

Model Gdml Taktiksel Harp Sahası Ynetim Sistemi (THSYS) Tasarımı

Ethem Fatih Can¹, zgr Aydın Tekin¹ ve Mahmut Devrim Tokcan²

¹Bilkent niversitesi Bilgisayar Mh. Blm, Ankara

²Orta Doęu Teknik niversitesi Enformatik Enstits, Ankara

{efcan, oatekin}@cs.bilkent.edu.tr, e159569@metu.edu.tr

ZETE

Askeri bir hareketin başarısında, emirlerin alt birimlere, raporların da st birimlere doęru olarak iletilmesi ve her birimin kendine dşen grevi emir komuta baęlamında yerine getirmesi esastır. Mttefik hareketlarda ise saha bilgisinin mttefik birimler arasında doęru ve hızlı bir şekilde paylaşılabilmesi hayati neme sahiptir. Endstriyel veya araştırma tabanlı olsun, THSYS zerine yapılan alıřmalar standart bir dil geliřtirme zerine yoęunlařmaktadır ki bu ortak dil sayesinde atıřmada yer alan btn birimler atıřmanın iřleyiři hakkında gncel bilgiye sorunsuz eriřebilsin. Bu alıřmada THSYS iin  farklı yaklařım kullanılmıřtır. İlk olarak dil bilgisi geliřtirme yaklařımı, ikinci ve nc olarak ise sırasıyla st-st modelden st model geliřtirme yoluyla ve ayırlama yoluyla model gdml tasarım yaklařımları geliřtirilmiř ve karřılařtırılmıřtır.

Abstract— In a military operation distribution of commands, feedback of controls, splitting the intelligence, and execution of missions are the crucial issues. Besides, in a joint war-fight, exchange of information is the most important issue. These issues are handled by battle management systems. Either it is industrial application or research project, the basic aim of such systems is to provide a common language so that any instrument of the combat will be well-informed. In this work, we elaborate on a tactical battlefield management system (TBMS) utilizing the benefits of the model-driven approach. Our system provides high level abstractions of the instruments of a war-fight. We first illustrate a grammar based domain specific language (DSL) for TBMS, then we provide two metamodels designed in two different ways; from scratch and extension of an existing system (UML Profiling). Finally, we discuss the technical frameworks available to develop such metamodels and the differences of the three approaches: grammar based DSL, metamodel from scratch, and metamodel from extending an existing model.

1 GİRİŐ

Bir řavař sahasında birok birim yer alır. Her birim, genel bir ama çerevesinde kendisine atanmıř zel bir grevi yerine getirmeye alıřır. Bu birimler farklı saflardan olabileceęi gibi aynı safta da olabilir. Burada, karıřıklıęa neden olmaması iin btn birimlerin aynı safta oldukları ve tek bir genel ama iin savař sahasında oldukları durum gz nne alınmıřtır.

Savař sahası iin planlanan taktiklerin başarıya ulařabilmesi ancak birimler arası etkin haberleřme ile mmkndr. Her birim kendi zel grevini yerine getirirken, aynı zamanda, dięer birimleri de evresel ve taktiksel deęiřiklikler hakkında bilgilendirebilmelidir. Bilindięi gibi Kıbrıs barıř hareketi sırasında haberleřme sistemlerinin kmesi sonucu birliklerin kendi safındaki birlikleri karřı saftan grp tahrip etmiřtir.

Sahadaki birimlerin sayısı arttıka emir komuta yapısı daha da karmařıklařır, birimlere zel grevlerin atanması ve komutanlar tarafından birimlerin ynetilmesi daha da zorlařır. Bu nedenle komutanlar iin sahanın genel grnmn ve zel grevlerin takibini saęlayacak detaylı grnmler sunacak sistemler gereklidir. THSYS bu amala geliřtirilen, harp sahası iin taktik geliřtirme ve gerek-zamanlı uygulama bilinci saęlayacak bir sistemdir.

Bahsedildięi zere, mesajlařma (haberleřme)

THSYS'nin en önemli kısmını oluşturmaktadır. Varolan teknolojik altyapıyı kullanarak, sayısal mesajlar, güvenli ve etkin bir şekilde birimler arası iletişimi sağlar. Böylece, askeri taktikler doğru bir şekilde sahaya hızla taşınabilir.

THSYS sistemlerinin genel özellikleri şunlardır:

- ön-tanımlı mesaj kalıpları,
- ön-tanımlı yapıda olmayan serbest mesajlaşma desteği,
- mesaj geçerliği için başlangıç ve bitiş tarih-zaman dilimleri belirleme imkanı,
- hava destekli fotoğraf gibi mesaj ekleri,
- saha ve olay görüntüleme amaçlı coğrafik bilgi sistemi,
- mesajlar için sayısal kodlama,
- gelen ve giden mesajların kaydedilmesi.

THSYS için model güdümlü yaklaşım hedefimizin iki ana nedeni vardır. Birincisi, THSYS dünyada yoğun olarak ilgilenilmektedir, birçok farklı örnekleri vardır; fakat henüz istenilen seviyede amaçları karşılar hale gelememiştir. THSYS için geliştirilmiş bilinen bir üst-model yaklaşımı yoktur. İkincisi, taktiksel sahanın coğrafi konumuna bağlı olarak, sahada yer alacak birden fazla müttefik saf için ortak bir dil ya da model bilinmemektedir. Safaların kullandıkları farklı sistemler doğru haberleşmeyi sağlayacak uyum paketlerinden yoksundur. Farklı sistem modellerinin birbirine uyumunu sağlayacak model dönüşüm araçları üst-model yaklaşımı içerisinde kullanılabilir.

Yazının ilerleyen kısımları şu şekilde düzenlenmiştir. THSYS üzerine yapılan gerek endüstriyel gerekse eğitim amaçlı çalışmalar bir sonraki bölümde, Bölüm 2, tartışılmıştır. Bölüm 3'te ilgilenilen alan incelenmiş, Bölüm 4'te THSYS için dil bilgisi geliştirilmiştir. Daha sonra, Bölüm 5'te THSYS için sıfırdan ve ayırlama yoluyla geliştirilen üst-modeller sunulmuştur. Bölüm 6'da geliştirilen yaklaşımlar karşılaştırılmıştır. Son olarak da Bölüm 7'de ileriye yönelik yapılması planlanan geliştirmeler sunulmuştur.

2 BENZER ÇALIŞMALAR

THSYS üzerine olan ilgi, bu konuda çalışan şirketler göz önüne alındığında, oldukça yükündür. Çalışmaların odak noktası alt görevlerin

otomatik ve daha hızlı yerine getirilmesidir. Haberleşme elemanlarının entegrasyonu ile emir komuta koordinasyonunun sağlanması hedeflenmektedir.

