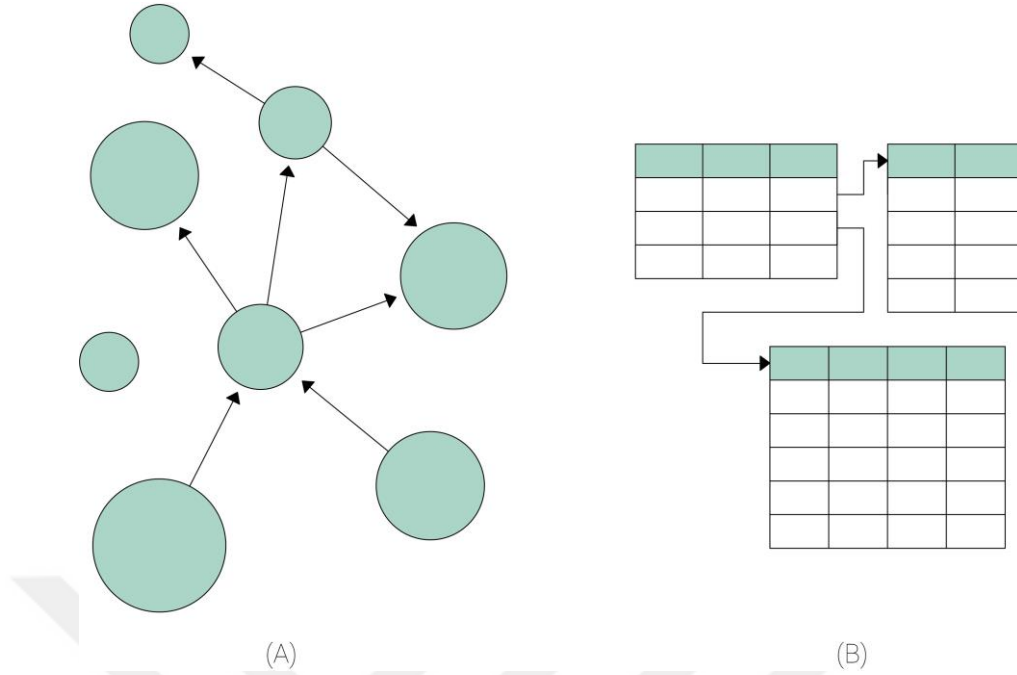
- Bir ontolojiyi inşa etmenin tek bir doğru yolu yoktur. Uygulamadaki gereksinimlerine ve gelişebileceği alanlara göre birçok farklı şekilde kurulabilir.
- Ontolojiyi tasarlamak ve kurmak iterasyonlarla ilerleyen bir süreçtir.
- Oluşturulan sınıflar gerçekteki objelere ve bunların çalışma mantığına olabildiği kadar yakın olmalıdır.

Ontoloji tasarlama ve kurma süreci boyunca karar verilmesi ve yapılması gereken işleri aşağıdaki gibi sıralayabiliriz:

- Ontolojinin kapsamını belirleme,
- Sınıfları tanımlama,
- Sınıfları hiyerarşik yapı içerisinde yerleştirme,
- Özellikleri tanımlama,
- Özellikleri ilişkilendirme ve sınırlama,
- Bireylerin özelliklerine göre aldıkları değerleri belirleme,
- Kuralları, kısıtlamaları ve aksiyomları tanımlama.

2.1.8. İlişkisel ve grafik veri tabanları

Bu çalışmada bilginin yoğun olduğu bir model oluşturulurken, veri tabanı yapısı olarak çoğunlukla kullanılan ilişkisel veri tabanları (Relational Database Management Systems, RDMBS) yerine grafik veri tabanlarının kullanımı tercih edilmiştir. Sebeplerini açıklamak için bu iki yapıyı Şekil 2.1’de görüldüğü gibi karşılaştırabiliriz.



Şekil 2.1. Veri tabanları karşılaştırması: a) grafik veri tabanları b) ilişkisel veri tabanları

Öncelikle ontolojilerin geleneksel veri tabanı sistemlerinin bir alternatifi olmadığını belirtmek gerekmektedir. Ontolojilerin en bilinen amaçları arasında bilgisayarların format yanında verinin anlamını da anlayıp yorumlaması gelmektedir. Birçok uygulamada ontolojiler bilginin bir şeması veya gösterimi denilebilecek biçimde kullanılırken veri tabanları böyle bir sistemdeki veriyi kaydetmek amacı ile kullanılmaktadır. Bu veriyi bir veri tabanı sisteminde depolayıp bir uygulama vasıtası ile güncelleyip bir insan tarafından okunarak yorumlanmasını ve bilgiye dönüştürülmesini sağlamak mümkündür. Birçok geleneksel veri tabanı bu şekilde çalışmaktadır. Bazı çalışmalarda ise bir uygulama için önce ontolojinin yaratıldığı, bunun yenilenerek uygulamanın ve veri tabanının otomatik olarak tekrardan bir yazılımcı ihtiyacı olmadan esnek bir şekilde değiştirilebildiği çalışmalar vardır (Mahmudi vd., 2018). Bazı uygulamalarda ise ontoloji sorunu çözmenin tek yolu olabilmektedir. Bahsedilen uygulamalar miktar olarak veriden çok bilgiye ihtiyaç duyan uygulamalardır. Eğer belirli alandaki bilgi karmaşık yapıda ise ve alanın sınırlandırılmayıp yeni bilgiler ile ilişkiler tanımlanabilir olması gerekiyor ise ontolojinin bu tipteki durumları kolaylıkla kaldırabilmesi ontoloji kullanıcılarına büyük avantaj sağlar. Dahası tanımlanan ilişkiler ve çıkarımlar aksi durumlarda kullanılamayacak bir takım bilgilerin sorgulanabilmesini mümkün kılmaktadır. SPARQL sorgu dili hazırlanırken amaçlanan en önemli getirilerden biri de budur.

2.1.9. Ontolojilerin uygulama alanları

İnsanlar, birçok farklı alanda eskisine kıyasla çok fazla miktarda, farklı kaynaklardan ve farklı alanlardan ve sistemlerden bilgi edinme imkanına sahiptir. Şirketlerde de bilgi sistemleri oldukça gelişmiş olmasına rağmen bilgi hala farklı formlarda depolanmaya ve paylaşılmaya devam etmektedir. Birçok farklı veri tabanı, tablolar, sunumlar, farklı formatlardaki dokümanlar ile çalışanlar bilgiye ulaşımı sağlıyor ve bu şekilde paylaşımı yapıyor. Fakat tüm bilgi farklı formatta olduğu ve birbirine göre anlam ifade etmediği için insanlar ve makineler tarafından anlaşılması adına kavramların ve arayüzlerin de her bir paylaşımda tekrar tekrar oluşturulması gerekmektedir. Bu şekilde bilgi ilişkilendirilmesi sağlanmadan bütünsel bir sistem oluşturulması mümkün olmuyor ve sıklıkla iletişim kopukluklarından kaynaklanan hatalar ile vakit kaybı yaşanması engellenememektedir. Daha özelleşmiş bir şekilde örneklendirirsek; şirketlerdeki süreç haritalarını ele alabiliriz. Bu haritalar en iyi haliyle UML veya SysML dilinde yazılmış olsun. Kullanılan ERP sistemi ise büyük olasılıkla ilişkisel bir veri tabanında (Relational Database Systems, RDS) depolanıyor olacaktır. Eğer bahsedilen şirket bir endüstri şirketi ise bu şirketin üretim alanından gelen bilgiler de satın aldıkları makineler veya kurulan sistemlerin beraberinde gelen bazı standartlara tabi bilgiler olacaktır. Şirketin süreç haritalarında belirttiği operasyonların çoğu için de kullanılan bir PLM sistemi varsa eski sistemlerin bu sisteme adaptasyonlarının yapılmadığı sıklıkla rastlanan bir durumdur.

Eğer üretim şirketleri gibi karmaşık yapılara sahip kurumlar tarafından bu sorunun önüne geçilmeye çalışılıyorsa verinin tüm ilişkilerin tanımlanabileceği bir formatta tutulmasının getirebileceği faydalar göz ardı edilemez. Ontolojilerin belki de en büyük faydası farklı alanlardaki bilgilerin arasındaki ilişkileri görünebilir ve tanımlanabilir kılmasından gelmektedir. Bu özelliği ile ontolojiler bilgi yönetim sistemi çeşidi sayılabilir. Herhangi bir alandaki bilgiyi istenilen şekilde modelleyerek saklanmasını mümkün kılarlar. Tabii ki bu bilginin kullanıcılara kolay erişilebilecek biçimde ve daha nitelikli biçimde sunulması da üstün özelliklerinden biridir. Tüm bu özellikleri sebebi ile ontolojilerin yapay zekâ projelerinde, anlam içeren web uygulamalarında, sistem mühendisliği alanında, yazılım mühendisliği çalışmalarında, bilgiye dayalı biomedikal bilgi sistemlerinde, kütüphane hizmetlerinde, şirketlerde bilgi erişim hizmetlerinde ve veri mimarisi gibi alanlarda yoğun bir kullanımı olması beklenmekte ve görülmektedir.

2.1.10. İmalat alanı ile ilgili ontolojiler

İmalat alanında ontolojiler birçok farklı açıdan yaklaşımlar ile ilgi çekici bir konudur. İmal edilecek ürünlerin geometrik modellenmesi, konfigürasyon ve ürün ağaçlarının modellenmesi, ürün verisinin farklı formatlara çevrimi, üretim şekillerinin ve süreçlerinin modellenmesi, kaynak ve gereklilikler ile bunların alt konuları ontolojik olarak farklı bakış açıları ile ifade edilebilmektedir. Ontolojiler akademik olarak da birçok yayında kullanılmış veya yayının ana konusunu teşkil etmiştir.

PLM ile ilgili birçok ontoloji çalışması literatürde mevcuttur. Matsokis ve Kiritsis (2010) PLM sistemleri için bir referans ontoloji yaratmaya çalışmışlardır. Bir diğer çalışmada EU-FP7 projesi amePLM için destekleyici bir çalışmadan bahsedilmiştir. (Bruno vd., 2015)

Sanal işletme (Virtual Enterprise) ve organizasyon modelleme alanında yapılan çalışmalardan Uschold vd.nin (1998) çalışmasında iş koluna göre gerekli terim ve tanımlamalar yapılmıştır. Sonrasında bu tüm terim ve tecrübelerini amaçları doğrultusunda bu ontoloji üzerinde tanımlamışlardır. Yapılan çalışmada ve bu ontolojinin kullanımında elde ettikleri verileri paylaşmışlardır. TOVE ve ADACOR da daha eski projeler arasında sayılabilir. (Fox,1992; Leitão ve Restivo, 2006)

Tedarik zinciri ile ilgili çalışmalar yapılmıştır. Birinde ürün merkezli bir tedarik zinciri ontolojisinin tesis genelindeki sistemlere uyumluluğu üzerinde durulmuştur. (Lu vd., 2012)

Ürün ve süreç geliştirmede kaynak kapasite modellemesi ve karar destek sistemleri için de çalışmalar yapılmaktadır. Bir çalışmada oluşturulan OWL bazlı imalat kaynak kapasite ontolojisi (MaRCO) ile imalat için sunulan kaynakların tanımlanması ve kolaylıkla kombine edilerek kullanılması için bir model oluşturulmuştur. Bu çalışma ile kullanıcının ihtiyaç duyulan kaynakları potansiyel adaylar içinden hızlı bir şekilde seçebilmesi hedeflenmiştir. (Järvenpää vd.,2019)

Veri aktarımı ve değişimi için oluşturulan bir ontoloji örneği imalat alanında sıklıkla kullanılmaktadır. ISO 10303 (STEP) de CAx sistemler arası bilgi aktarımını sağlayan oldukça yaygın biçimde kullanılan bir ontolojidir.

Siber fiziksel sistemler için ise Petnga ve Austin'in (2016) çalışması, bu tarz sistemler için uygulanabilir ontoloji bazlı bir veri toplama çerçevesi ve ilişki çerçevesi sunar.

PSL farklı formatlardaki imalat ile ilgili çalışan aralarında süreç modelleme, süreç planlama, simülasyon, iş akışı, proje yönetimi, iş süreçleri gibi yazılımların arasında bilgi akışına yardımcı olmak amacı ile oluşturulmuş bir formattır. Bu araçların PSL formatı ve kendi formatlarını kullanarak birlikte çalışabilmesi mümkündür. Böylelikle de her sistem otomatik bir şekilde PSL aracılığı ile diğer sistemlerle iletişim kurabilir. (Schlenoff vd., 2000)

MASON ontolojisi ise NIST tarafından sunulan PSL ontolojisine benzemekle beraber bir üst ontolojidir. İmalat alanını kaynaklar, girdiler ve operasyonlar olarak biçimlendirmiştir. Bunların aralarındaki ilişkileri ontolojik olarak tanımlamaya çalışmıştır. (Lemaignan vd., 2006)

2.1.11. İmalat alanı ile ilgili önemli bir ontoloji: ISA-95

İmalat alanında ontolojilerin en önemli olanlarından biri olan ISA-95, birçok farklı açıdan işletmelerdeki işletme sistemleri süreçlerini ve üretim ile ilgili kontrol sistemlerini birbirine bağlayan, uluslararası ölçüde kabul görmüş bir standarttır.

Model ve terminoloji ismiyle adlandırılan ilk kısımda bir organizasyonun yönetimi ile ilgili sistemleri üretim operasyon ve kontrol sistemlerine ilişkilendiren arayüzler tanımlanmıştır.

İkinci kısımda ise ilk kısımda tanımlanan bilgiyi modellemeye ayrılmıştır. Bunu yaparken bilginin özelliklerini listeler, örneklendirir ve UML modelleme dili kullanarak modellemeyi yapar.

Üretim ortamı birçok aktivitenin bir araya gelmesi şeklinde düşünülmektedir ve bunlar arasındaki bilgi akışını sağlanması gerekmektedir. Standart, üçüncü kısımda bilginin akışını ve üretimdeki aktiviteleri tanımlar. Bu aktiviteleri genel olarak grupladığımızda üretim, kalite, bakım ve stok gibi başlıklar karşımıza çıkar. Bu kısımda

anlatılan modeller bunlar gibi aktivitelerin açık bir şekilde tanımlanması ve anlık durumun takibi amacıyla kullanılabilir.

Üçüncü kısımdaki aktiviteleri üretim uygulama sistemi aktiviteleri olarak tanımlayabiliriz. Dördüncü kısımda bu sistemin aktivitelerinin arasında aktarılabilecek olan bilgiyi düzenler. Dolayısıyla bu sistemi modellemiş olur.

Standart beşinci kısımda, iş süreçlerini, daha önce birinci ve ikinci kısımda tanımlanmış olan modeller ile bağdaştırır. Bir organizasyonun tüm kısımları ve süreçleri arasında bilgi alışverişi olabilmektedir. Fakat bu standart ERP gibi iş süreçlerini kapsayan sistemleri üretim ortamında kullanılan MES gibi sistemlerle ilişkilendirmeye odaklanır.

ERP ve PLM gibi yazılımlar ISA95 içerisinde son katmanda yer alırlar. ISA95 bu tip yazılımlar için veri yapılarının oluşturulmasında ve bunların üretim kontrolü ve yönetimini sağlayan MES tipindeki yazılımlar gibi yazılımla iletişimi sağlayacak standart modelleri sunmaktadır.

Organizasyonlar bu iki yazılımı beraber kullanmak ya da sadece birini kullanmak konusunda tercih yaparlar. Bu tercihi yaparlarken ürettikleri ürünün karmaşık yapıda olması, üretim miktarı ve tasarımını da yapıyor olup olmadıklarına göre ihtiyaçlarını belirlerler. Genellikle siparişe göre üretilecek üründe ürün tasarım mühendislik çalışması az ise sadece ERP yazılımları yeterli olarak görülür. Fakat ürün tasarım süreci uzun ve birçok farklı birimin bir arada çalışmasını gerektiriyorsa da PLM tercihi ağırlıklıdır. Organizasyonlar çoğunlukla iki tip yazılımı da kullanır ve bunlar arasındaki bilgi iletimi için bir arayüz oluşturmak zorunda kalmaktadırlar.

2.2. Ontolojiler için Yazılım Altyapısı

2.2.1. Anlamsal ağ nedir?

Geleneksel web sitelerinin içeriği sadece insanlar tarafından okunabilir ki bu verilerin otomatik işlenmesi için uygunsuzdur ve veri ile ilgili diğer bilgileri araştırmak için

yetersizdir. Bahsedilen veri kümeleri, birbiri ile bağlantılı olmayan veri siloları olarak düşünülebilir. Veri kullanımındaki bu sınırlama farklı bir format kullanılarak giderilebilir. İşte bu yazılıma sözdizimi yanında anlam da sağlayan yapı Anlamsal Ağ (Semantic Web) olarak anılır. 2000’li yılların başında geliştirilmiştir (Berners-Lee vd., 2001). Web 2.0, anlık mesajlaşmalar, bloglar, forumlar, sosyal medya platformları ve web sendikasyonlarının arkasındaki teknolojilerin bir koleksiyonu için kullanılan genel bir terimdir. Ağın gelecek jenerasyonu, yapay zeka yardımı ile gerçekleştirilecek ve bilgisayar üretimi içerikleri de kapsayacak olan, özelleştirme, anlamlı içerikler ve daha sofistike web uygulamaları için genel bir terim olan Web 3.0 olarak adlandırılmıştır. (Sikos, 2015)

Tipik bir web sayfası yapılandırıcı elemanlar, belirli formatta yazılar ve multimedya objeleri içerir. Web tasarımcıları tarafından yaratılan başlıklar, yazılar, bağlantılar ve diğer web sitesi elemanları, bilgisayarlar için anlamsızdır. Tarayıcılar web belgelerini bağlantılar üzerinden görüntüleyebiliyorken, sadece insan aklı bilginin anlamını yorumlayabilir, bu yüzden insanların anlayabildikleri ile bilgisayarların anlayabildikleri arasında büyük bir fark vardır. Görüntüler için alternatif metin belirtilse bile, veri ilgili veri ile yapılandırılmaz veya ilişkilendirilemez ve geleneksel web sayfası paragraflarındaki insanların okuyabildiği kelimeler, herhangi bir belirli sözdizimi veya yapı ile ilişkilendirilemez. İçeriği olmadan web siteleri tarafından sunulan bilgi, arama motorlarına belirsiz gelebilir. (Sikos, 2015)

İçerikler web sitelerine yapılandırılmış veri ekleyerek, yapılandırılmış veri setleriyle alakalı biçimlendirilmiş dipnotlar veya adanmış harici meta data dosyaları olarak ve onları diğer web sitelerine bağlayarak, makine tarafından işlenebilir ve anlamı açık hale getirilebilir. Diğer faydalarının arasında, yapılandırılmış veri dosyaları işleme için geleneksel web sitelerinden daha geniş bir görev aralığı sağlar ve çok daha verimlidir. Yapılandırılmış veri formatları, özellikle sorguların etkili bir biçimde bilgi alışverişini gerçekleştirilebildiği, ilişkisel veri tabanları Access ve SQL veri tabanları içinde kullanılmaktadır (Sikos, 2015). Çünkü ilişkisel veri tabanlarında anlamsal ağ teknolojilerinin kullanılmasını sağlayacak adreslemeyi kolayca yapılabilmesini sağlayan standartlar tanımlanmıştır (Arenas vd.,2011). Anlamsal ağ standartları tarafından desteklenen GraphDB, neo4j, Amazon Neptune gibi ticari veri tabanı yazılımı paketleri de son yıllarda piyasaya çıkarılmaktadır.

Anlamsal ağ, geleneksel web sitelerinde nadiren görülen veya hepsinde kullanılmayan, kendine özgü birçok özelliğe sahiptir. Örneğin, verinin büyük bir kısmı, veri paylaşımının ve dağıtımının gerçekten ücretsiz olmasına imkân vererek, bariz şekilde tanımlanmış açık lisansla sunulmuştur. Resmi olarak tanımlanmış veri bağlantıları, kesin ifade doğrulamasıyla birlikte otomatik bilgi keşfi mümkünlüğü sağlar. Nesne ve özelliklerin her biri, bir web adresiyle ilişkilendirilebilir, böylece herhangi bir kaynağa kolayca ulaşılabilir.

2.2.2. Anlamsal ağ bileşenleri

Anlamsal ağ bileşenleri Şekil 2.2’de gösterilmiştir. En yaygın kabul edilmiş bilgi-yönetim standartları Kaynak Tanımlama Çerçevesi (Resource Description Framework, RDF), Web Ontoloji Dili (Web Ontology Language, OWL) ve Basit Bilgi Organizasyonu Sistemi (Simple Knowledge Organization System, SKOS)’tur.

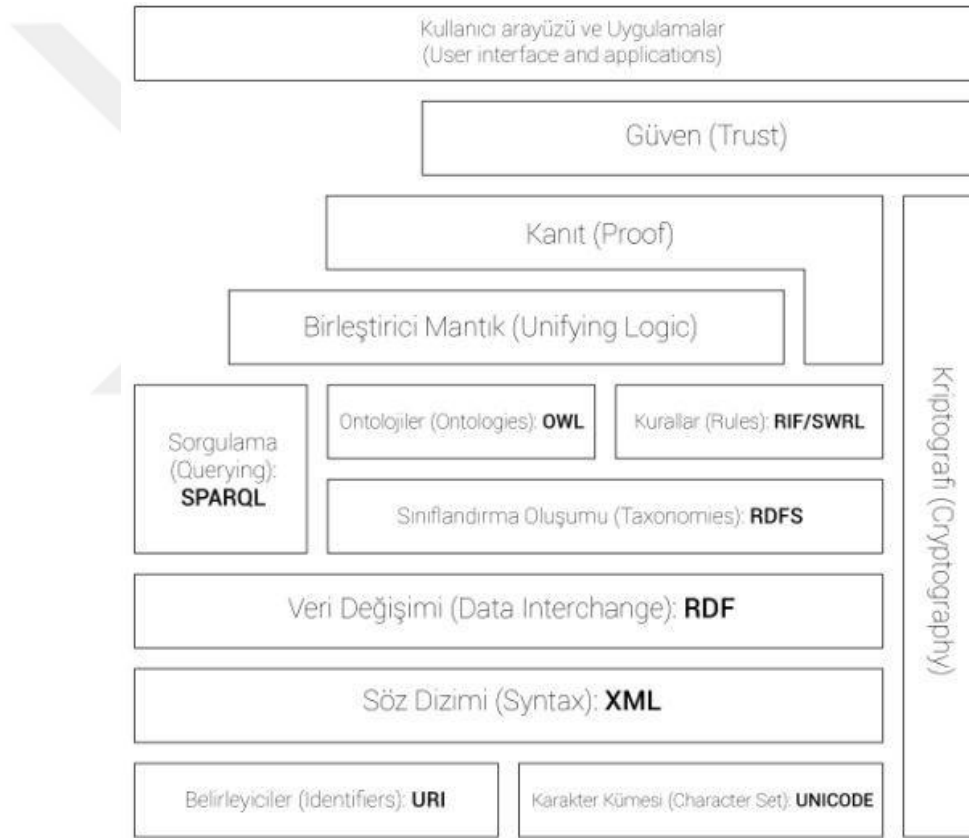
RDF, internet ağı üzerindeki bilgiyi temsil etmek için düzenlenmiş bir çeşit veri modeli çerçevesidir. RDF üzerine kurulu olduğu XML sözdizimi üzerine kendi semantik ilişkilerini kurarak kendi sözdizimini (syntax) oluşturur.

RDF dili oluşturulurken belli amaçlarla oluşturulmuştur. Bunları şu şekilde sıralayabiliriz:

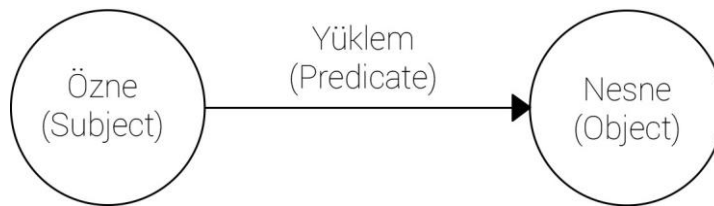
- Ağ üzerindeki kaynaklardaki bilgiyi sistemlerin kullanmasını sağlayacak bir metadata olarak,
- Sınırlı olmayan, kısıtlanmamış modellerden gelecek bilgiyi kullanabilecek uygulamalara kaynak olarak,
- Bilgiyi yaratıldığı uygulamalar dışında da diğer alanlarda okunup işlenebilecek şekilde düzenleyerek,
- Uygulamalar arası ortak bir kaynak olarak birleştirildiğinde sorunsuz yeni bilgi kaynağının yaratılması amaçlanarak,
- Ağ üzerindeki bilginin sadece insanlar değil yazılımlar tarafından da okunup dünya genelinde ortak bir oluşturulması için yaratılmıştır.

RDF çok az kısıtlaması olan esnek bir dildir. Bağımsız izole uygulamalarda kullanılabileceği gibi asıl değeri bilginin paylaşılması süreçlerinde ortaya çıkmaktadır. (Klyne ve Carrol, 2004)

RDF yapısının temelinde üçlüler vardır. “RDF Graph” olarak adlandırılan yapı ise bu üçlülerin bir araya gelmesi ile oluşur (Şekil 2.3). Her bir üçlü içerisinde “subject” yani özne, “object” yani nesne ve bu ikisini birbirine bağlayan “predicate” olarak tanımlanan bağlayıcı öge barındırır.



Şekil 2.2. Anlamsal ağ bileşenleri



Şekil 2.3. İki kaynaklı bir RDF şeması

Özne ve nesne olarak belirtilen her bir nokta (RDF Node) birbirine bağlayıcılar ile birbirine bağlanır. Bu noktalar RDF yapısında 3 farklı şekle sahip olabilen yapılardır.

- Uluslararası kaynak tanımlayıcı (Internationalized Resource Identifier, IRI)
- Ön tanımlı bilgi (Literal)
- Boş nokta (Blank Node)

IRI ve Literal RDF yapısında “resource” yani kaynaklar olarak adlandırılırlar ve var olan her hangi bir şeyi temsil edebilirler. Örnek olarak herhangi bir doküman, makine parçası, soyut bir kavram ve ya bir sayı bile kaynak olarak tanımlanabilir.

Üçlü, RDF dilinde bir ifade belirttiğinde buna “RDF Statement” adı verilir. Bu ifade içinde bağlayıcılık görevi üstlenen “predicate”te bir IRI ile tanımlanır. Boş nokta bir ifadede yer aldığı anda ise bunun anlamı bağlayıcının isimlendirilmemiş bir nokta ile ilişkilendirilmiştir.

RDFS yani RDF Schema, RDF dilini genişleterek sınıfları ve bunların bazı özelliklerinin tanımlanmasına ve sınıflandırmalara izin veren bir dildir. Ayrıca RDF dilinde özellik (property) olarak belirtilen bileşene “domain” yani değer alabileceği alan ve “range” yani değer alabileceği aralık eklenebilir. Bunun yanında RDF sınıf (class) ve özelliklerini (property), RDFS kullanarak sınıflandırmak mümkündür. (W3C, 2014)

W3C tarafından tanımlanan OWL ise bir ontoloji dilidir. Bu anlamsal ağ aracı ile bir alanı tanımlayan kavramları, kavram gruplarını ve bunlar arasındaki ilişkileri daha zengin ve karmaşık biçimlerde gösterebilecek bazı tanımlamalar yapar. Ontoloji denildiğinde akla gelen anlamsal ağ aracı OWL’dir. Bu çalışma yayınlandığı esnada yayında olan versiyonu OWL 2’dir. Bu, 2004 yılında yayınlanan ilk versiyonunun ardından 2012 yılında yayınlanan ikinci versiyonudur. W3C Web Ontology Working Group tarafından hazırlanan son haline sitesinden erişilebilir. (W3C, 2012)

2.2.3. Kısaltmalar

Ontoloji çalışmalarında genellikle kısaltmalar kullanılarak gösterim yapılır. Tanımlanan bu standartlar Çizelge 2.1’de gösterilmektedir.

Çizelge 2.1. Kullanılan standart kısaltmalar

Prefix (Kısaltma)	Açılım
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#
xsd:	http://www.w3.org/2001/XMLSchema#

2.2.4. Web ontoloji dili

Ontoloji kelimesi, felsefe çalışmalarında orijinal olarak varoluşun doğası olarak tanıtılmıştır. Bilgisayar biliminde ise ontoloji, makinelerin okuyabildiği bir şekilde sınıflar (kavramlar), özellikler (nitelikler), ilişki tipleri ve girdiler (bireyler, örnekler) şeklinde tanımlarla, bir organizasyon, araştırma projesi, tarihsel bir olay vb. gibi dünyanın bir parçasını açıklamak için bir veri modeli oluşturup, var olanı kavramlaştırma olarak adlandırılır. OWL yani Web Ontology Language ontolojileri tanımlamak için standart bir araçtır. Gelişmiş ontoloji dilleri, OWL de bunlardan biridir, aşağıda tanımlanan bileşenlerden oluşur;

- **Sınıflar (Classes):** Sınıflar genellikle grupları veya yaygın özellikleri paylaşan üyeleri temsil eder. Sınıflarda hiyerarşi daha yüksek seviye (süper class veya parent class) ve daha alçak seviye (sub class veya child class) sınıflar olarak ifade edilmiştir. Örneğin bir kurumu bir sınıf olarak ele alırsak, bu kurumdaki farklı birimleri onun altsınıfları olarak inceleyebiliriz. Ayrıca bu kurum da benzer özelliklere sahip kurumları bir araya getiren bir organizasyonun altsınıflarından biri sayılabilir.

- Nitelikler (Attributes): Objeleri ve sınıfları belirten bakış açıları, özellikler, karakteristikler veya parametreler olarak tanımlanır. Bunlar, sınıfların ve bu sınıfların bireylerinin insana veya makine bakış açısı ile anlamlandırılmasını sağlayan elemanlardır.
- Bireyler (Individuals): Örnekler veya objeler de denilebilir. Bir veya birden fazla sınıfın özelliklerini taşıyan bireylerdir. Atomik oluşlardır.
- İlişkiler (Relations): Sınıflar arasında, bireyler arasında, bir sınıf ve bir birey arasında, tek bir obje ve bir koleksiyon arasında veya koleksiyonlar arasındaki mantıksal bağlar olarak tanımlanabilir.
- Fonksiyon terimleri (Function terms): Belirli ilişkilerden tarafından şekillendirilmiş karmaşık yapılardır.
- Kısıtlamalar (Restrictions): Bir kaynağın alabileceği değer aralığının veya limitlerinin tanımlanmasıdır.
- Kurallar (Rules): Kurallar “If-Then” (Eğer öyleyse böyledir) şeklindeki mantıksal çıkarımlardır.
- Aksiyomlar (Axioms): Kurallarla birlikte ontolojiyi tarifleyen mantık çerçevesindeki ifadelerdir. Aksiyomlar örnekler veya sınıfların değerleri üzerindeki kısıtlamaları onlara yüklemek için kullanılır, bu yüzden de genellikle mantık bazlı dilleri kullanarak ifade edilirler. Ayrıca aksiyomlar ontolojinin tutarlılığını doğrulamak için uygundur.
- Olaylar (Events): Nitelik veya ilişkilerdeki değişikliklerdir.

2.2.5. Ontoloji mühendisliği

Ontoloji mühendisliği, ontolojiler yaratmak için metodolojiler ve metotları kapsayan bilgisayar biliminin bir alanıdır. Anlamsal ağ ontolojilerinin amacı, aynı alanın farklı alanları veya farklı bakışlarının organizasyonları karşısında birlikte çalışabilirlik sağlayan uygulamalar arasında iletmeye hazır paylaşılan bilgi ortamı yaratmaktır. Ontoloji mühendisliği alanının ilgilendiği kavramlardan en önemli olanları şunlardır:

- Ontoloji dönüşümleri (Ontology Transformations), yeni bir amaç için var olan ontolojinin yeni gereksinimleri ile ilgilenmek için ontolojinin geliştirilmesidir.