Thales şirketinin Tactical Battlefield Management System (T-BMS) isimli uygulaması emir komuta mesajlarının görsel dağılımı ve güvenli veri haberleşmesi üzerine kurulmuştur [1].

TROP taksiksel seviyede saha yönetim sistemine bir diğer örnektir. Temel işlevleri, görev hazırlığı, emir, komuta, haberleşme ve veri paylaşımıdır. Uygulamanın hali hazırda bulunan donanım ve yazılım paketleri ile uyumlu olduğu ileri sürülmektedir [2].

SitaWare saha yönetim sistemi, saha komutanlarının ve birlik komutanlarının sahadaki olayları gözlemlemelerine yardımcı olmak amacıyla piyasaya sürülmüştür. Diğer bir amacı da sahadaki alt birimlerin koordinasyonu ve sahanın geneli hakkında bilgilendirilmesidir [3].

Çatışma yönetim sistemi, taktiksel komuta ve kontrollerin en üst birimden en alt birimlere kadar çift yönlü olarak iletilmesini sağlar [4].

BattleHawk çatışma yönetim sistemi, "Vehicle Platform" ve "Infantry Digital Soldier System" gibi yazılımlarla birlikte, haberleşme, seyrüsefer ve güvenlik bakımından harp sahalarının kolayca yönetilmesini amaçlar [5].

Endüstriyel uygulamalara ek olarak, çeşitli bağımsız araştırma merkezlerinde de benzer sistemler geliştirilmiştir. Araştırmaların odağı ortak çatışmalar için sefer planlama, yürütme, koordinasyon yönlendirmedir. Örneğin, Amerika Birleşik Devletleri'ndeki "Tactical Battle Management Core Systems (TBMCS)" sahadaki savaş uçaklarının, hava yakıt ikmal tankerlerinin, helikopterlerin, insansız hava araçlarının ve güdümlü füzelerin eş-güdümlü hareketini sağlamak amacıyla geliştirilmiştir. TBMCS başlangıçtaki geniş amaçlı sunucustemci uygulaması halinden, daha gelişmiş internet tabanlı ticari bir şirket haline gelmiştir [6].

Araştırmacılar herkesçe anlaşılabilir bir Savaş Yönetim Dili "Battle Management Language (BML)" geliştirmiştir. BML askeri harekattaki iletişim dilinin standartlaştırılması yönündeki ilk adımlardandır. Bu dille taktiksel sahadaki aktif birimlerin emir komutası

sağlanır [7]. "Coalition Battle Management Language (C-BML)" dili "Simulation Interoperability Standards Organization" tarafından geliştirilmiştir ki bu dil hem canlı birimlerin hem de insansız birimlerin benzetim (simulasyon) ve gerçek-zamanlı işleyişi için ortak bir dildir [8].

[9] bir grup askerden oluşan bir askeri birim için model önermiştir. Bu modelde dağıtık süreçlerin eş-zamanlı olarak sürdürülebilirliği esas alınmıştır.

[10]'da model güdümlü bir taktiksel hava sahası yönetim sistemi "JSSEO" geliştirilmesi incelenmiştir. JSSEO, OMG'nin "Model Driven Architecture (MDA)" yöntemini kullanarak platformdan bağımsız bir model sunar. JSSEO'nun dayandığı ortak süreç mimarisi müttefiklerin kullandığı farklı sistemlerin entegrasyonu problemini daha kontrol edilebilir bir hale getirir.

Özetle, endüstriyel saha yönetim sistemleri şirketlerin kendi ihtiyaçlarını esas alarak mevcut yazılımların ve dillerin taktiksel sahada kullanılmak üzere birleştirilerek ortak bir dil oluşturmak çabasına dayalıdır. Fakat bütün saha yönetim modellerini veya dillerini kapsayan, hepsinin temel aldığı bir üst-model hala üretilmemiştir. A ve B ülkelerinin kullandığı farklı iki taktiksel yönetim sisteminin ortak bir harekatta tümleşik şekilde kullanılması problemi şirketler bazında tam olarak çözülmüş değildir. Taktiksel saha yönetimi üzerine çalışan bağımsız araştırmacılar ise ortak dil geliştirilmesi üzerine çalışmış ve uzun yıllar dil bilgisi tabanlı yöntemler denenmiştir. Böylece dil bilgisine dayalı birçok araç geliştirilmiş, özel amaçlı dillerin yazılabildiği ve kullanıcı tarafından tanımlanan kurallara göre test edilebilen editörler tasarlanmıştır. Bölüm 4'te, bu araçlardan açık kaynak kodlu Eclipse ortamında çalışabilen XText yazılımı kullanılarak THSYS için geliştirilen dil bilgisi verilmiştir. İlerleyen bölümlerde ise dil bilgisine değil üst-modele dayalı yöntemler verilmiş, dil bilgisi yöntemine göre üstünlükleri tartışılmıştır.

3 ÇALIŞMA ALANI ANALİZİ

Askeri birimler hiyerarşik yapıdadır. Bu yapıda emirler üst birimlerden alt birimlere, raporlar

da alt birimlerden üst birimlere iletilir. Bu çift yönlü haberleşme yapısı sayesinde emir komuta zincirindeki her birim hareket hakkında yalnızca ana görevin kendisini ilgilendiren kısmını bilir ve yürütür. Bu şekilde bir haberleşme geleneksel radyo, telsiz ağı ile sağlanamaz. Bölüm 1'de de sıralanan temel ihtiyaçları karşılayacak iyi tanımlanmış bir hareket yönetim sistemine ihtiyaç vardır. Bu ihtiyaçlar özetle şunlardır. Emirlerin daha hızlı iletilmesi ve yürütülmesi, böylece ana görevin çabucak tamamlanması, tüm birimlerinin anlayacağı ortak bir dil tanımlanması, görev alanlarının belirlenip görevlerin sorumlulara iletilmesi, görev tatbik durumunun (rapor) akıllıca ve otomatik olarak güncellenmesi, alarm durumlarında ilgili birimlerin derhal haberdar edilmesi ve emir komutaların müttefik çatışma durumları için entegrasyonu.

Bizim harp sahası modelimizde, sahada ana unsur olarak bir çatışma (combat) belirli bir zamanda olur. Bir ana görev (mission) ve bu ana görevin özel parçaları (task) vardır. Çatışmada sıfır ya da daha fazla sayıda birlik yer alır. Her birliğe bir özel görev atanır. Her birliğin bir komutanı (commander) ve sıfır ya da daha fazla sayıda askeri (trooper) vardır. Bir birliğe teçhizat (equipment) olarak araç (vehicle), silah (weapon) ve mühimmat (ammunition) tahsis edilebilir. Birliğin kullandığı bir iletişim sistemi (message management system) vardır ve birlikler arası haberleşme bu sistem ile sağlanır. İleti (message) bir emir (order) veya rapor (report) olabilir. İletiler çeşitli yapılarda olabilir. Çalışma alanı elemanlarına dair terimler sözlüğü Tablo 1'de sunulmuştur.