- Ontoloji birleşimleri (Ontology Merging), aynı bilgi alanının iki veya daha çok ontolojisinden yeni tek bir tutarlı ontoloji yaratılmasıdır.
- Ontoloji entegrasyonu (Ontology Integration), farklı bilgi alanlarından iki veya daha çok kaynak ontolojisinden yeni bir ontolojinin yaratılmasıdır.
- Ontoloji haritalaması (Ontology Integration), farklı ontolojilerden girdiler arasında anlamsal ilişkiler tanımlamak için biçimsel bir ifadelerdir.
- Ontoloji hizalaması (Ontology Alignment), iki veya daha fazla ontoloji arasında, ilk ontolojinin ifadesinin ikinci ontolojinin ifadesini doğruladığı tutarlı ve uyumlu bir bağlantı yaratma sürecidir. (Sikos, 2015)

2.2.6. Çıkarımlar

Genel anlamda çıkarımlar, anlamsal ağ içerisinde hali hazırda tanımlanmış ilişkilerden yeni ilişkiler çıkarmak olarak anlatılabilir. Anlamsal ağda veri, “resources” yani kaynaklar arasındaki bir ilişki dizisi olarak modellenmiştir. “Inference” yani çıkarımlar da, otomatik prosedürlerin verilere ve sözlükler biçimindeki bir dizi kurala dayanarak yeni ilişkiler oluşturabileceği anlamına gelir. Bu yeni ilişkiler uygulamada amaçlanana göre istenirse veriye eklenebilir ya da sorgu çerçevesinde sunulup silinebilirler.

Ontolojiler genellikle sınıflandırma yöntemlerine odaklanır, sınıfları ve alt sınıfları tanımlamak, bu sınıfların bireylerini sınıflarla ilişkilendirmek ve bu ilişkileri karakterize etmek ontolojilerin ana işlevlerindendir. Öte yandan kurallar yeni ilişkilerin tanımlanabilmesi amacı ile bir takım mekanizmaları tanımlamaya ve bu ilişkilere yenilerini eklemeye çalışırlar.

Örnekler ile ayırmaya çalışırsak W3C’nin tavsiye ettiği anlamsal ağ araçlarından RDFS, OWL ve SKOS (Alistair Miles vd., 2010) gibileri ontoloji tanımlamaya çalışan araçlardır, fakat RIF (Kifer ve Boley, 2013) kural bazlı yaklaşımları geliştirmek amacı ile geliştirilmiştir.

Anlamsal ağda çıkarımlar, yeni ilişkiler keşfederek, verilerin içeriğini otomatik olarak analiz ederek veya genel olarak ağdaki bilgiyi yöneterek, ağdaki veri

entegrasyonunun kalitesini iyileştirmek için tercih edilen araçlardan biridir. Çıkarım temelli teknikler bütünleşik verilerdeki olası tutarsızlıkları keşfetmede de önemlidir.

W3C’de çıkarımlar şu örneklerle açıklanmıştır. Birinci örnekte “Flipper bir yunustur.” ifadesi ontolojide var olan “Her yunus bir memeli hayvandır.” ile birleştirildiğinde anlamsal ağ uygulamasının ontolojiye yapılan bu eklemenden Flipper’in bir memeli hayvan olduğunu çıkarıp ontolojiye ekleyebilmesi olasılığının doğduğu anlatılmıştır. İkinci örnekte ise iki kişinin aynı isim, aynı ana sayfa ve aynı e-mail adresine sahipler ise aynı kişi olabilecekleri çıkarımının yapılabileceği gösterilmiştir. Ontolojinin ve kuralların kullanımı bu anlamda oldukça örtüşür ve ikisi de belli durumlarda benzer amaçlarla kullanılabilir.

2.2.7. Açık ve kapalı dünya varsayımları

Ontolojiler, Açık Dünya Varsayımı (Open World Assumption, OWA) şeklinde isimlendirilen açık yaklaşımı sergiler. Bunun da kapalı alan yaklaşımını takip eden geleneksek veri tabanı sistemlerinden farklılıkları Çizelge 2.2’de gösterilmektedir.

Çizelge 2.2. Açık ve Kapalı Dünya Varsayımları Arasındaki Farklar

Açık Dünya Varsayımları	Kapalı Dünya Varsayımları
Genişletilebilir.	Sabitlenmiş.
Aynı isimli kaynaklara izin verilir.	Özgün isimlendirme varsayımını güder.
Grafik yapılıdır, çoklu giriş yapılabilir.	Düz ve hiyerarşik yapılıdır.
Tamamlanmış bilgiye izin verir.	Tüm bilginin tamamlanması gerekir.
Mantıksal aksiyomlarla kısıtlamalar yapılabilir.	Bütünsel kısıtlamalar hatalı veri girişini engeller.
Birden fazla şema mümkündür.	Tek şema varsayılır.
Monotonik mantık güder.	Monotonik olmayan mantık güder.

3. MATERYAL VE YÖNTEM

3.1. Kullanılan Yazılım Araçlar

3.1.1. Ontoloji editörü

Protégé, bir ontoloji geliştirme editörüdür. OntoStudio, TopBraid gibi başka ontoloji editörleri de vardır fakat anlaşılır olması, erişilebilir olması ve kullanım kolaylığı açısından tercih edilmiştir. Ayrıca açık kaynak kod uygulaması olması da tercih edilmesinin sebeplerindendir. Ontoloji geliştirirken ihtiyaç duyulabilecek birçok eklentiye de sahiptir, bunun yanında bu eklentiler geliştirilmeye devam etmektedir.

3.1.2. Java geliştirme ortamı

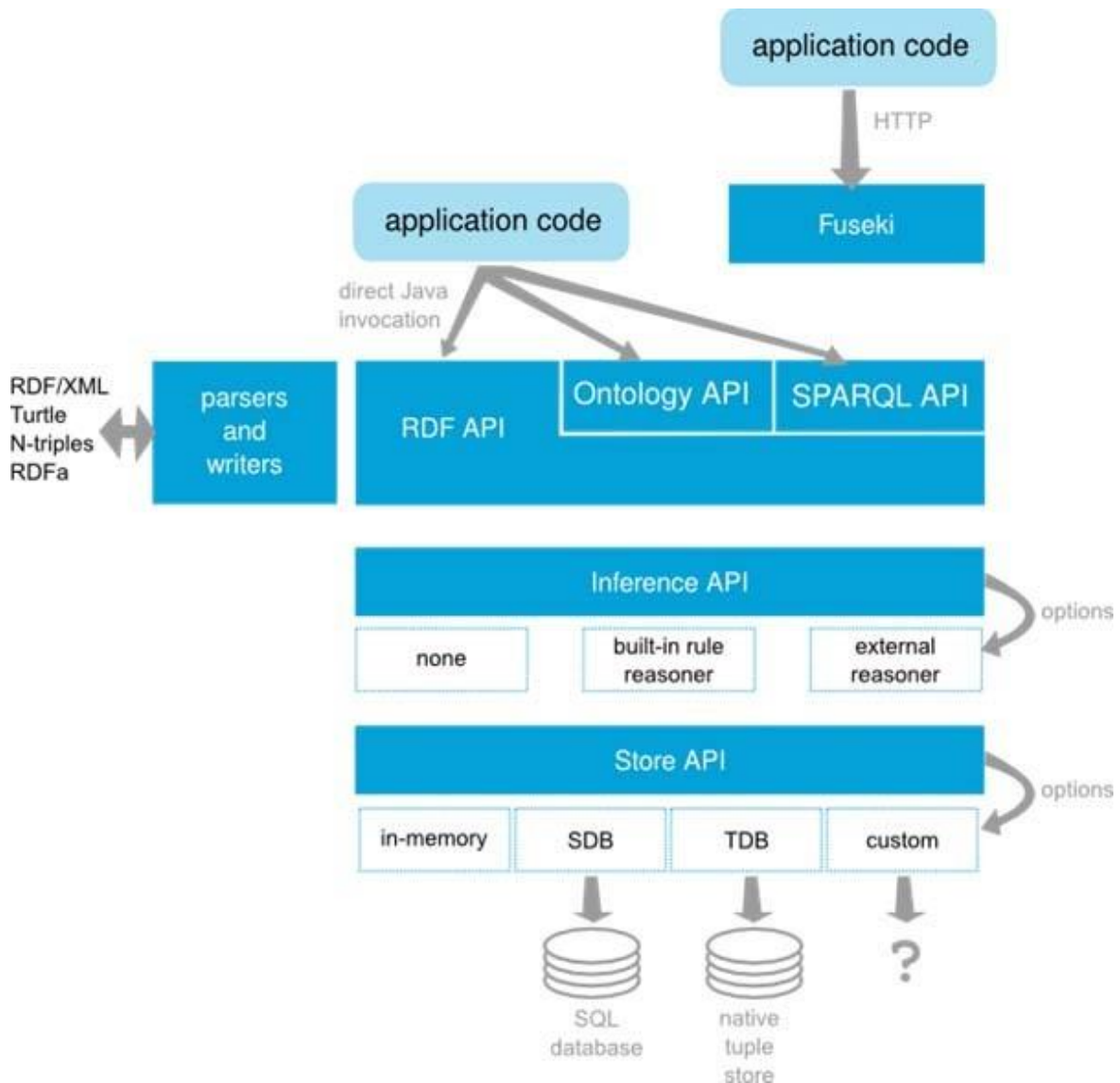
Eclipse IDE kullanılarak java dilinde SWT kütüphanesi kullanarak bir arayüz yaratıldı. Bu arayüze “örnek.rdf” içindeki verileri aktarmak için de Apache Jena Kütüphanesi kullanıldı.

3.1.3. Apache Jena kütüphanesi

Apache Jena Java dilinde yazılmış semantik web uygulamaları geliştirmek için kullanılan bir platformdur. Jena; RDF, RDFS, OWL, SPARQL, GRDDL (Gleaning Resource Descriptions from Dialects of Languages) ve çeşitli çıkarsama motorlarının semantik uygulamaları kodlarken kullanılabileceği bir kütüphane yaratır. Birçok farklı çıkarsama motorunu destekler, RDF/XML, Turtle, N3 ve ilişkisel veri tabanı formatlarında seriler oluşturur.

Jena, RDF ifadeleri ya da üçlülere bir araya toplayarak bir veri modeli oluşturur. Bu platform ayrıca RDF ve OWL dosyalarını yaratmak, okumak, yazmak ve değiştirmek için çeşitli araçlar ve gerekli uygulama programlama arayüzünü (Application Programming Interface, API) de sağlar. (Apache Jena, 2011)

Yukarıda belirtilen yapıya göre çeşitli kaynaklarda depolanan veri ile kullandığı API'ler vasıtası ile etkileşir. Örneğin TDB olarak belirtilen “Triple Data Store” da depolanan üçlüler ile etkileşimi düzenleyen bir API Jena platformunda bulunmaktadır. Bunların üstüne Jena tarafından desteklenen çeşitli çıkarsama motorları kurulabilmektedir. Jena platformu yardımı ile OWL veya RDF formatları ile RDF API ya da Ontology API yardımı ile direkt olarak da etkileşim sağlanabilmektedir. SPARQL API ise SPARQL sorgularını bu model üzerinde çalıştırma amaçlıdır. Yukarıda belirtilen Fuseki ise web uygulamalarında HTTP protokolünü kullanarak modeli manipüle edebilen bir sunucu (server) yapısıdır.



Şekil 3.1. Apache jena yapısı (Apache Jena, 2011)

3.2. Kurulan Ontoloji Bazlı Model

Modeli kurarken ontoloji kullanıldı. Bunun en büyük sebeplerinden biri de ontolojiyi belirli bir mantık çerçevesinde kurulduğunda ontolojinin hem tasarımında hem de uygulamasında çıkarsama motorlarının imkânlarından faydalanabilme şansı yakalanıyor oluşudur (Horrocks, 2013). Bu çalışmada genel anlamda özetlemek gerekirse; havacılıkta üretimi yapılan motor parçalarının yeni parça tanıtımı (New Part Introduction, NPI) için kullanılacak metin bazlı MRL dokümanları yardımı ve üretimdeki aktivitelerin kontrolü için kullanılacak bir mantıksal çıkarım ile ilerleyen çalışmalarda genişletilebilir bir ontoloji oluşturulmuş ve bunun kontrolü için bir yazılım arayüz yapılmaya çalışılmıştır.

Kurulan modelin ilk aşamasında ve daha sonra eklenmesi düşünülen aşamaları için amaçlar, tasarlanan kullanıcıları, sağlayabileceği faydalar ve kurulurken kullanılan prensipler aşağıda anlatılmıştır.

3.2.1. Modelin amaçları

Modelin amacı üretim mühendislerine kendi üretim sistemlerine ilk kez adapte edecekleri parçaları müşterilerinin gereksinimlerine uygun ve eksiksiz bir şekilde çoklu üretime aktarana kadar yapacakları çalışmalarda, farklı iş modellerindeki bilgiyi bir araya toplama ve yaptıkları çalışmayı modellemelerinde kullanacakları araçları belirlemede yol gösterici nitelikte bir model oluşturmaktır. Bu model, birçok farklı sistemden alınacak bilginin toplanmasında kullanılacak yazılımlara da kaynak olarak kullanılabilir.

Modelin tanımlanacak parçanın sorumlusu mühendis tarafından kullanılması ve değişiklik yapılması uygun olacaktır. Zira bu aşamada değişiklik yönetimi ile ilgili tanımlar model üzerinde yapılmayacaktır. Ayrıca zaman yönetimi ilgili tanımlamalar da yaratacağı çakışmalardan dolayı ihmal edilmiştir. Amaca da uygun olacağı düşünülmektedir.

MRL bu çalışmada genel bir çerçeve olarak kullanılmıştır. Bu çerçevede farklı modeller üretilir. Farklı bakış açıları ile aynı model kullanılarak süreç ile ilgili veri yönetimi sağlanabilir.

İmalat süreci açısından 5, 6, 7, 8, 9'uncu seviyeler ağırlıklı incelenecektir. Bu çerçevede her seviye için belirlenen sorulardan yola çıkarak bu soruların içinde barındırdığı aktiviteler ve kavramların tanımlanabileceği bir model oluşturulacaktır.

3.2.2. Modelin kullanıcıları

Yeni parçanın imalat sistemine tanıtılması (NPI) sürecinin yöneticisi tek parça için kurulan modeli yönetmek ile yükümlü olacaktır. Tüm parçalara ait dosyaların bir araya getirilmesi sistem yöneticisi tarafından yapılmalıdır. Sorgular ise her kullanıcıya açık olacaktır.

3.2.3. Fayda sağlayabileceği konular

Ontoloji bazlı modelin birçok fayda sağlamaya ön ayak bir altyapı oluşturması ve ileride üzerine eklenecek yapılar ile aşağıda sıralandığı gibi faydalar sunması beklenmektedir.

Öncelikle yeni parçaların tesise adaptasyonu için, ilgili bilgilerle hazır ilişkilerin belirtildiği bir çerçeve (framework) sunması beklenmektedir. Bu çerçevede yeni ilişkilerin, aktivitelerin, kavramların eklenerek veya çıkarılarak parçaya özgü bir model elde edilmesi beklenmektedir. Tüm yeni parçalara ilişkin bilgilerin bu şekilde dosyalarda toplanarak bir araya getirildiğinde aynı kavramları kullanacakları için özelleşmiş sorgulara imkan tanınması beklenmektedir.

Üretim tesisi ve ortamı içerisinde elde edilen bilgiyi makineler tarafından okunabilir (machine readable) olarak kaydetmek yapılacak sorgular ve bilgiye ulaşım açısından büyük kolaylıklar getireceği gibi bunun yanında önsezili (predictive) sistemler için de büyük bir kaynak sağlamaktadır.

Standartlaştırma kuşkusuz üretimde en önemli konulardan biridir. Gelecek yıllarda da önemini koruyacağına inanılmaktadır. Tüm başlıklarda standartlaştırma hem üretim tesisleri içerisinde hem de tüm kaynaklar ve tedarikçiler açısından da önemlidir. Bu çalışma, oluşan bilgi haznesine standartlaşmış bilgilerin daha kolay biçimde eklenebileceği bir yapıda düzenlenmiştir. Özellikle standartları uyulması gereken kurallar şeklinde

modelleyerek ve kurallaştırarak kullanabilme olanağı ilgi çekici bir özellik olarak öne çıkacaktır.

Standartlaştırmanın faydalarının yanında, üretim tesislerinde standart olmayan, kalıplaşmış form ya da bilgi formatlarına uymayan, toplansa bile ulaşımı ve kullanımı olmayan bilginin sisteme dahil edilebilme olanağı yaratması açısından çok büyük faydaları olabilecektir. Zaman içerisinde farklı formatların oluşmasına büyük katkısı olacaktır. Bunun en büyük faydalarından biri arasında üretim adına kayıtlı tecrübe (know-how) oranının büyük ölçüde arttırılması sayılabilir.

Üretim alanındaki bilgi akışında sağlayacağı kolaylıklar ve otomasyona yatkınlığı ile üretim tesislerinin dijital teknolojileri kullanırken karşılaşılabilecek zorlukları azaltması önemli etkilerinden biri olarak görülebilir. Yapılacak projeleri yönetmek adına da bilgi kaynağı geliştikçe önemi artan bir platform olarak hizmet edebilir.

Sistem ve modellere erişim kolaylığı sağlayabilir. Herhangi bir ilişkiden yola çıkılarak model üzerinde belirtilmiş başka bir noktaya bağlantıları kullanarak ulaşmak mümkündür. Yapının hiyerarşik olmaması da organizasyonel anlamda büyük kolaylıklar getirir. Örneğin aynı ekipmanın farklı sistemlerde başka şartlarda kullanımını modellemek mümkündür.

Üretimde insan kaynağı tesis içerisindeki kaynakları ve sistemleri öğrenebilmek adına büyük zaman harcamaktadır. Bu sürecin ardından başka işle ilgilenmeye başladığında ise elde edilen bilgiler kaybolmaya başlamaktadır. Yapılan değişikliklerde bu süreç baştan başlanılarak tekrar edilmektedir. Model ve kaynak bilgilerinin tamamen otomasyonu sağlanmadığı sürece bu süreç maddi anlamda büyük yük oluşturmaktadır. Böyle sistemlerle var olan sistemin kullanılacak olan insan kaynağına aktarılması büyük kolaylık getirecektir.

Süreçlerin yürütülmesi açısından yapılan modellere uygun gitmeyen süreçlerde süreç yöneticisinin tercihlerine bağlı olarak sürece devam edilir. Bu durumda önceki tecrübelerin kaydı var ise ulaşımı zor olmaktadır. Yoksa da süreç yönlendiricisinin tecrübesi öne çıkar. Süreçlerin dinamik olarak modellenmesi, standartların süreçlerin yönlendirmesi ve standart olmayan durumlarda da tecrübelerin kolaylıkla sorgulanması ile daha etkili bir süreç yönetimi sağlanmış olacaktır.

Değişikliklerin ilgili yerlere iliştilen linkler ile zamanı geldiğinde devreye girmesi büyük zaman katkısı sağlayacak bir iyileştirme olarak sisteme sunulabilecektir. Bu tipteki bir altyapı buna çok uygundur.

Farklı birimler arasında tanımlanmış arayüzlere uygun olarak akışı ve bu akışın üzerinde değişiklik yapma kolaylığı da bir bilgi sistemi altyapısı olarak grafik veri tabanlarının uygulamalarını imalat gibi karmaşık bilgi alanlarında arttıracaktır.

3.2.4. Model kurulurken kullanılan prensipler

Ontoloji tasarımı bazı prensipler belirlenmiştir. Gómez-Pérez vd. (2004) tarafından bahsedildiği üzere;

- “Interoperability”, birlikte çalışabilirlik prensibi anlamsal ilişkileri iyi kurulmuş bir modelin başka ortamlarda farklı girdilerle çalışabilmesi ve başkaları tarafından anlaşılabilmesini ifade eder.
- “Re-usability”, tekrar kullanılabilirlik ise kurulan modelin farklı modüllerinin aktarılacak ve işlenecek bilgiyi azaltması için tekrar kullanımı desteklemesi anlamına gelir.
- “Extensibility”, genişletilebilirlik ilkesi de modelin yeni fonksiyonların eklenmesi ve farklı alanlarda bilgilerin eklenebilmesine uygun olması demektir.
- “Modularity”, modelin modüler olması geliştirme aşamasını kolaylaştırması ve karmaşıklığı azaltması için farklı yapıdaki bilgiyi toplayabilmesidir.

Bunların yanında ontoloji ve kullanımı için oluşturulacak uygulamanın pratik bir kullanıma sahip olması, geri bildirim verebilmesi de beklenmektedir.

3.2.5. İmalat olgunluk seviyeleri

MRL, DOD tarafından hazırlanmış kaynaktır. Tasarımı yapılan bir parçanın imal edilmesi süreci içerisinde hangi olgunluk seviyesinde olduğunu belirlemede kullanılır. Bu genellikle projelerde kısmi ödemeler gibi belirli seviyelerde yapılacak eylemler için

bağlayıcı olarak kullanılır. Aynı zamanda bir yol göstericidir. Bir projenin bir üretim sistemine tanıtılmasında genel bir model sunar. (Anonim, 2018)

Üretim sistemini organize bir şekilde göstermek amacı ile sistemdeki risk alanları baz alınarak belirli gruplar şeklinde MRL içinde temsil edilmişlerdir. Buna göre aşağıdaki gibi gruplarda değerlendirmeler yapılarak farklı boyutlar temsil edilebilir:

- Teknoloji ve Geliştirme Tesisleri
- Tasarım
- Maliyet
- Malzemeler
- Kapasite ve Kontrol
- Kalite
- İmalat işgücü (Mühendislik ve Üretim)
- Kaynaklar ve Tesisler
- Üretim Yönetimi

Tüm bu başlıklar altında birçok alt başlık yer almaktadır. Bunlar da ontolojide belirlenmiş ve birbirleri ile ilişkilendirilmiştir.

MRL bu çalışma grubu tarafından 10 aşamalı olarak planlanmıştır. Yukarıdaki kapsamda belirlenen sorular proje planına göre bu 10 aşamanın içinde yerleştirilmişlerdir. Bunların karşılık geldiği Teknoloji Olgunluk Seviyesi (Technology Readiness Level, TRL) doküman içerisinde bulunabilir. Bunun yanında kontrat hazırlayıcılar ve takip edenler için çeşitli mihenk taşı noktalar da gösterilmiştir. Bu 10 aşama aşağıdaki gibi belirlenmiştir. (Anonim, 2018)

- MRL 1 : Temel İmalat Uygulamalarının Belirlenmesi
- MRL 2 : İmalat Konseptinin Tanımlaması
- MRL 3 : İmalat Konseptinin Doğrulanması
- MRL 4 : Teknolojinin Laboratuvar Ortamında Üretim Kapasitesinin Kazanılması
- MRL 5 : Üretim Benzeri Ortamda Prototip Parçaların Üretim Kapasitesi Kazanılması

- MRL 6 : Üretim Benzeri Ortamda Prototip Sistem veya Alt Sistemlerin Üretim Kapasitesi Kazanılması
- MRL 7 : Temsili Üretim Ortamında Sistem, Alt Sistemler veya parçaların Üretim Kapasitesi Kazanılması
- MRL 8 : Pilot Üretimin Yapılması, Düşük Kapasiteli Üretime Hazır Olunması
- MRL 9 : Düşük Kapasiteli Üretim Yapılması, Tam Kapasite ile Üretime Hazır Olunması
- MRL 10 : Tam Kapasite ile Üretimin Yapılması, Yalın Üretim Pratiklerine Hazır Olunması

3.2.6. Yeni parçanın imalat sistemine tanıtılması

MRL içerisindeki başlıklar projenin sahibini kontrol açısından ilgilendirirken üretimi gerçekleştirecek olan tesis ve sorumluları tarafından belli kısımlara yoğunlaşmaktadır. Bu çalışmada MRL tarafından belirtilen alanlar yeni ürünü bir tesise tanıtan NPI birimleri açısından değerlendirilecektir.

NPI yani yeni parça veya ürünlerin daha önceden oluşturulmuş ya da oluşturulmakta olan bir sisteme tanıtılarak yeni bir sistem kurulması sürecidir. Bu süreç doğru yapılmasının getireceği düşük maliyet avantajları ve hızlı yapılmasının getireceği rekabet avantajları ile öne çıkar.

Tarihsel sürece bakıldığında NPI sürecinin önem kazanması 1980’li yıllarda gerçekleşiyor: İkinci dünya savaşı sonrasında 1950-1960’lı yıllarda dünya çapında büyük oranlarda artan talep Avrupa ve Amerika’daki üreticilerin üretim sistemlerindeki pek de efektif olmayan kısımlarını görmesini engelliyor. Daha sonra Japonya’nın yeni bir rakip olarak ortaya çıkışı ile beraber bazı batılı şirketler, savaş öncesi ve sonrası yine batılı şirketler tarafından düşünülen fakat daha çok Japon rakipleri tarafından benimsenen ve uygulamaya alınan bazı fikirleri tekrar benimsemeye başlıyorlar. Fakat kalite artırma, genel giderleri düşürme, müşteri odaklı ürün geliştirme ve stok maliyetlerinin azaltılması gibi fikirlere yoğunlaşan bu şirketler büyük resmi kaçırıyorlardı. Yirminci yüzyılın sonlarında ortaya çıkan bu yeni rekabet aracı düşük maliyette ve yüksek kalitede yapılan hızlı ürün adaptasyonuydu (Backhouse ve Brookes, 1998). 1980’li yıllarda NPI süreçleri

birçok kurum ve firma tarafından araştırılmaya başlandı. 1985 yılında MIT tarafından 5 yıl sürecek ve beş milyon dolara mal olan bir araştırma başlatıldı. Bu çalışma sonunda "The Machine That Changed The World" isimli kitap yayınlandı. Bu kitapta önemli bir bölüm de Batılı şirketler ve Japon şirketleri arasındaki ürün adaptasyon yöntemleri farklılıklarına ayrıldı. (Backhouse ve Brookes, 1998)

NPI süreçlerine ilişkin birçok araştırma bu yıllardan sonra yayınlandı. Bu çalışmalarda önemi vurgulanan konulardan biri de iletişim ve bilgi sistemleridir. Bu konuda birçok firma ve kurum çeşitli çözümleri ile süreçlerini devam ettirmektedirler. Genellikle de ayrıık birçok sistem üzerinden çalışılmakta ve birimler arası çalışmalarda problemler yaşanmaktadır. Bu çalışmada NPI süreçlerini MRL kapsamındaki sorularla belirlediğimiz çeşitli aktivelerle ontoloji üzerinden ilişkilendirirken bu ontolojinin ileride yapılacak olan tüm sistemleri aynı ontoloji üzerinden birleştirme çalışmalarına ön ayak olması hedeflenmektedir.

4. UYGULAMA

Bir ontolojik modeli modellemek için, tekrar eden bir problemin farklı alanlardaki genel kullanımlarına ihtiyaç duyulur. (Gangemi, 2005)

Literatüre bakıldığında ontoloji ile modellemenin birçok farklı şekilde yapılabileceği görülebilir. Bu çalışma için tüm imalat alanı modellenmiştir. Aşağıda belirtilen soruları cevaplamaya yönelik bir ontoloji modeli yaratılmaya çalışılmıştır:

- Seçilen parçanın üretiminde MRL seviyesinin belirlenmesi için cevaplanması gereken sorular nelerdir, hangileri cevaplanmıştır, bunları cevaplamak için ulaşılması gereken alanlar nerelerdir?
- Üretim modelinin, sürecin yönetiminde kullanılması için gereken aktiviteler ve bunların birbirleri ve çeşitli kavramlarla ilişkilerini düzenleyeceği bağlamlar nelerdir?

Tüm alan ihtiyaca göre ilerleyen zamanda modellenenecek olmasına rağmen MRL alanının üzerine çalışabilmesi için bu alanın getirdiği sorular ile ilişkili kavramlar oluşturuldu. Birçok kavram sadece sınıf (class) olarak bırakıldı. Fakat istenilen uygulamaya göre ontolojiye ekleme yapılabilir ve kullanılmayan herhangi bir kavram üzerinde de değişiklik yapılabilir. Aktivite alanında daha odaklı olarak yeni parçanın sisteme adaptasyonu kısmı diğer sorumluluk alanlarına geçilmeden önce örnekleme ve sorunları görme amacı ile birçok ilişki kullanılıp tanımlamalar yapılarak birleştirildi. Bir mühendisin yapmaya hazır aktiviteler kısmında güncellenen aktiviteleri tamamlayarak bir projenin adaptasyonunu takip edebilmesi amacı güdüldü. Bu durum yapının geliştirilmesi mümkün ve kolay olduğu için ilerleyen zamanda denenerek geliştirilecektir. Bunun için potansiyel kullanım alanlarına ilişkin bir kısım çalışmanın ilerleyen kısmında sunulmuştur. Genel yapı aşağıda anlatıldığı şekilde kurulmuştur.

Kurulan yapıya göre, bir üretim tesisine ait genel bilgiler “ManufacturingFacility” sınıfı altında toplanmaya çalışılmıştır. Bu alan (domain) içinde tesis içerisindeki fiziki yapılar, ekipmanlar, standartlaştırılmış bilgiler gibi daha sonra parça imalatı sürecinde

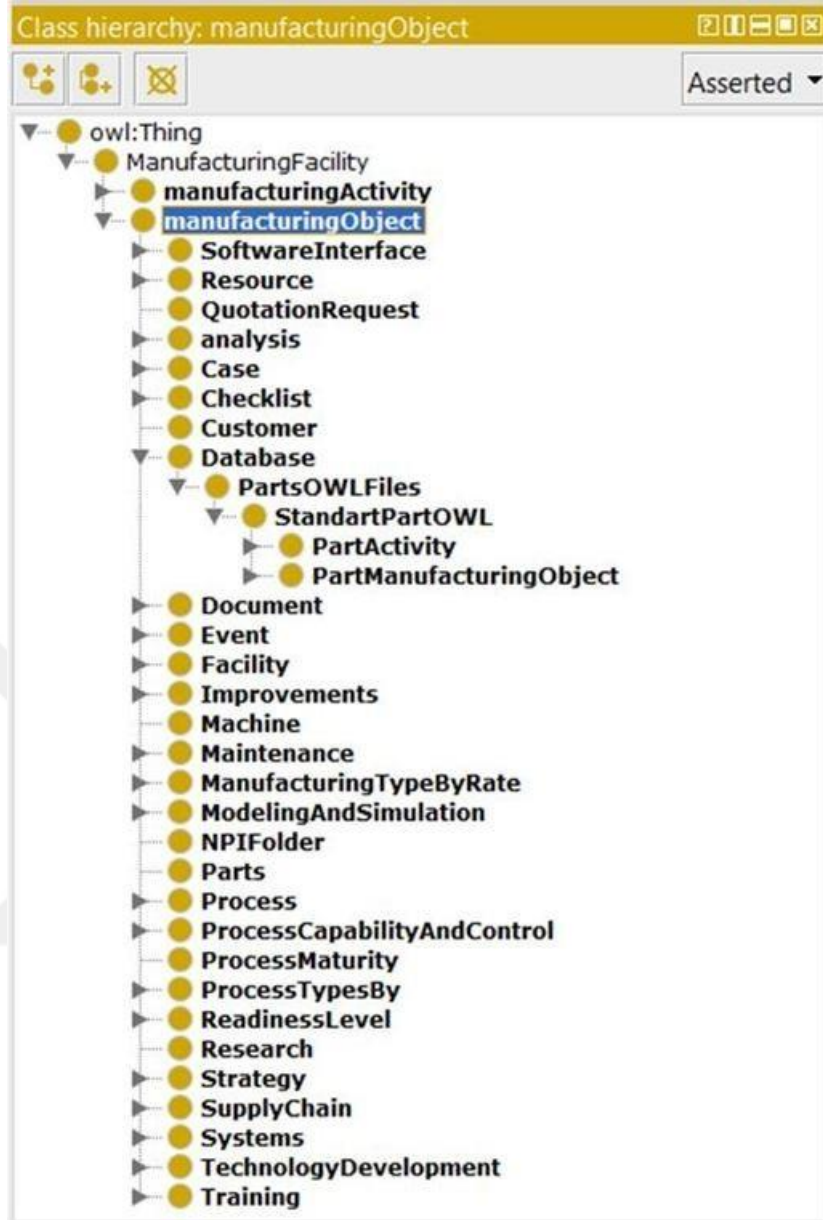
kullanılabilecek sınıflar, elemanları ve ilişkileri bulunması planlanmıştır. Bu bilgi kümesinin bilgiyi içermesi veya büyük boyutlu bilgilere kısayolu tutması gerekir. Başka çalışmalarda bunun üstüne bir genel imalat alanı da tanımlanabilir fakat bu çalışmada gerek duyulmamıştır. MRL sorgulamasında ihtiyaç duyulan kavramların sınıf olarak tanımlanması ile “ManufacturingFacility” sınıfı oluşturulmuştur. Bu sınıfa bilgi giriş çıkışını sürecin mimari yapısı ile ilgilenen bir sistem mimarının yapması umulmaktadır. Başka bir sınıfta ise “StandartPartOWL” bulunmaktadır. Bu dosyanın içindeki sınıflar ile beraber her parça için kopyalanarak süreç yürütücüsü tarafından depolanması, geliştirilmesi ve parça adaptasyon süreci ardından tekrar “ManufacturingFacility” ile birleştirilerek depolanması ve sorgular için kullanılması öngörülmektedir. Ontolojide bu, Sekil 4.1’de görüldüğü gibi, iki sınıf şeklinde biçimlenmiştir.