4 DİLBİLGİSİ TANIMLAMA

Bu bölümde, çalışma alanı olarak seçtiğimiz THSYS için tanımladığımız dil bilgisi sunulmuştur. Amaç, bütün birimlerce anlaşılacak ortak bir dil geliştirmektir ki emirler hızlı bir şekilde yürütülebilir ve saha içi bilgi akışı hızlı, güvenli ve doğru bir şekilde sağlansın. Önerilen dil bilgisi BNF (Backus-Naur-Form) üst dil yapısında tasarlanmıştır. Geliştirilen dilin kısa bir bölümü aşağıda örnek olarak sunulmuş, tamamı da Ek A'da verilmiştir.

```
<Combat> ::= 'Combat' <Name> <operationDate>
           <Mission> <Troops>
```

TABLO 1
Çalışma alanı terimler sözlüğü

Terim	Açıklaması
Çatışma (combat)	Bir grup birliği içeren askeri çatışma.
Birlik (troop)	Bir grup asker ya da askeri birim.
Ana görev (mission)	Çatışmadaki birliklerinin genel görevi.
Özel görev (task)	Ana görevin özel bir parçası.
Teçhizat (equipment)	Birliğe tahsis edilen kaynak.
Personel (personnel)	Bir komutan veya bir grup asker.
İletişim sistemi (Message Management System)	Bir birliğe ait giden, gelen iletilerin yapısını belirleyen, yönlendiren ve kaydeden sistem.
Konum (Location)	Dünya üzerinde bir noktayı belirten enlem, boylam çifti.
İleti (Message)	Özel yapıda emir, komuta mesajları.
Emir (Order)	Alt birimlere görev bildiren ileti yapısı.
Rapor (Report)	Alt birimlerin üst birimlere gönderdiği ileti yapısı.

```
<Troops> ::= <Troop> | <Troops>
<Troop> ::= 'Troop' <Name> <Location>
    <Task> 'Commander' <Personnel>
    'Trooper' <Personnels> 'Equipment'
    <Equipments> <MsgManSys> | <Troops>
```

Önerilen dil bilgisi BNF olarak tanımlandıktan sonra Eclipse XText eklentisi kullanılarak dil kuralları XText ile yazılmış, çeşitli ek kısıtlamalar eklenmiş ve otomatik olarak dil editörü üretilmiştir. Bu editör ile kullanıcı kendi uygulamasını rahatlıkla yazabilir ve dil bilgisine uymayan imla hataları anında kullanıcıya bildirilir. Kullanılan XText kodunun bir kısmı aşağıda örnek olarak sunulmuş, tamamı ise Ek B'de verilmiştir.

```
Combat :
    "Combat" name=ID
    operationDate = DateTime
    mission = Mission
    (troops += Troop)* ;
Troop :
    "Troop" name=ID
    location = Location
    tasks += Task+
    "Trooper" trooper+=Personnel+
    "Commander" commander=Personnel
    ("Equipment" equipments +=
        (Vehicle | Weapon | Ammunition))*
    msgManSys = MsgManSys
```

```
(subTroops += Troop)*;
```

XText ile ek kısıtlar iki farklı dilde tanımlanabilir. Birincisi "Object Constraint Language (OCL)" türevi bir dil olan "Check Language", diğeri ise Java. Bizim tercihimiz Java olmuştur. Aşağıda kısıtı gerçekleyecek olan Java kodunun Eclipse XText ortamında nasıl etkinleştirileceğini açıklayan örnek sunulmuştur.

```
// java-based API for validation
fragment=validation.JavaValidatorFragment
...
composedCheck = "MyDslJavaValidator"
}
```

Dil editöründe kullanıcıyı hatalı girdilere karşı kısıtlayacak olan Java kodu, "MyDslJavaValidator.java" dosyasına yazılmış ve kısıt koşullarının nasıl girildiğine dair örnek aşağıda sunulmuştur. Tamamı da Ek C'de verilmiştir.

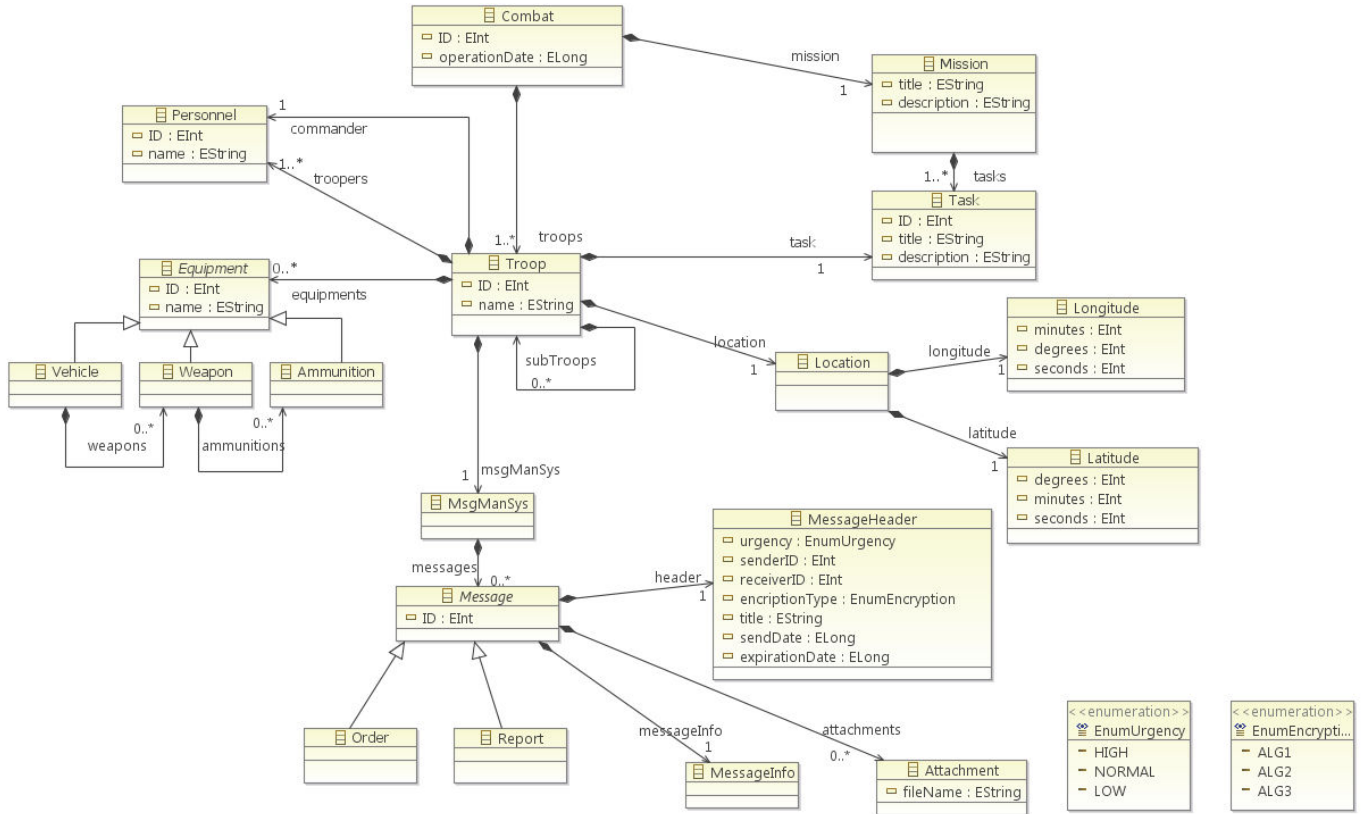
```
public void checkLatitude(
    Latitude latitude){
    if( latitude.getDegrees() <0
    || latitude.getDegrees() >90){
    ...
    }
```