Şekil 4.1. “ManufacturingFacility” sınıfları

Her parça için kopyalanarak kurulduğunda kullanılacak parçaya ilişkin kayıtlar aşağıdaki gibi hiyerarşik yapıda gösterilmiştir. Fakat daha önce de bahsedildiği gibi bu yapı değiştirilebilir. Arayüzlerde değişiklik olmaması için olabildiğince sınıf isimleri ile ilişkilendirme yapılmıştır.

“StandartPartOWL” içindeki kısım her parça üretime girdiğinde başına parça adı gelecek ve “Standart” kısmı silinecek şekilde kopyalanarak süreç yöneticisinin arayüz yardımı ile değişiklikler yapmasına imkân sağlayacak şekilde yapılandırılmıştır. (Şekil 4.2)

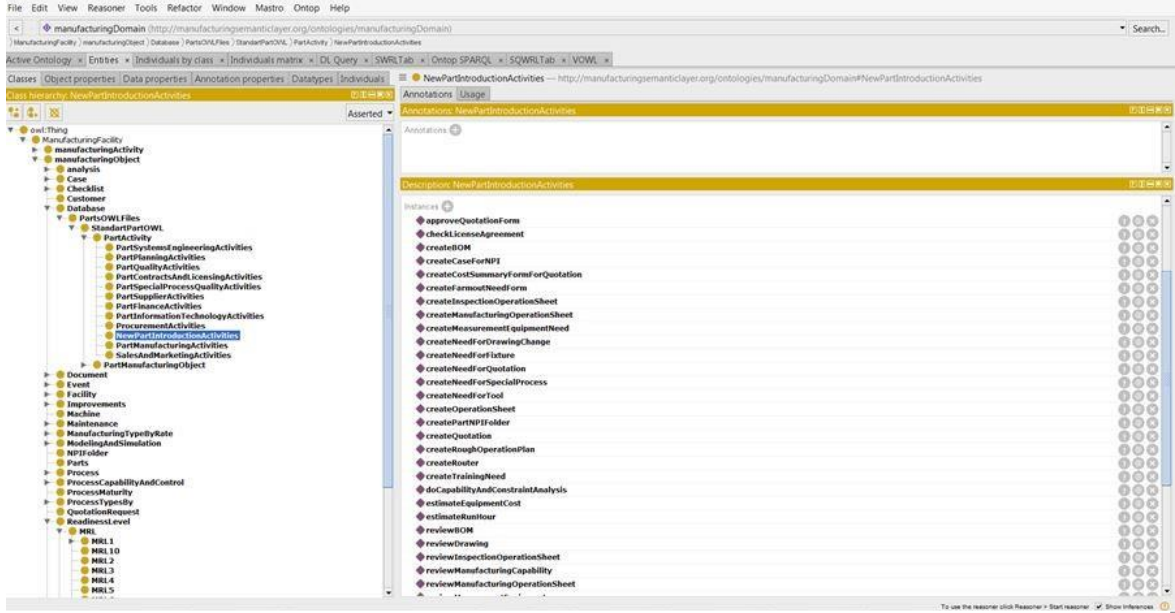


Şekil 4.2. “StandartPartOWL”

4.1. İmalat Aktiviteleri

Aktiviteler bu çalışma kapsamında hazırlanan modelin iki önemli unsurundan biridir. Kavramları, kavramların ve ilişkilerinin oluşturduğu modelleri, modellerin birleşerek oluşturduğu sistemleri aktiviteler yardımıyla birbirlerine bağlayarak proje yönetimi bakış açısıyla işler bir sistem oluşturmak mümkündür. Ontolojik olarak modellenmiş bir sistem kendi başına ontoloji editör programları yardımı ile zaman içinde değiştirilebilir fakat bu sistemi aktiviteler üzerinden yönetmek daha makul görülmüştür. Kurulan modelin projenin yönetiminde kolaylık sağlaması ve sonraki çalışmalarda proje

aktörlerinin insandan yazılımlara dönüşümünde kolaylık sağlaması açısından bu yaklaşım fayda sağlayacaktır. Bu sebeple bu kısma ağırlık verilmiştir. Bu çalışmada modellenmiş bir sürecin aktiviteleri ontoloji editörü ile Şekil 4.3'teki gibi görünmektedir.

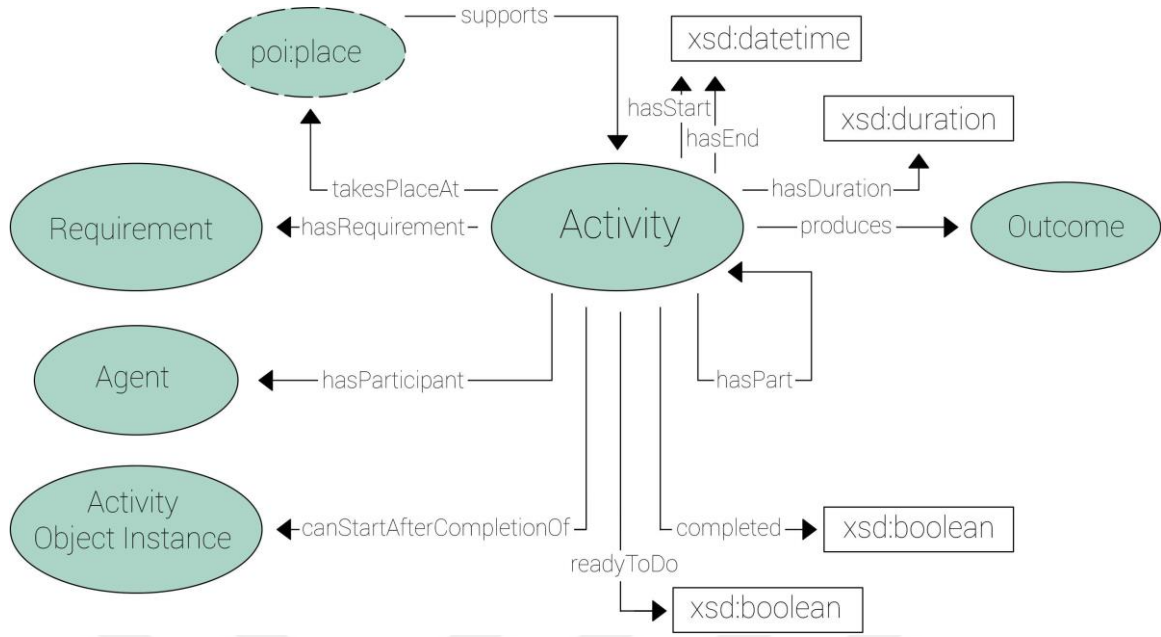


Şekil 4.3. Aktivite sınıfı bireyleri

4.2. İmalat Aktiviteleri için İlişkiler

Aktiviteler çoğunlukla şirketlerde belli gruplar sorumluluğunda yürütülmektedir. Bu çalışmada da aktiviteler bu sorumluluk isimleri altında toplanmıştır. Örneğin “quoting” aktivitesi bir birey (instance) olarak “SalesAndMarketing” sınıfı (class) altında yer almaktadır.

İlişkileri tanımlarken öncelikle Abdalla vd. (2014) tarafından yapılan bir çalışma kullanıldı. Bu çalışmanın belirttiği ilişkiler tanımlandı ve uygulamada kullanılacak bazı ilişkiler de eklenerek modele farklı bazı çıkarım kapasiteleri de kazandırıldı. Model, daha sonra eklenecek özellikler de bunlara uyumlu bir şekilde eklenebilir halde açıktır. Buna göre oluşan model aşağıdaki Şekil 4.4'te şema halinde gösterilmiştir. Tüm aktiviteler çeşitli ilişkilerle birbirine bağlanmış veya çeşitli değerler verilerek özellikleri belirtilmiştir.

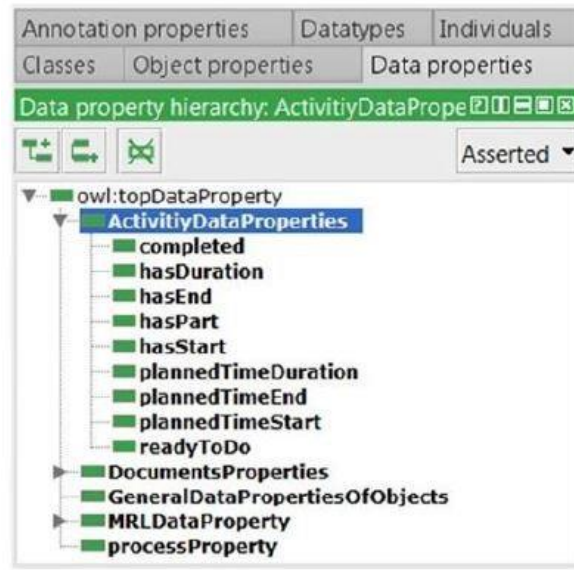


Şekil 4.4. Aktivite ilişkileri

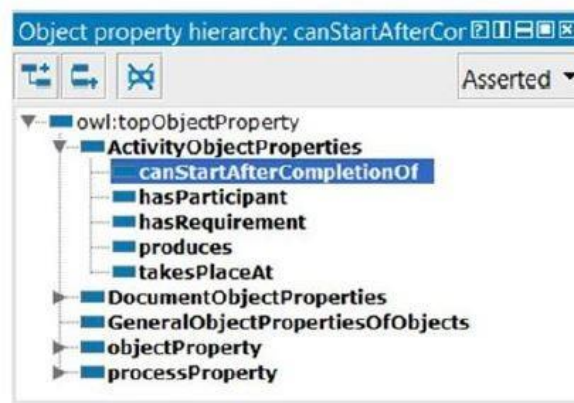
Aktiviteleri birbiri ile ilişkilendiren bağlantılar “topObjectProperty” altında toplanmıştır. Üzerine de sistem hangi yöne geliştirilmek istenirse eklemeler yapılabilir. Bu eklemelerden bazıları şunlardır:

- “hasStart” ilişkisi aktivitenin başlangıç zamanını temsil etmesi için bir “dataProperty” olarak tanımlanmıştır.
- “hasEnd” ilişkisi de aktivitenin bitiş zamanını temsil etmesi için bir “dataProperty” olarak tanımlanmıştır.
- “hasDuration” ilişkisi ise aktivitenin toplam süresini belirlemek adına bir “dataProperty” olarak tanımlanmıştır. (Şekil 4.5)

Aktiviteler için tanımlanan object properties yani kavramların bireylerine ilişkilendiren özellikler ise Şekil 4.6’da gösterilmiştir.



Şekil 4.5. Aktiviteler için veri tipi özellikler



Şekil 4.6. Aktiviteler için obje tipi özellikler

“hasPart” ilişkisi parça için kayıt edilen modelde varsayılan olarak yer alacak fakat her bir parça modelinde özelleştirilebilecektir. Bu ilişki yardımı ile raporlamalarda belli konulardaki aktivitelerin durumu takip edilebilecektir. “isPartOf” ilişkisi ile terstir. “hasPart” ilişkisine sahip her parça aynı zamanda bu üçlünün tersi olan “isPartOf” ilişkisine sahip olacaktır.

“hasPart” aktiviteleri başka bir genel aktivitenin altında birer alt aktivite olarak gösterir. Örneğin; “ActA hasPart ActB” üçlünün anlamı “ActA” aktivitesinin “hasEnd” özeliğinde belirtilen tarih ile “hasStart” özeliğinde belirtilen tarihlerin “ActB” tarihlerinin arasında olduğu tanımı yapılmıştır.

“canStartAfterCompletionOf” ilişkisi ile diğer aktivitenin bu bitmeden başlayamayacağı anlamı ontolojiye yansıtılmıştır. Uygulamada eğer bu ilişki ile bağlı tüm kavram ve aktivitelerde “completed” özelliği “true” ise aktivitenin “readyToDo” özelliği de “true” olmaktadır.

“completed” ilişkisi ise aktivitenin tamamlandığını belirtir. xsd:boolean olarak tanımlanmış bir özelliktir. Yani ya true olarak değer alır ve tamamlandığı anlamına gelir ya da false değeri alarak tamamlanmadığını bildirir.

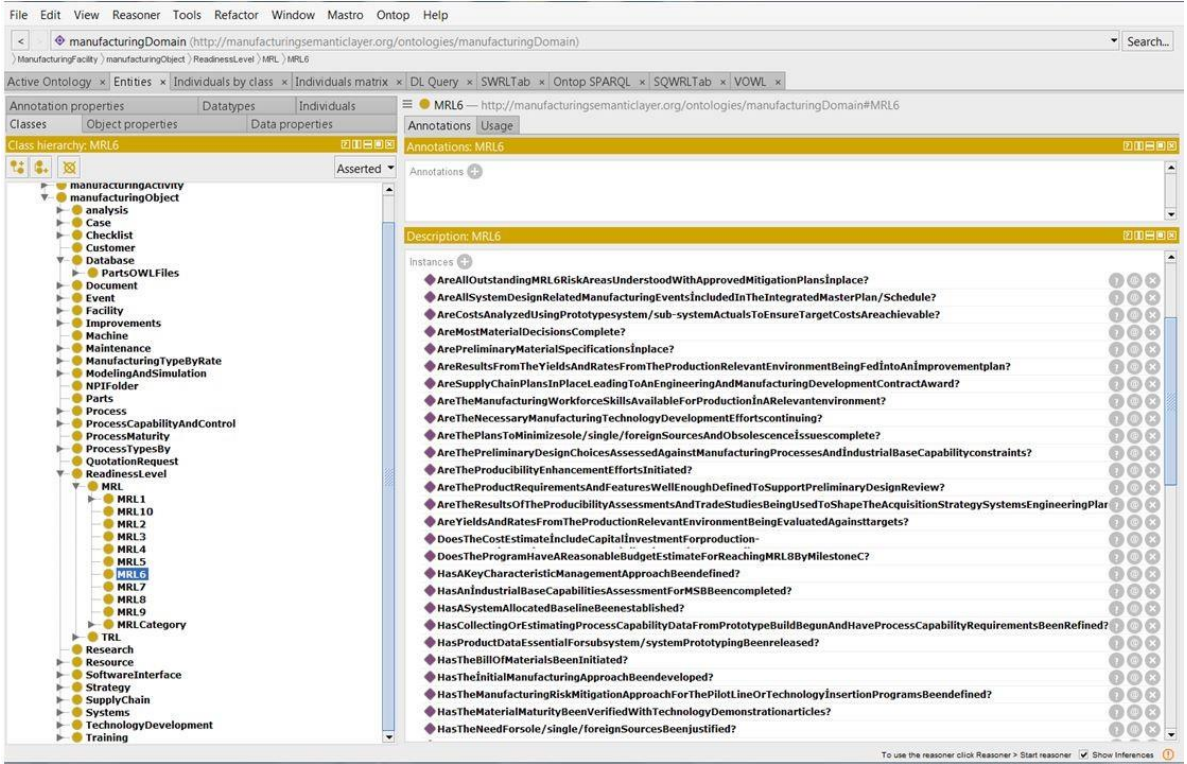
“readyToDo” ilişkisi ise aktivitenin yapılmak için hazır olması anlamına gelmesi için tasarlandı. Buna göre bir aktivitenin “canStartAfterCompletionOf” ilişkisi ile bağlı olduğu tüm aktiviteler “completed” özelliği “true” olarak belirlenmişse bu özellik de “true” olarak değişecektir.

4.3. İmalat Olgunluk Seviyesi Soruları

MRL bu çalışmada genel bir çerçeve olarak kullanılmıştır. Bu çerçevede farklı modeller üretilebilir. Farklı bakış açıları ile aynı model kullanılarak süreç ile ilgili veri yönetimi sağlanabilir.

Modelleme çalışmasında aynı kaynak tarafından hazırlanan ve sorulardan oluşan bir çizelge kullanılmıştır. Tüm seviyeler için sorulması gereken sorular ontoloji içerisine eklenmiştir. Daha sonra bu sorulara istinaden imalat alanı içinde yer alacak gerekli kavramlar ontolojiye eklenmiştir. Sorunun hangi kavramlar ile ilişkili olduğu oluşturulan üçlüler yardımı ile ontoloji içine tanımlanmıştır. Bunlar arayüz yardımı ile soru ile ilişkili kavramlara yönlendirme sağlayabilecektir. Daha sonra yapılabilecek yazılımlar yardımı ile çoğu sorunun ilişkilerde belirtilen alanlardan otomatik cevaplar alıp son kararı ontolojiyi düzenleyene bırakması hedeflenmektedir.

MRL başlığı altındaki soruların hepsi birer individual yani birey olarak ontolojide yer almıştır. Bu sorular hangi MRL adımına ait ise o sınıf altında gösterilmişlerdir. Ayrıca konularına göre de bir sınıflama yapılarak aynı soru o sınıfların altında da yer almıştır. Hangi konularda gereklilikler tamamlandı gibi bir sorguya böylelikle cevap verilebilir.



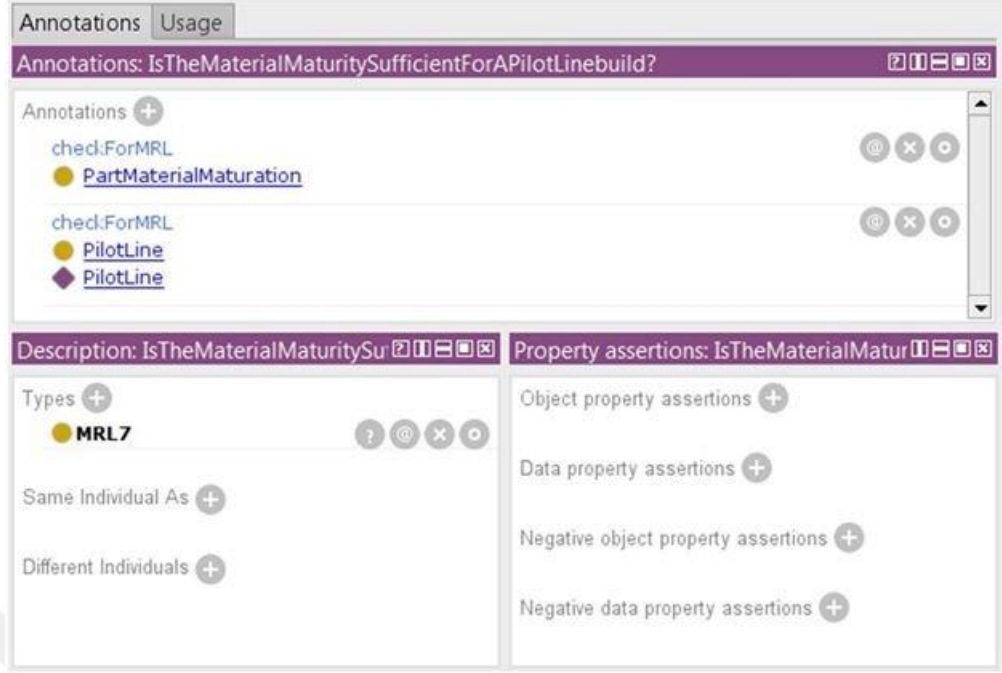
Şekil 4.7. MRL adımlarındaki soruların bireyler olarak gösterim örneği

Şekil 4.7’de örneği görüldüğü gibi çeşitli adımlardaki ilgili sorular birer birey (instance) olarak hepsi birer sınıf (class) ile gösterilmiş olan ilgili adım altında toplanmıştır. Yaklaşık 450 civarında soru ilgili adımlara girilmiştir ve her yeni parçada özellikleri değiştirilerek kullanılabilir olacaktır.

4.4. İmalat Olgunluk Seviyeleri İçin İlişkilendirmeler

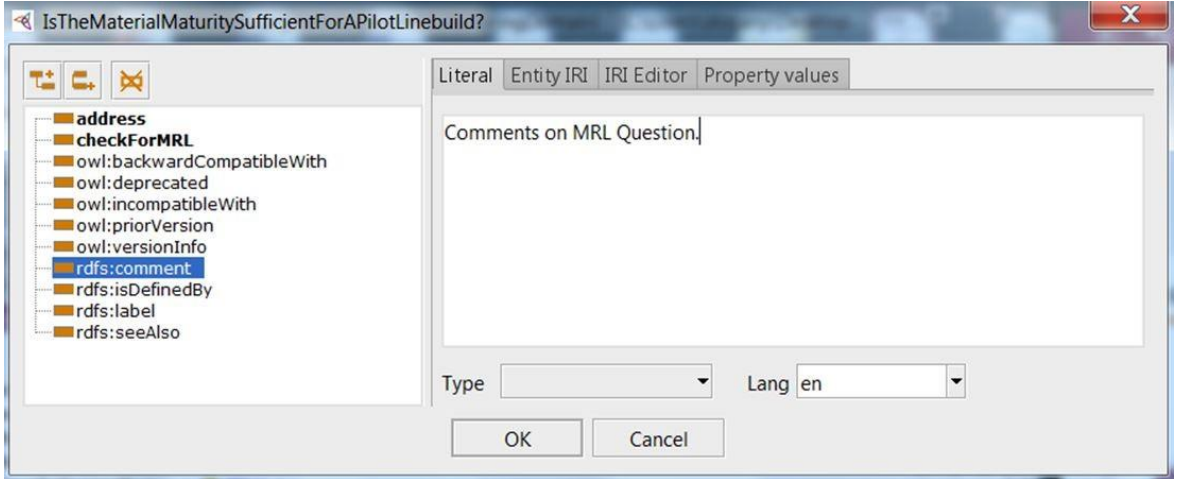
MRL için bu çalışmada şimdilik 3 farklı ilişki tanımlanmıştır.

Bunlardan ilki ve en önemlisi “checkForMRL” ilişkisidir (Şekil 4.8). Bu ilişki ile tüm MRL bireyleri yani soruları bazı konular ile ilişkilendirilmiştir. Bunu yaparken Protege ontoloji editöründe “annotationProperty” özelliği kullanıldı. Bu şekilde yapılmasının sebebi bir “Individual” etiketli noktanın bir sınıfa özne-bağ-nesne şeklinde ancak sınıfın nesne olduğu bir biçimde bağlanmasının mümkün olmasıdır.



Şekil 4.8. “checkForMRL” ilişkisi ile gösterilmiş üçlüler

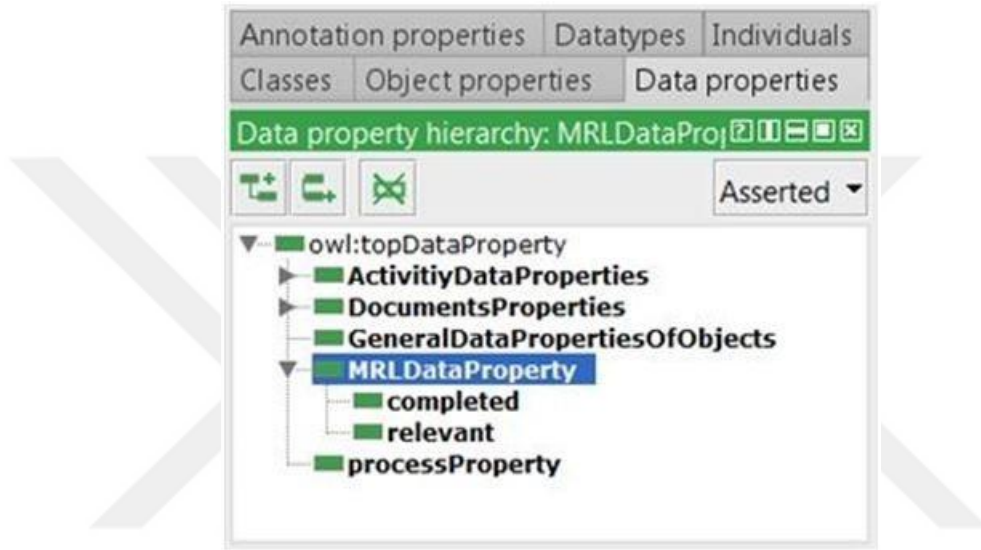
Bu çalışmada kullanılan başka bir “annotationProperty” olan RDFS dili dahilinde kullanılabilen “rdfs:comment” ilişkisidir. (Şekil 4.9)



Şekil 4.9. “rdfs:comment” ilişkisi

Çeşitli MRL adımlarında bulunan soruların cevaplanıp cevaplanmadığını anlamak amacı ile “completed” ilişkisi kullanılmıştır. “xsd:boolean” cinsinden “true” veya “false” olarak değer alır. Arayüz içerisinde “Open/Closed” şeklinde gösterilmiştir. (Şekil 4.10)

Bir başka “data property” yani veri olarak özellik ise “relevant” ilişkisi olarak tanımlanmıştır. Bu ilişki de boolean olarak değer alır. Arayüzde “Relevant/Irrelevant” olarak gösterilmiştir. (Şekil 4.10)

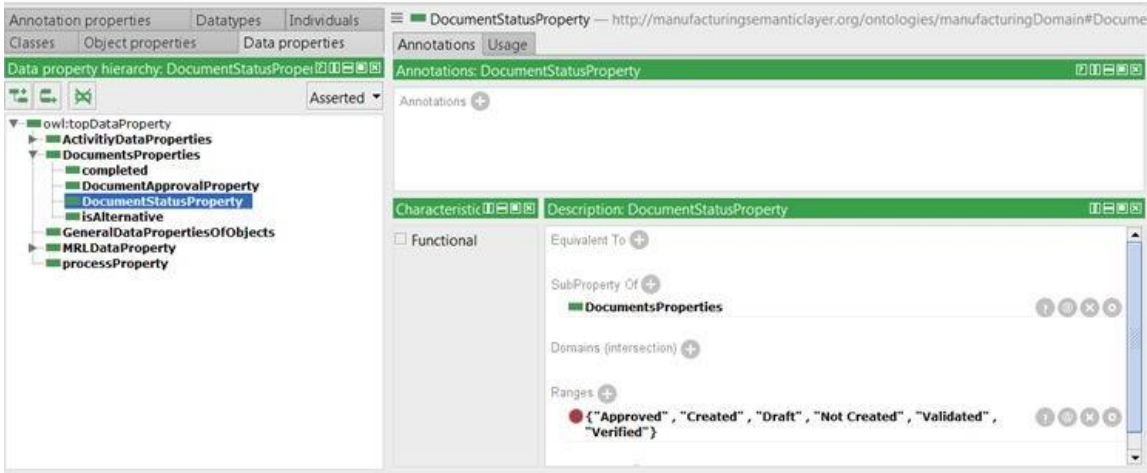


Şekil 4.10. MRL Veri ilişkileri

4.5. Kavramların Kullanımı

Bu çalışmada yaratılan kavramlar MRL soruları baz alınarak yaratılmasına rağmen birçok farklı çalışmadan ve tecrübeden bazı kavramlar da eklendi. Fakat sistemin geliştirileceği yönler doğru ve mevcut işletmenin veri yapısına göre yeni kavramlar eklenmesi, mevcut olanların değiştirilmesi için sistem açık ve genişletilebilir bir şekildedir.

Örnek olarak aşağıda verilmiş olan “documents” kavramı incelenebilir. Bu kavram üretim süresince üretilen dokümanlar için yaratılmış bir sınıftır. Bu sınıf altında yer alan bireyler, yani dokümanlar, için tanımlanan veri tipi bir özellik olan “DocumentStatusProperty” aşağıda görüldüğü gibi tanımlanmıştır. Veri tipi özelliklerde tanımlanabilecek değer aralığı aşağıdaki ontoloji editörü ekran görüntüsünde görüldüğü gibi “Ranges” başlığı altında görülebilir. (Şekil 4.11)



Şekil 4.11. “DocumentStatusProperty”

4.6. Jena ile Kodlanan Dosyalar

Jena kütüphanesi kullanılarak 5 arayüz için 5 dosya hazırlanmıştır. Bu dosyalar arayüzün görünüşleri için SWT kütüphanesi yardımı ile kurulan dosyalara ilişkilendirilmişlerdir:

Bu dosyalar “_ProjectMain”, “_Main”, “_ActivityTemplate”, “_MRLTemplate” ve “_ObjectTemplate” isimleriyle yazılmışlardır. Ek açıklamalarda bu kodlardan bazı örnekler verilmiştir. Bunlardan biri Ek Açıklamalar A.1’de verilen yapılmaya hazır aktiviteleri görüntüleme için yazılan koddan bir kısımdır. Tüm aktivitelerin “readyToDo” özelliğini kontrol eder ve olumlu olanları listeler. Ek Açıklamalar A.2’de ise aktivitelerin farklı özelliklerine erişmeyi sağlayan metot verilmiştir. Ek Açıklamalar A.3 de imalat olgunluk seviyesi sorularının özelliklerini görüntüleyen bir metottur. Kavramların özelliklerini görüntüleyen metot ise Ek Açıklamalar A.4 içerisinde verilmiştir. Ek Açıklamalar A.5 ise modelde arayüz yardımı ile değişiklik yapmayı amaçlayan bir kod kesitidir. Değişiklikleri örneklemek amaçlı ek açıklama kısmında gösterilmiştir.

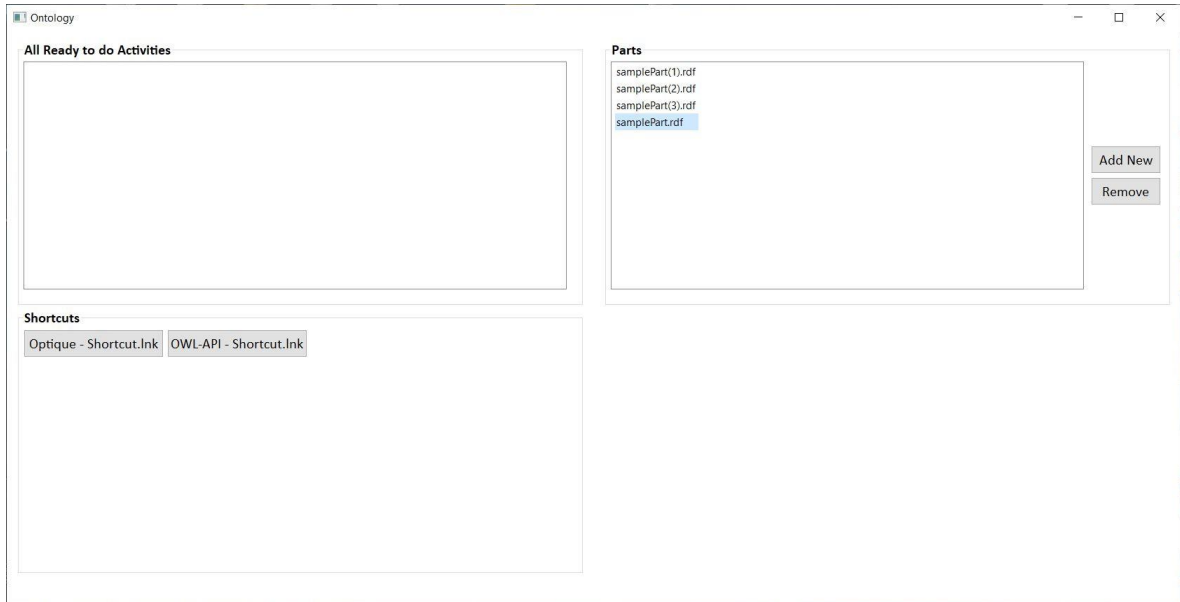
4.7. Uygulama Programı

Ontolojiler için kullanım yerine göre genel ve özelleşmiş birçok arayüz çalışması yapılmıştır. Bu çalışmada ise odaklanılan konulara özgü bir arayüz hazırlanmıştır.