5 THSYS İÇİN ÜST MODEL TANIMLAMA

Bu bölümde, THSYS için Bölüm 3'te belirtilen ihtiyaçları karşılamak üzere yeni üst modeller tasarlanmıştır. İki farklı yöntem kullanılmıştır. İlk olarak sıfırdan EMOF üst-üst modeline dayanan bir üst model, ikinci olarak da var olan UML üst modelinin ayrıştırmaya yöntemi (profiling) tasarlanması yöntemleri ele alınmıştır. Her iki üst model için soyut sözdizim modeli, somut sözdizim modeli ve anlamsal kurallar tanımlanmıştır. Üst modellere uyumlu örnek modeller sunulmuştur.

5.1 SIFIRDAN ÜST MODEL TANIMLAMA

THSYS için sıfırdan bir üst model EMF (Eclipse Modeling Framework) kullanılarak tanımlanmıştır. Bu üst model Eclipse için geliştirilen Java tabanlı bir MOF olan Ecore üst üst modeline dayanır ve Ecore model olarak geçer [11]. Bir sonraki bölümde bu üst modelin temelini oluşturan soyut sözdizimi modeli verilmiştir.



Şekil 1. Sıfırdan oluşturulan üst model.

5.1.1 Abstract Syntax

Ecore kullanılarak geliştirilen soyut sözdizim modeli Şekil 1’de verilmiştir. Modelin tasarımında kullanılan gösterim UML’e benzer ve kendini açıklayıcı şekildedir. Okuyucunun temel UML gösterimine hakim olduğu düşünülerek modelde kullanılan gösterimle ilgili daha fazla açıklama yapılmaya ihtiyaç görülmemiştir.

Tablo 2’de üst model elemanları ve ilintili oldukları MOF elemanları verilmiştir.

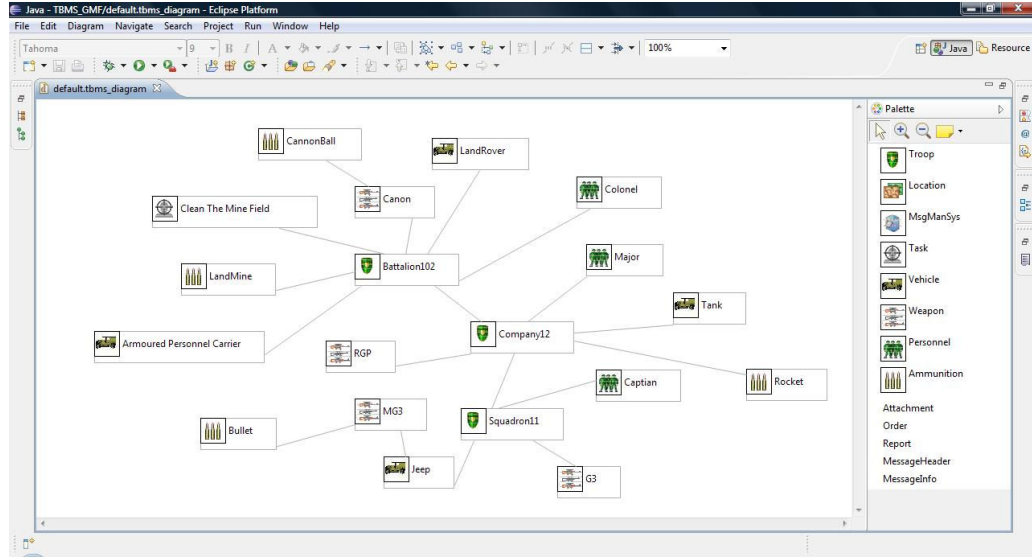
5.1.2 Somut sözdizim modeli

Önceki bölümde oluşturulan Ecore modelinin kullanıcıya sunumunda kullanılacak olan somut gösterim modelini oluşturmak için Eclipse’in GMF (Graphical Modeling Framework) aracı kullanılmıştır. Bu aracın kullanılmadığı durumlar için EMF hazır bir ağaç yapısı gösterim modeli sunmaktadır. Öcelikler bizim GMF ile hazırladığımız somut sözdizimi bir örnek model ile Şekil 2’te gösterilmiştir. Hazırladığımız somut sözdizimi gerçek hayatta

TABLO 2
Üst modelin MOF ile bağlantısı

TBMS Element	MOF Element
TBMS::Combat	MOF::Class
TBMS::Troop	MOF::Class
TBMS::Personnel	MOF::Class
TBMS::MsgManSys	MOF::Class
TBMS::Attachment	MOF::Class
TBMS::MessageInfo	MOF::Class
TBMS::MessageHeader	MOF::Class
TBMS::Equipment	MOF::Interface
TBMS::Message	MOF::Interface
TBMS::Location	MOF::Interface
TBMS::EnumUrgency	MOF::DataType
TBMS::EnumEncryptionType	MOF::DataType

kullanılan simgelerden oluşmaktadır. Örneğin personel taşıma aracı olan “Personal Carrier” askeri bir araç simgesi ile gösterilmiştir. Böylelikle kullanılan somut gösterim kendi kendini açıklar niteliktedir. Örnek modele ilişkin ağaç gösterimi Bölüm 5.1.4’de verilmiştir.



Şekil 2. Example model using a custom concrete syntax.

5.1.3 Statik anlamsal kurallar

Şu ana kadar THSYS’de kullanılan elemanların ve bu elemanların birbirleri ile olan ilişkilerinin yer aldığı soyut sözdizim modeli ve bu modelin kullanıcıya sunulacak olan somut sözdizim modeli tanımlanmış oldu. Soyut modelde elemanlar arası ilişkiler sayısal bağlamda kısıtlanmaktadır. Örneğin her birliğin bir adet özel görevi olmak zorundadır veya bir birliğin birçok aracı olabilir. Bu bölümde üst modelimiz için soyut modelde getirilemeyen kısıtlamaya yönelik bazı statik anlamsal kurallar tanımlanmıştır. Eclipse ortamı bu kuralların iki farklı şekilde tanımlanmasına imkan sağlar. Birincisi geleneksel yöntem olan OCL (Object Constraint Language) kullanımı, ikincisi de özel Java fonksiyonlarının soyut model ile birleştirilmesi yöntemidir. Bu çalışmada ikinci yöntem tercih edilmiştir.

Tablo 3’de üst modelimize eklenen statik kurallara verilmiştir. Modelin bu kurallara uyum testi Java fonksiyonları olarak gerçekleşmiş, örnek bir Java fonksiyonu Ek D’de verilmiştir. Kuralların üst modele bağlantısı Ecore’un sunduğu “EAnnotation” ile yapılmıştır. Böylece Java fonksiyonlarımız model yaratılma esnasında otomatik olarak model editör tarafından çağrılmaktadır.

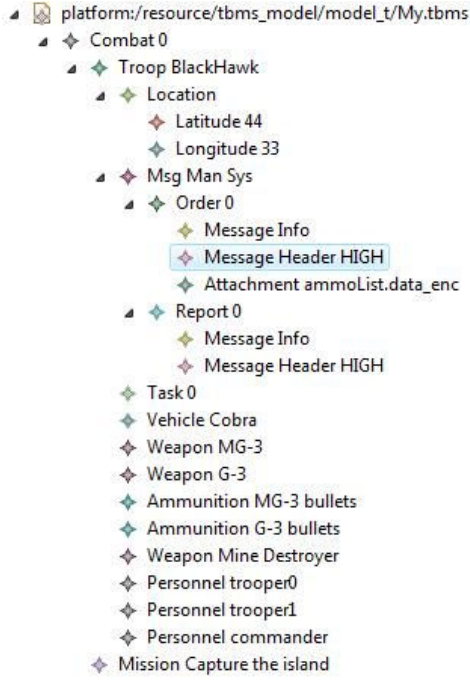
TABLE 3
Üst model kısıtları

THSYS elemanı	Kısıt
Troop → name	Birliğe bir isim atanması zorunludur.
Personnel → name	Personele bir isim atanması zorunludur.
MessageHeader → sendDate	İleti gönderme zamanı, iletinin geçerliğinin bitişinden önce olmalıdır.
Equipment → name	Techizata bir isim atanması zorunludur
Latitude→degree	Enlem derecesi [-90, 90] aralığında olabilir.
Latitude→minutes	Enlem dakikası [0, 60] aralığında olabilir.
Latitude→seconds	Enlem saniyesi [0, 60] aralığında olabilir.
Longitude→degree	Boylam derecesi [-180, 180] aralığında olabilir.

5.1.4 Örnek model

Sıfırdan yaratılan üst modele ilişkin örnek model ağaç yapısında ve XML düzeninde olmak üzere iki ayrı somut sözdizim modeli ile gösterilmiştir. Daha önce tanımladığımız özel somut modeli gösterimi için Bkz. Şekil 2. XML düzenindeki örnek modelin bir kısmı aşağıda, tamamı da Ek E’de verilmiştir. Şekil 3’de de ağaç yapısı gösterimi sunulmuştur.

```
<troops name="BlackHawk">
  <location>
```



Şekil 3. Sıfırdan tanımlanan üst model için model ağaç yapısı gösterimi.

```
<latitude degrees="44"
```

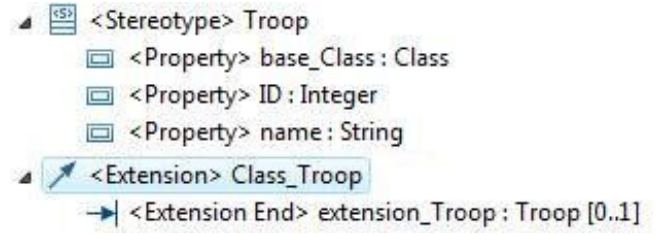
5.2 GENİŞLETME İLE ÜST MODEL TANIMLAMA

Hafif genişletme yöntemi var olan bir üst modele biçim eklenerek yapılır. Bu bölümde UML üst modeline amacımıza uygun biçimler tanımlayarak tasarladığımız THSYS üst modeli anlatılmıştır. Biçimlerin kendi değişkenleri, diğer deyişle etiketlenmiş değişkenleri olabilir. Bu çalışmada UML sınıf üst modeli Eclipse ortamında EMF 2.5 ve UML2.2 araçları kullanılarak biçimlendirilmiş ve amaca uygun olarak bazı etiketlenmiş değişkenler tanımlanmıştır. Biçimlendirme aşamasında alınan Eclipse görüntüsü Şekil 4’de verilmiştir.

5.2.1 Soyut sözdizim modeli

Çalışma alanımız olan THSYS için genişletme yöntemiyle tasarladığımız üst model Şekil 5’te gösterilmiştir. Daha önce sıfırdan üst model tanımlanması bölümünde anlatılan ihtiyaçlar aynen dikkate alınmıştır.

Tablo 4’te üst modelimizin elemanları ve karşı düşen UML üst model elemanları verilmiştir.



Şekil 4. during the Extension Operation

TABLO 4
Üst modelin MOF ile olan bağlantısı

Stereotype	Metaclass
Combat	UML::Class
Task	UML::Class
Mission	UML::Class
MessageInfo	UML::Class
Personnel	UML::Class
Message	UML::Class
Troop	UML::Class
Location	UML::Class
Equipment	UML::Class

Kullandığımız sınıf elemanlarının değişkenleri ise “UML:Property” üst model elemanına bağlıdır.