Java dilinde bu programı hazırlamak için açık kaynaklı olarak geliştirilen bir bütünleşik geliştirme ortamı (Integrated Development Environment, IDE) olan Eclipse kullanılmıştır. Bu uygulama 2001 yılında IBM tarafından ortaya çıkarılmıştır ve sürekli olarak güncellenmektedir. Özellikle java dili ile ilişkide olan uygulamalarda oldukça başarılıdır. Apache Jena kütüphanesi kullanılarak “example.rdf” dosyalarına erişim sağlayacak kodlama yapılmıştır. Bu kodlar ekte de sunulmuştur. Bunlar yardımı ile arayüz için yazılan kodlar çalışmakta ve kullanıcı arayüz programı çalışmaktadır. Arayüz için ise yine java uygulamaları için kullanılan SWT kütüphanesi kullanılmıştır. Bu kodlar da ek açıklamalarda sunulmuştur.

4.7.1. Proje arayüzü

Bu arayüzde herhangi bir kullanıcının ana arayüzü olarak çalışacaktır. Tüm parçalara ilişkin yapılmaya hazır aktivitelerin toplandığı “All Ready To Do Activities” alanı, aktiviteleri ve olgunluk seviyeleri takip edilen parçaların ana arayüzlerine ulaşmak için “Parts” alanı ve kullanıcının üzerinde çalıştığı herhangi bir projeye ait kısayollarını ekleyebileceği bir “Shortcuts” alanı ile oluşturulmuştur. (Şekil 4.12)



Şekil 4.12. Proje arayüzü

4.7.2. Parça arayüzü

Bu arayüzde parça ile ilgili aktivitelere ve olgunluk seviyelerine ulaşmak mümkündür. ReadyToDo ilişkisi ile yapılmaya hazır halde, yani öncesinde yapılması gereken işler completed olarak tanımlanmış süzülerek readyToDo özelliği “true” olarak değiştirilmiş işler ilk hanede listelenerek karşımıza çıkacaktır. Bu aktivitelerden herhangi birine tıklanarak bunların, aktivitelere ait özelliklerin yer aldığı aktivite sayfasında gösterilmesi sağlanabilmektedir.

İkinci kısımda MRL seviyeleri tek tek sıralanmıştır. Bu seviyelere ilişkin soruların, bu soruların cevaplanıp cevaplanmadığına ilişkin bilgilerin ve bu sorularla ilişkili kavramlara ulaşım sağlayan linklerin yer aldığı MRL sayfasına yönlendirme sağlamak için görüntülenmek istenilen seviye üzerine tıklanabilir. (Şekil 4.13)



Şekil 4.13. Parça arayüzü

4.7.3. Aktiviteler arayüzü

Bir parçaya bir ait aktivitenin özelliklerinin görüntülediği bir ekran olarak tasarlanmıştır. Bahsedilen tüm ilişkilerin bilgilerini yansıtır. “Sub Activities” kısmı yanındaki ekleme ve çıkarma butonları seçili aktivitenin alt aktivitelerini ekleyip çıkarmak için kullanılacaktır. “Activity Requirements” kısmının yanındaki ekleme ve çıkarma butonları ise aktivenin yapılmaya hazır durumda olup olmadığını belirleyen aktivitelerin bu aktiviteye ilişkilendirilip çıkarılması amacı ile kullanılacaktır. (Şekil 4.14)

The screenshot displays the 'Activity Template' window for an activity instance named 'examinePartDrawing'. The interface is divided into several sections:

- Activity Instance Name :** examinePartDrawing
- Super Activities:** A large empty box for listing super activities.
- Sub Activities:** A list containing 'examinePartList' with a checkbox. To the right are 'Add New' and 'Remove' buttons.
- Activity Information:** A table showing start, end, and duration information.

Start	N/A
End	N/A
Duration	N/A
- Ready To Do :** YES
- Completed :** NO
- Activity Requirements:** A list containing 'createQuotation' with a checkbox. To the right are 'Add New' and 'Remove' buttons.

Şekil 4.14. Aktiviteler arayüzü

4.7.4. İmalat olgunluk seviyeleri arayüzü

Seçilen parçanın seçilen MRL adımı ile ilgili süreç yöneticisine yöneltilen soruların bulunduğu bir arayüz açılır. Soruların her biri için eklenmiş tüm ilişkiler ve bilgiler yer alıyor. Bu soruların parça için alakasız olup olmadığını süreç yöneticisinin belirlemesi

amacıyla “Relevant/Irrelevant”, soru ile ilgili aksiyonların tamamlanıp tamamlanmadığı belirlemek amacı ile de “Open/Closed” butonları ve aksiyonlar eklendi. (Şekil 4.15)

The screenshot displays a web application window titled "MRL Template". Inside, there is a section titled "List name : MRL6". Below this, five questions are listed, each with a set of buttons for actions and status.

Question	Question Text	Buttons
Question 1	HavePlansForTheIdentifiedManufacturingFacilitiesBeenDevelopedToProduceThePilotLinebuild? PartPilotLinePlan	Closed Relevant
Question 2	HasTheBillOfMaterialsBeenInitiated? PartBOM	Closed Irrelevant
Question 3	HaveManufacturingFacilitiesBeenIdentifiedToProduceThePilotLinebuild? PartManufacturingStrategy PartLocation	Closed Relevant
Question 4	AreResultsFromTheYieldsAndRatesFromTheProductionRelevantEnvironmentBeingFedIntoAnImprovementplan? PartInspectionInProcess PartInspectionFinal	Closed Relevant
Question 5	HaveInitialRequirementsBeenIdentifiedForAcceptanceTestProceduresAndIn-processAndFinalInspectionRequirementsForEMDunits? PartInspectionInProcess PartInspectionFinal	Open Irrelevant

Şekil 4.15. İmalat olgunluk seviyeleri arayüzü

4.7.5. Kavramlar arayüzü

Bu arayüzde tanımlanmış kavramlar ve ilişkilerini görmek mümkündür. Bu ekrandaki alt ve üst kavramlara ilgili kısımlardan üzerlerine tıklayarak ulaşılabilir. Her bir kavramın bireylerini de ilgili bölümde görmek ve bunlara yenilerini eklemek veya var olanları çıkarmak bu hanenin yanındaki butonlar sayesinde yapılabilir. (Şekil 4.16)

Object Template

Object or Instance Name : PartManufacturingStrategy

Comments

Parça imalat Stratejisi belirlleme çalışması ön çalışmaları oluşturulmuştur.@tr
It has decided to create a in-house manufacturing strategy.@en

Super Classes

PartStrategy

Sub Classes

PartPrototypeProduction
PartPilotLine
PartLowRateInitialProduction
PartFullRateProduction

Object Individuals

☐ PartInitialManufacturingStrategy

Add New
Remove

Object Relations

PartManufacturingStrategy comment Parça imalat Stratejisi belirlleme çalışması ön çalışma...
PartManufacturingStrategy comment It has decided to create a in-house manufacturing str...
PartManufacturingStrategy comment Parça imalat Stratejisi belirlleme çalışması ön çalışma...
PartManufacturingStrategy comment It has decided to create a in-house manufacturing str...
PartManufacturingStrategy comment It has decided to create a in-house manufacturing str...
PartManufacturingStrategy subClassOf http://manufacturingsemanticlayer.org/ontologies/...
PartManufacturingStrategy type http://www.w3.org/2002/07/owl#Class

PartManufacturingStrategy comment Parça imalat Stratejisi belirlleme çalışması ön çalışma...
PartManufacturingStrategy comment It has decided to create a in-house manufacturing str...
PartManufacturingStrategy comment Parça imalat Stratejisi belirlleme çalışması ön çalışma...
PartManufacturingStrategy comment It has decided to create a in-house manufacturing str...
PartManufacturingStrategy subClassOf http://manufacturingsemanticlayer.org/ontologies/...
PartManufacturingStrategy type http://www.w3.org/2002/07/owl#Class

Şekil 4.16. Kavramlar arayüzü

5. SONUÇ VE ÖNERİLER

Üretimde bilgi sistemlerinde ontolojilerin önemi hiyerarşisiz, birlikte çalışabilir sistemlere olan ihtiyaç ve standardizasyonun önem kazanmasıyla kuşkusuz artacaktır. Havacılık için üretim yapan sektörler gibi bilgiye bağımlı sektörlerde bu önem katlanarak etki gösterecektir. Çıkarımlar yapabilen ve birlikte çalışabilen sistemler de bilgiye ulaşım imkânları arttıkça insan kaynakları tarafından yapılan ve asıl işe değer katmayan birçok işi gözlemci ile veya gözlemci olmadan yapabilir hale geleceklerdir. Bilgi yönetimi sistemlerinin bu şekilde gelişmesi için ontolojiler gibi bilgiyi modelleme kullanılabilen yapıların bu sistemlerin merkezinde yer alması gerekecektir.

Havacılıkta imalat, birçok farklı alandan ileri teknolojilerin kullanıldığı atölye tipi imalat sistemlerinin çoğunlukla kullanıldığı dolayısı ile birçok değişik alanda ancak uzmanları ile yürütülebilecek süreçleri kapsayan bir alandır. Gaz türbinleri gibi motorların parçaları, çoklukla genel malzemelerden ziyade özel malzemelerden ve geometrik şekilleri oldukça karmaşık şekilde tasarlanabilirler. Bunu yanında mikro yapılarını ve yüzey özelliklerini tayin edecek birçok farklı yöntem de imalat tasarımlarında yer alabilir. Geleneksel imalat yöntemleri ve ileri yöntemler birlikte kullanılarak imalatı yapılan parçaların, prototip imalatından yüksek adetlerde üretimine kadar geçen süreç de birçok sektörden daha fazla gereksinime sahiptir. Bu durum da parçaların imalatı esnasında elde edilen, kullanılan ve saklanan bilginin niteliğinin yüksek olmasını zorunlu kılar. Bu çalışmada havacılıkta üretimi yapılan motor parçalarının üretim sistemine adaptasyonuna odaklanan, metin bazlı MRL dokümanları kullanılarak, üretimdeki aktivitelerin kontrolü ve süreç takipçisinin yönlendirilmesi için kullanılacak bir sistem hazırlanmıştır. İlerleyen çalışmalarda genişletilebilecek bir ontoloji oluşturulmuş ve bu sistemin kontrolü için de 5 arayüzden oluşan bir program oluşturulmuştur.

Bu sistem kullandığı ontoloji altyapısı itibarı ile önemlidir. Aynı dili kullanarak modellenebilecek birçok sistem vardır. Standartlar ile tespiti yapılan bilgiler (parça üç boyutlu modeli gibi), sistem dışından gelecek bilgiler (takım, fiktür, malzeme, sarf malzemeleri ile ilgili bilgiler gibi), alan içinde üretilen bilgiler (performans parametreleri, öngörü sistemleri, en iyi uygulamalar gibi) ve üretim alanında ihtiyaç duyulabilecek her

türlü bilgi modellenip saklanıp ilişkilendirilebilir. Bu da hem bilginin yerine ulaşması hem nitelikli saklanması ve zamanı geldiğinde kolaylıkla ulaşılması açısından fayda sağlar.

Bu çalışmada kullanılan bilgi depolama aracı ontoloji ve RDF formatında seçildiği için sistemin üzerine yeni bir model inşa etmek ve bunu diğerlerine bağlamak kolay bir şekilde yapılabilecektir. Oluşturulmuş sistemi kullanım esnasında büyütmek ve geliştirmek öncelikli hedef olarak belirlenmiştir. Dolayısıyla oluşturulan kavramları büyütmek ve geliştirmek diğer kavramlarla ilişkilerini oluşturmak yoluyla yeni modelleri ontoloji üzerine kurmak devam eden çalışmalarımızdır.

Bilinen birçok endüstri bazlı ontoloji geliştirme çabasına katkıda bulunacağına inanılan bu çalışma ileride küresel ilişkili bilgi yaklaşımlarına da birleştirilebilecek bir sisteme dönüştürülmeye çalışılacaktır.

KAYNAKLAR DİZİNİ

- Abdalla, A., Hu, Y., Carral, D., Li, N., & Janowicz, K., 2014, An ontology design pattern for activity reasoning. CEUR Workshop Proceedings.
- Alistair Miles, S. R. A. L. / U. of O., & Sean Bechhofer, U. of M., 2010, SKOS Simple Knowledge Organization System Reference. W3C Recommendation 18 August 2009.
- Anonim, 2018, Manufacturing Readiness Level (MRL) Deskbook Versi 2.4., http://www.dodmrl.com/MRL_Deskbook_2018.pdf
- Apache Jena, 2011, Apache Jena Homepage. Apache Jena [Online]. Available at: <http://Jena.Apache.Org/> [Accessed December, 2019].
- Arenas, M., Bertails, A., Prud'hommeaux, E., & Sequeda, J. F. Eds., 2011, A Direct Mapping of Relational Data to RDF. W3C Working Draft 20 September 2011.
- Backhouse, C., Brookes, N., 1998, Concurrent Engineering: What's Working where
- Berners-Lee, T., Hendler, J., & Lassila, O., 2001, The semantic web. Scientific American, pp. 28–37. <https://doi.org/10.1038/scientificamerican0501-34>
- Bruno, G., Antonelli, D., & Villa, A., 2015, A reference ontology to support product lifecycle management. Procedia CIRP. <https://doi.org/10.1016/j.procir.2015.06.009>
- Chungoora, N., & Young, R. I. M., 2011, The configuration of design and manufacture knowledge models from a heavyweight ontological foundation. *International Journal of Production Research*. <https://doi.org/10.1080/00207543.2010.504754>
- Cutting-Decelle, A. F., Young, R. I. M., Michel, J. J., Grangel, R., Le Cardinal, J., & Bourey, J. P., 2007, ISO 15531 MANDATE: A product-process-resource based approach for managing modularity in production management. Concurrent Engineering Research and Applications, 15. <https://doi.org/10.1177/1063293X07079329>
- Despres, C., & Chauvel, D., 2012, Knowledge horizons: The present and the promise of knowledge management. In Knowledge Horizons: The Present and the Promise of Knowledge Management. <https://doi.org/10.4324/9780080496016>
- DIN. DIN SPEC 91345:2016-04, Reference Architecture Model Industrie 4.0 (RAMI4.0); DIN; Berlin, Germany, 2016
- Efstratios, A., 2015, Semantic Based Virtual Environments for Product Design, Diploma thesis, University of Patras, Computer Engineering and Informatics Department, 70

p.