5.2.2 Somut sözdizim modeli

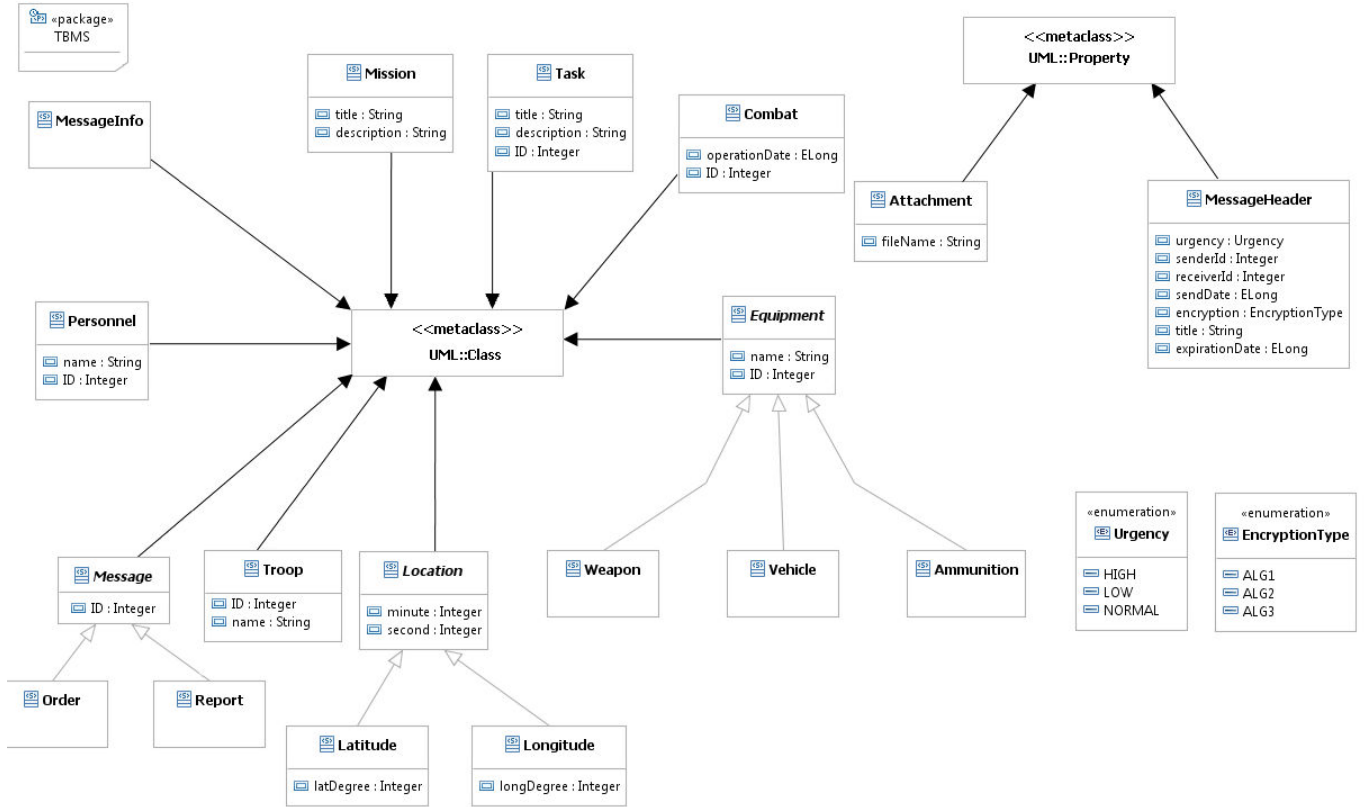
Ele alınan genişletme yöntemi UML üst modeline dayandığından burada kullanılan somut gösterimler UML somut gösterimiyle birebir örtüşür. UML somut gösterimi kullanıcılar tarafından iyi bilindiği düşünülerek yeni bir somut gösterim modeli tanımlanmamıştır.

5.2.3 Statik anlamsal kurallar

Genişletilmiş üst modelimize dair tanımlanan kurallar önceki bölümlerde anlatılan sıfırdan üst model tasarlanmasında kullanılan kurallar ile aynıdır.

5.2.4 Example Model

Şekil 6’de genişletilmiş üst modelimiz ile uyumlu olacak şekilde tasarlanmış örnek bir model sunulmuştur. Yukarıda da belirtildiği gibi bu modelin somut gösterimi UML gösterimi ile aynıdır.



Şekil 5. Var olan UML üst modelinden genişletme yoluyla elde edilen THSYS üst modeli.

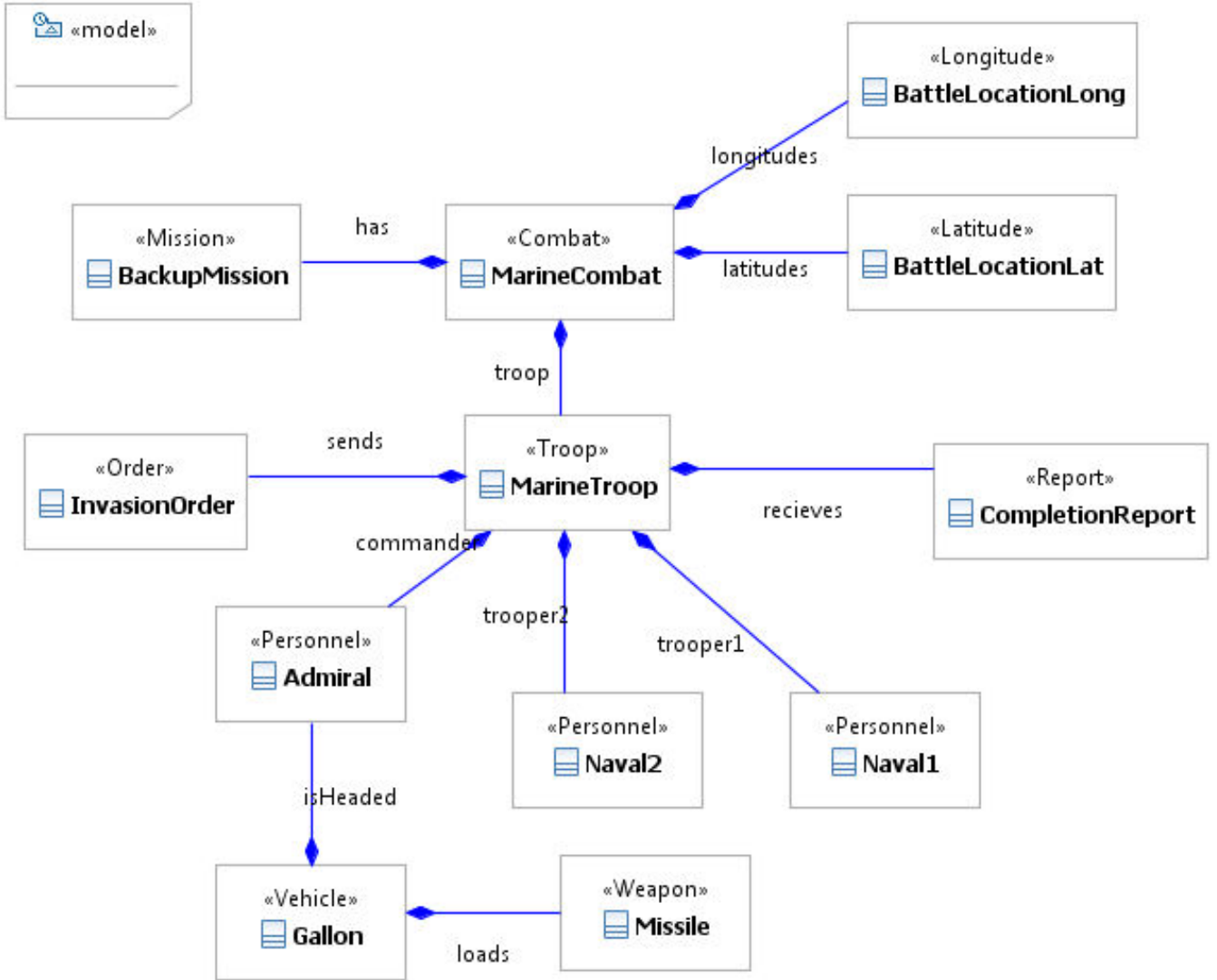
6 TARTIŞMA

THSYS sistemleri çalışma alanı olarak seçilmiş ve bu sistemler için ortak dil tanımlama amacıyla üç ayrı yöntem izlenmiştir. İlk önce BNF ve XText kullanılarak bir dil bilgisi geliştirilmiştir. BNF kendisi ile tanımlanan bir üst dildir. XText de aynı şekilde kendisi ile tanımlanır ve EBNF'in özel biçimidir. Böylece "?", "*" ve "+" gibi dil geliştirmeyi kolaylaştırıcı yetenekleri vardır. Ne BNF ne de EBNF için öntanımlı veri tipi yoktur, örneğin eşsiz tip anlamına gelen "ID" veri tipi vardır. Bu da soyut sözdizimi modeli için tanımlanması gereken statik anlamsal kuralların tanımlanması ek yükünü getirmektedir. Diğer yandan, XText ile hem öntanımlı veri tipleri kullanılabilir hem de statik anlamsal kısıtlar tanımlamak mümkündür. Ayrıca XText, geliştirilen dil için otomatik olarak bir editör oluşturabilir, böylece geliştirilen dili kullanıcı rahatlıkla kullanarak fakat aynı zamanda da kısıtlara uymak zorunda kalarak kendi dil modelini oluşturabilir.

İkinci yöntem olarak THSYS için sıfırdan bir üst model oluşturulmuştur. Bu yaklaşım için MOF'un özel hali olan Eclipse Ecore üst üst modeli kullanılmıştır. Eclipse EMF, Ecore üst modelinin UML gösterimi kullanan bir görsel arayüz ile tasarlanmasına imkan sağlar. Ecore'un "EAnnotations" elamanları da statik anlamsal kuralların rahatlıkla üst modele eklenmesini sağlar. 2010 yazında kullanıma sunulacak olan EMF eklentisi ile OCL biçiminde kısıt tanımlamak daha da kolaylaşacaktır. Şimdiki son sürüm EMF yalnızca Java fonksiyonları ile kısıt tanımlamaya imkan vermektedir ki bunun tek tek elle yazılması kolay değildir.

Üçüncü yöntem olarak da var olan bir UML üst modeli biçimler tanımlanarak genişletilmiştir. Her sınıf için basit bir biçim tanımlanmış fakat biçimler arasında ilişkiler tanımlanmamıştır. Bu yöntem için de sıfırdan üst model tanımlamada belirtilen kısıt ekleme zorlukları aynen geçerlidir.

Bir çalışma alanı için bahsedilen bu üç yöntemden hangisinin kullanılması gerektiği ile



Şekil 6. Genişletilmiş üst modele uyumlu örnek model.

ilgili olarak bizim çıkarımlarımız şunlardır. Çalışma alanına özel bir dil (DSL) tanımlanması için en zor yöntem dil bilgisi geliştirme yöntemidir çünkü tüm türetmeye açık ve kapalı (terminal/nonterminal) terimlerin ve özel veri tiplerinin tanımlanması ve dikkatle yazılması gerekmektedir. Bu durum dil bilgisi tanımlama yönteminin daha alt seviye programlama hatalarına açık bir konuma getirmektedir. Fakat dilbilgisi tanımlama yöntemi uzun yıllardır yazılımcılar tarafından kullanıldığından, bu yöntem için gerekli araçlara ulaşmak çok kolaydır. Son yöntem olarak incelenen UML üst modelinin biçimlendirilmesi

yöntemi gerçekleştirilmesi en kolay olan yöntemdir fakat bu yöntem ancak çok fazla biçim değişikliği gerektirmeyen hafif biçimlendirme durumları için uygundur. Eğer uygulama alanı olarak THSYS gibi büyük bir sistem seçilmişse ve ihtiyaçların karşılanabilmesi için yüksek seviye (ağır) soyutlaştırma gerektiriyorsa, bu durumda izlenmesi gereken yöntem ikinci yöntem olarak sunulan sıfırdan üst model tanımlama yöntemi olmalıdır.

7 SONUÇ

Bu çalışmada taktiksel harp sahası yönetim sistemi (THSYS) çalışma alanı olarak seçilmiş,

alandaki benzer çalışmalar gözden geçirilip bu alana özel üç farklı dil geliştirme yöntemi incelenmiştir. İlk olarak Eclipse XText yazılımı kullanılarak dil bilgisi geliştirme yöntemi uygulanmıştır. Daha sonra model güdümlü yaklaşımlardan, Eclipse Ecore kullanılarak sıfırdan üst model geliştirme yöntemi ve UML biçimlendirme kullanılarak var olan UML üst modelinin hafif biçimlendirilmesi yöntemi uygulanarak THSYS için üst modeller elde edilmiştir. Eclipse GMF aracı ile THSYS için özel olarak hazırlanan somut sözdizim modeli gerçek hayattan simgeler kullanılarak oluşturulmuş, üst modellere uyumlu M1 seviye örnek modeller sunulmuştur.

Son olarak, uygulanan üç yöntem karşılaştırılmıştır. Dilbilgisine dayalı yöntemin, alışılacağı üzere uygulamasının kolay olduğu ancak mevcut ihtiyaçları karşılamak, özellikle ortak dil oluşturmak için yeterli olmadığı sonucuna varılmıştır. Yeni geliştirilmekte olan açık kaynak yazılımlar ile desteklenen model güdümlü yöntemlerin zaman içinde daha başarılı sonuçlar vereceği düşünülmektedir. Çok yakında bu çalışmaya eklenecek olan model-model ve model-kod dönüşümleri ile model güdümlü tasarım yöntemlerin üstünlüğünün daha iyi anlaşılacağı düşünülmektedir.