KAYNAKLAR DİZİNİ (devam)

- Fox, M. S., 1992, The TOVE project: Towards a common-sense model of the enterprise. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). <https://doi.org/10.1007/bfb0024952>
- Frankovic, B., Budinska L., and Dang, T.T., 2002, Creation of Ontology for Planning and Scheduling, 3rd International Symposium of Hungarian Researchers on Computational Intelligence, Budapest, Hungary
- Gaines, B. R., Norrie, D. H., & Lapsley, A. Z., 1995, Mediator: an intelligent information system supporting the virtual manufacturing enterprise. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 964–969. <https://doi.org/10.1109/icsmc.1995.537892>
- Gangemi, A., 2005, Ontology design patterns for semantic web content. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 262–276. https://doi.org/10.1007/11574620_21
- Gómez-Pérez, A., Fernández-López, M., Corcho, O., & Gomez-Perez, A., 2010, Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. In with examples from the areas of knowledge management, e-commerce and the Semantic Web.
- Gruber, T. R., 1993, A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199–220. <https://doi.org/10.1006/knac.1993.1008>
- Guarino, N., 1998, Formal Ontology and Information Systems. Formal Ontology in Information Systems: Proceedings of the 1st International Conference. <https://doi.org/10.1.1.29.1776>
- Horrocks, I., 2013, What are ontologies good for? In *Evolution of Semantic Systems* (pp. 175–188). <https://doi.org/10.1007/978-3-642-34997-3>
- IIC., 2019, The Industrial Internet of Things Volume G1: Reference Architecture. Industrial Internet Consortium. <https://doi.org/10.1145/1514095.1514110>
- ISO/IEC. ISO 16100-6:2011(E): Industrial Automation Systems and Integratio- Manufacturing Software Capability Profiling for Interoperability; International Organization for Standardization: Geneva, Switzerland, 2015; Volume 2011.
- Kifer, M., & Boley, H. , 2013, RIF Overview (Second Edition).
- Klyne, G., & Carroll, J. J., 2004, Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. In W3C

KAYNAKLAR DİZİNİ (devam)

- Leitão, P., & Restivo, F., 2006, ADACOR: A holonic architecture for agile and adaptive manufacturing control. Computers in Industry. <https://doi.org/10.1016/j.compind.2005.05.005>
- Lemaignan, S., Siadat, A., Dantan, J. Y., & Semenenko, A., 2006, MASON: A proposal for an ontology of manufacturing domain. Proceedings - DIS 2006: IEEE Workshop on Distributed Intelligent Systems - Collective Intelligence and Its Applications. <https://doi.org/10.1109/DIS.2006.48>
- Lu, Y., Panetto, H., Ni, Y., & Gu, X., 2013, Ontology alignment for networked enterprise information system interoperability in supply chain environment. International Journal of Computer Integrated Manufacturing. <https://doi.org/10.1080/0951192X.2012.681917>
- Mahmudi, K., Liem, M. M. I., & Akbar, S., 2018, Ontology to relational database transformation for web application development and maintenance. Journal of Physics: Conference Series. <https://doi.org/10.1088/1742-6596/971/1/012031>
- Mas, F., Arista, R., Oliva, M., Hiebert, B., Gilkerson, I., & Rios, J., 2015, A Review of PLM Impact on US and EU Aerospace Industry. Procedia Engineering, 1053–1060. <https://doi.org/10.1016/j.proeng.2015.12.595>
- Matsokis, A., & Kiritsis, D., 2010, An ontology-based approach for Product Lifecycle Management. Computers in Industry. <https://doi.org/10.1016/j.compind.2010.05.007>
- Nolin, J., & Olson, N., 2016, The Internet of Things and convenience. Internet Research, 26, 360–376. <https://doi.org/10.1108/IntR-03-2014-0082>
- Noy, N. F., & McGuinness, D. L., 2001, Ontology Development 101: A Guide to Creating Your First Ontology. In Stanford Knowledge Systems Laboratory. <https://doi.org/10.1016/j.artmed.2004.01.014>
- Petnga, L., & Austin, M., 2016, An ontological framework for knowledge modeling and decision support in cyber-physical systems. Advanced Engineering Informatics. <https://doi.org/10.1016/j.aei.2015.12.003>
- Ray, S. R., & Jones, A. T., 2006, Manufacturing interoperability. Journal of Intelligent Manufacturing. <https://doi.org/10.1007/s10845-006-0037-x>
- Schlenoff, C. I., Gruninger, M., Tissot, F., Valois, J., Lubell, J., & Lee, J. W., 2000, The Process Specification Language (PSL) Overview and Version 1.0 Specification. NIST Interagency/Internal Report (NISTIR).

KAYNAKLAR DİZİNİ (devam)

- Sikos, L. F., 2015, Mastering Structured Data on the Semantic Web. In Mastering Structured Data on the Semantic Web. <https://doi.org/10.1007/978-1-4842-1049-9>
- Thoben, K. D., Wiesner, S. A., & Wuest, T., 2017, “Industrie 4.0” and smart manufacturing-a review of research issues and application examples. International Journal of Automation Technology. <https://doi.org/10.20965/ijat.2017.p0004>
- Uschold, M., King, M., Moralee, S., & Zorgios, Y., 1998, The enterprise ontology. Knowledge Engineering Review. <https://doi.org/10.1017/S0269888998001088>
- Usman, Z., Young, R. I. M., Chungoora, N., Palmer, C., Case, K., & Harding, J., 2011, A manufacturing core concepts ontology for product lifecycle interoperability. Lecture Notes in Business Information Processing, 5–18. https://doi.org/10.1007/978-3-642-19680-5_3
- W3C., 2014, RDF Schema 1.1. In World Wide Web Consortium.
- W3C OWL Working Group., 2012, OWL 2 Web Ontology Language Document Overview. OWL 2 Web Ontology Language.
- Zeid, A., Sundaram, S., Moghaddam, M., Kamarthi, S., & Marion, T., 2019, Interoperability in smart manufacturing: Research challenges. Machines. <https://doi.org/10.3390/machines7020021>

EK AÇIKLAMALAR

Ek Açıklama -A: Jena kütüphanesi kullanılarak hazırlanan kodlar

Ek Açıklama-A.1 “readyToDo” aktiviteleri görüntüleme için yazılan koddan bir kısım

```
String NS = "http://manufacturingsemanticlayer.org/ontologies/manufacturingDomain" +
"#";
OntModel base = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM );
base.read( partName, "RDF/XML" );

OntClass Activities = base.getOntClass( NS + "Activities" );
boolean direct = false;
Iterator it = Activities.listInstances(direct);

//check if the Individual is readyToDo
while(it.hasNext()){
    Individual oi = (Individual)it.next();
    Property readyToDo = base.getProperty( NS+"readyToDo" );
    try {
        if(oi.getPropertyValue(readyToDo).asLiteral().getBoolean()==true) {
            AllReadyToDoList.add(oi.getLocalName().toString());
        }
    }catch (Exception s) {}
}
```

Ek Açıklama-A.2 Aktivitelerin Özelliklerini görüntülemek için yazılan koddan bir kısım

```

public static void getActivityProperties(String partName, String ActName) {
    String NS = "http://manufacturingsemanticlayer.org/ontologies/manufacturingDomain" + "#";
    OntModel base = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
    base.read(partName, "RDF/XML");
    OntClass Activities = base.getOntClass(NS + "PartActivity");
    Individual oi = (Individual) base.getIndividual(NS + ActName);
    System.out.println(oi.getLocalName());
    String ActnameStr = oi.getLocalName();
    // print list if the Individual hasPart property
    Property Part = base.getOntProperty(NS + "hasPart");
    hasPartList.clear();
    try {
        for (StmtIterator j = oi.listProperties(Part); j.hasNext(); ) {
            Statement s = j.next();
            hasPartList.add(s.getObject().asResource().getLocalName());
            System.out.print("hasPart=>" + s.getObject().asResource().getLocalName());
        }
    } catch (Exception s) {System.out.println("N/A");}
    // print list if the Individual isPartOf property
    Property isPartOf = base.getOntProperty(NS + "isPartOf");
    isPartOfList.clear();
    try {
        for (StmtIterator j = oi.listProperties(isPartOf); j.hasNext(); ) {
            Statement s = j.next();
            isPartOfList.add(s.getObject().asResource().getLocalName());
            System.out.print("isPartOf=>" + s.getObject().asResource().getLocalName());
        }
    } catch (Exception s) {System.out.println("N/A");}
    // print property if the Individual has a Start
    Property Start = base.getProperty(NS + "hasStart");
    hasStartStr = "";
    try {
        System.out.println(oi.getProperty(Start).getString());
        hasStartStr = oi.getProperty(Start).getString();
    } catch (Exception s) {
        hasStartStr = "N/A";
        System.out.println("N/A");
    }
    // print property if the Individual has a End
    Property End = base.getProperty(NS + "hasEnd");
    hasEndStr = "";
    try {
        System.out.println(oi.getProperty(End).getString());
        hasEndStr = oi.getProperty(End).getString();
    } catch (Exception e) {
        hasEndStr = "N/A";
        System.out.println("N/A");
    }
    // print property if the Individual has duration
    Property Duration = base.getProperty(NS + "hasDuration");
    hasDurationStr = "";
    try {
        System.out.println(oi.getProperty(Duration).getString());
        hasDurationStr = oi.getProperty(Duration).getString();
    } catch (Exception d) {
        hasDurationStr = "N/A";
        System.out.println("N/A");
    }
    // print property if the Individual has canStartAfter property
    Property canStartAfter = base.getOntProperty(NS + "canStartAfterCompletionOf");
    canStartAfterCompletionOfList.clear();
    try {
        for (StmtIterator j = oi.listProperties(canStartAfter); j.hasNext(); ) {
            Statement s = j.next();
            canStartAfterCompletionOfList.add(s.getObject().asResource().getLocalName());
            System.out.print("canStartAfterCompletionOf " +
s.getObject().asResource().getLocalName() + "\n");
        }
    } catch (Exception s) {System.out.println("N/A");}
    // check if the Individual is readyToDo
    Property readyToDo = base.getProperty(NS + "readyToDo");
    readyToDoStr = "";
    try {
        System.out.println(oi.getPropertyValue(readyToDo).asLiteral().getBoolean());
        if (oi.getPropertyValue(readyToDo).asLiteral().getBoolean()) {
            readyToDoStr = "YES";
        } else { readyToDoStr = "NO";}
    } catch (Exception s) { System.out.println("N/A");}
    // check if the Individual is closed
    Property completed = base.getProperty(NS + "completed");
    completedStr = "";
    try {
        System.out.println(oi.getPropertyValue(completed).asLiteral().getBoolean());
        if (oi.getPropertyValue(completed).asLiteral().getBoolean()) {
            completedStr = "YES";
        } else { completedStr = "NO";}
    } catch (Exception s) { System.out.println("N/A");}
}

```

Ek Açıklama-A.3 İmalat olgunluk seviyesi sorularının özelliklerini görüntüleyen koddan bir kısım

```

public static Object[][] getMRLIndividualsProperties(String partName, String MRLStageName) {

    String NS = "http://manufacturingsemanticlayer.org/ontologies/manufacturingDomain" + "#";
    OntModel base = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
    base.read(partName, "RDF/XML");
    OntClass manufacturingFacility = base.getOntClass(NS + "MRL");

    OntClass c = base.getOntClass(NS + MRLStageName);
    System.out.println(c.getLocalName());
    String MRLStageStr = c.getLocalName();

    // find number of individuals
    counter = 0;
    ExtendedIterator ite = c.listInstances();
    while (ite.hasNext()) {
        System.out.println(ite.next());
        counter++;
    }
    System.out.println(counter);
    Object[][] MRLInstanceProperties = new Object[counter][5];

    // add values to MRLInstanceProperties array
    ExtendedIterator it = c.listInstances();
    int i = 0;
    while (it.hasNext()) {
        Individual oi = (Individual) it.next();
        System.out.println(oi.getNamespace());
        String MRLIndividual = oi.getURI();
        String MRLIndividualName = MRLIndividual.substring(MRLIndividual.lastIndexOf("#") + 1);
        System.out.println(MRLInstanceProperties.length);
        MRLInstanceProperties[i][0] = MRLIndividualName;

        // check if the Individual is related
        Property relevant = base.getProperty(NS + "relevant");
        String MRLRelevantStr = "";
        try {
            System.out.println(oi.getPropertyValue(relevant).asLiteral().getBoolean());
            if (oi.getPropertyValue(relevant).asLiteral().getBoolean()) {
                MRLRelevantStr = "Relevant";
            } else {
                MRLRelevantStr = "Irrelevant";
            }
        } catch (Exception s) {System.out.println("N/A");}
        MRLInstanceProperties[i][1] = MRLRelevantStr;

        // check if the Individual is closed
        Property closed = base.getProperty(NS + "completed");
        String MRLClosedStr = "";
        try {
            System.out.println(oi.getPropertyValue(closed).asLiteral().getBoolean());
            if (oi.getPropertyValue(closed).asLiteral().getBoolean()) {
                MRLClosedStr = "Closed";
            } else {
                MRLClosedStr = "Open";
            }
        } catch (Exception s) {System.out.println("N/A");}
        MRLInstanceProperties[i][2] = MRLClosedStr;

        // print if the Individual has comment
        Property comment = base.getProperty("http://www.w3.org/2000/01/rdf-schema#comment");
        List<String> commentList = new ArrayList<String>();
        try {
            StmtIterator iter = oi.listProperties(comment);
            while (iter.hasNext()) {
                commentList.add(iter.nextStatement().getObject().toString());
            }
        } catch (Exception s) {System.out.println("N/A");}
        MRLInstanceProperties[i][3] = commentList;

        // list checkForMRL properties
        Property checkForMRL = base.getProperty(NS + "checkForMRL");
        List<String> checkForMRLList = new ArrayList<String>();
        StmtIterator iter = base.listStatements(oi, checkForMRL, (RDFNode) null);
        try {
            while (iter.hasNext()) {
                checkForMRLList.add(iter.next().getObject().asResource().getLocalName());
            }
        } catch (Exception s) {System.out.println("N/A");}
        MRLInstanceProperties[i][4] = checkForMRLList;
        i++;
    }
    for(int row = 0; row < MRLInstanceProperties.length; row++){
        for(int element = 0; element < MRLInstanceProperties[row].length; element++){
            System.out.printf("Row: %d Element: %d Value: %s\n", row, element,
                MRLInstanceProperties[row][element]);
        }
    }
    return MRLInstanceProperties;
}

```

Ek Açıklama-A.4 Kavramların Özelliklerini görüntülemek için metot

```

public static void getObjectProperties(String partName, String ObjectName) {
    String NS = "http://manufacturingsemanticlayer.org/ontologies/manufacturingDomain" + "#";
    OntModel base = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
    base.read(partName, "RDF/XML");
    OntClass c = base.getOntClass(NS + ObjectName);
    System.out.println(c.getLocalName());

    // listsuperclasses
    ExtendedIterator<OntClass> itersuper = c.listSuperClasses();
    try {
        SuperClassList.clear();
        while (itersuper.hasNext()) {
            SuperClassList.add(itersuper.next().getLocalName());
        }
    } catch (Exception s) {
    }

    // listsubclasses
    ExtendedIterator<OntClass> itersub = c.listSubClasses();
    try {
        SubClassList.clear();
        while (itersub.hasNext()) {
            SubClassList.add(itersub.next().getLocalName());
        }
    } catch (Exception s) {
    }

    // print if the Individual has comment
    Property comment = base.getProperty("http://www.w3.org/2000/01/rdf-schema#comment");
    try {
        StmtIterator iter = c.listProperties(comment);
        CommentList.clear();
        while (iter.hasNext()) {
            CommentList.add(iter.nextStatement().getObject().toString());
        }
    } catch (Exception s) {
        System.out.println("N/A");
    }

    // listIndividuals
    ExtendedIterator<Individual> inditer = base.listIndividuals(c);
    IndividualList.clear();
    while (inditer.hasNext()) {
        IndividualList.add(inditer.next().getLocalName().toString());
    }

    // print out property name and its values
    PropertyAndValueList.clear();
    for (StmtIterator ipp = c.listProperties(); ipp.hasNext();) {
        Property p = (Property) ipp.next().getPredicate();
        System.out.println("Property=>" + p.getLocalName());

        for (Iterator ivv = c.listPropertyValues(p); ivv.hasNext();) {
            String valuenam = ivv.next().toString();
            System.out.println(c.getLocalName() + " " + p.getLocalName() + " " + valuenam);
            PropertyAndValueList.add(c.getLocalName() + " " + p.getLocalName() + " " + valuenam);
        }
    }

    // print annotation Properties
    ExtendedIterator<AnnotationProperty> annoiter = base.listAnnotationProperties();
    while (annoiter.hasNext()) {
        AnnotationProperty anno = (AnnotationProperty) annoiter.next();
        System.out.println("Property=>" + anno.getLocalName());
    }
}

```

Ek Açıklama-A.5 Aktivite eklemek için yazılan kodların ana kısmı

```
//add Required-activity
OntProperty canStartAfter = base.getOntProperty( NS+"canStartAfterCompletionOf" );
Individual newInd = base.getIndividual(requiredActivityToAdd);
oi.addProperty(canStartAfter, newInd );

//to see what happens
//print property if the Individual has a canStartAfter property
try {
    for (StmtIterator j = oi.listProperties(canStartAfter); j.hasNext(); ) {
        Statement s = j.next();
        System.out.print("canStartAfterCompletionOf=>" + s.getObject().asResource(
            ).getLocalName() + "\n" );
    }
} catch (Exception s) {
    System.out.println("N/A");}

//save the file to the target folder
String fileName = "sample.rdf";
FileWriter out = new FileWriter( fileName );
try {
    base.write( out, "RDF/XML-ABBREV" );
}
```