TEŞEKKÜR

Dr. Bedir Tekinerdoğan'a değerli katkılarından dolayı teşekkür ederiz.

KAYNAKÇA

- [1] Thales. (2010) Tactical battlefield management systems: T-bms, comm@nder. [Online]. Available: <http://www.thalesgroup.com>
- [2] M. Suppliers and News. (2010) Tactical battlefield management systems - trop. [Online]. Available: <http://www.armedforces-int.com/article/tactical-battlefield-management-systems-trop.html>
- [3] Systematic. (2010) Sitaware battle management. [Online]. Available: <http://www.systematic.com>
- [4] B. Systems. (2010) Battle management system. [Online]. Available: <http://www.baesystems.com>
- [5] R. Globalnet. (2010) Battlehawk combat management system. [Online]. Available: <http://www.rfglobalnet.com/product.mvc/Combat-Management-Systems-CMS-0001?VNETCOOKIE=NO>
- [6] L. Martin. (2010) Tactical battle management core systems (tbmcs). [Online]. Available: http://www.deagel.com/Display-and-Network-Systems/TBMCS_a001286001.aspx
- [7] W. P. Sudnikovich, J. M. Pullen, M. S. Kleiner, and S. A. Carey, "Extensible battle management language as a transformation enabler," *Simulation*, vol. 80, no. 12, pp. 669-680, 2004.
- [8] S. I. S. O. (SISO). (2010) C-bml product development group. [Online]. Available: <http://www.sisostds.org>
- [9] A. Bordertsky, R. B. L. Duffy, E. Bach, and C. Oros, "A proposed model of battle rhythm at the tactical level," in *9th International Command and Control Research and Technology Symposium*, San Diego, CA, 2004.
- [10] R. W. Jacobs, "Model-driven development of command and control capabilities for joint and coalition warfare," in *9th International Command and Control Research and Technology Symposium*, San Diego, CA, 2004.
- [11] E. documentation Current Release. (2010) The eclipse modeling framework (emf) overview. [Online]. Available: <http://help.eclipse.org>

Ek A FULL VERSION OF THE GRAMMAR IN BNF

```

<Combat> ::= 'Combat' <Name> <operationDate> <Mission> <Troops>
<Name> ::= <Letter> | <Digit> | <Name>
<Letter> ::= 'a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z'
<Digit> ::= '1|2|3|4|5|6|7|8|9|0'
<Mission> ::= <Name> <ID> <Tasks>
<Tasks> ::= <Task> | <Tasks>
<Task> ::= <Name> <ID> <Name>
<ID> ::= <Letter> | <Name>
<Troops> ::= <Troop> | <Troops>
<Troop> ::= 'Troop' <Name> <Location> <Task> 'Commander' <Personnel>
'Trooper' <Personnels> 'Equipment' <Equipments> <MsgManSys> | <Troops>
<Personnels> ::= <Personnel> | <Personnels>
<Personnel> ::= <Name> <Name>
<Equipments> ::= <Equipment> | <Equipments>
<Equipment> ::= <Vehicle> | <Weapon> | <Ammunition>
<Vehicle> ::= 'Vehicle' <EquipmentC> <Weapons>
<Weapons> ::= <Weapon> | <Weapons>
<Weapon> ::= 'Weapon' <EquipmentC> <Ammunitions>
<Ammunitions> ::= <Ammunition> | <Ammunitions>
<Ammunition> ::= <EquipmentC>
<EquipmentC> ::= <ID> <Name>
<Location> ::= 'Location' <Latitude> <Longitude>
<Latitude> ::= 'Latitude' <Degrees> ('N|S') <Minutes> <Seconds>
<Longitude> ::= 'Longitude' <Degrees> ('E|W') <Minutes> <Seconds>
<Degrees> ::= <RealNumber>
<Minutes> ::= <RealNumber>
<Seconds> ::= <RealNumber>
<RealNumber> ::= <UnsignedReal> | "-" <UnsignedReal>
<UnsignedReal> ::= <Digits> | <Digits> "." <Digits>
<Digits> ::= <Digit> | <Digit> <Digits>
<MsgManSys> ::= 'MsgManSys' <Name> <Messages>
<Messages> ::= <Message> | <Message> <Messages>
<Message> ::= 'Message' <ID> <MessageHeader> <MessageInfo>
<Attachments> (<Order> | <Report>)
<Attachments> ::= <Attachment> | <Attachments>
<Attachment> ::= <ID>
<Order> ::= 'Order' ( 'MOVE' | 'FIRE' | 'FREE' | 'MISSION' )
<Report> ::= 'Report' ( 'NBC' | 'ENEMY' | 'AMMUNITION' | 'MINEFIELD' | 'FREE' )
<MessageHeader> ::= 'Header' 'Title' <Name> 'Sender' <ID>
'Receiver' <ReceiverIDs> 'SendDate' <DateTime>
'ExpirationDate' <DateTime> 'Urgency' <Urgency>
'EncriptionType' <EncryptionType>
<ReceiverIDs> ::= <ID> | <ReceiverIDs>
<Urgency> ::= 'HIGH' | 'NORMAL' | 'LOW'
<EncryptionType> ::= 'ALG1' | 'ALG2' | 'ALG3'
<DateTime> ::= <Date> '-' <Time>
<Date> ::= <Year> "/" <Month> "/" <Day>
<Year> ::= <Digit> <Digit> <Digit> <Digit>
<Month> ::= '01|02|03|04|05|06|07|08|09|10|11|12'
<Day> ::= '01|02|03|04|05|06|...|22|23|24|25|26|27|28|29|30|31'
<Time> ::= <HT> ':' <MT> ':' <ST>
<HT> ::= ('0|1' <Digit>) | ('20|21|22|23')
<MT> ::= '0|1|2|3|4|5' <Digit>
<ST> ::= '0|1|2|3|4|5' <Digit>
<MessageInfo> ::=
"Info" <Route> <Region> <FireType> <StartTime>
<StopTime> <NBCType> <Coordinate> <ImpactTime> <DetectionTime>
<DetectedUnit> <EngagementTime> <DetectedAmmunition>
<CoordinateFound> <Borders> <Name>
<Route> ::= 'Route' | <Location> | <Route>
<Region> ::= 'Region' 'Center' <Location> 'Radius' <Digits>
<FireType> ::= '' | 'IMMEDIATE' | 'DEFANSIVE' | 'CONTINUOUS'
<StartTime> ::= '' | <DateTime>
<StopTime> ::= '' | <DateTime>
<NBCType> ::= '' | 'NUCLEER' | 'BIOLOGICAL' | 'CHEMICAL'
<Coordinate> ::= '' | <Location>
<ImpactTime> ::= '' | <DateTime>
<Borders> ::= 'Borders' <Locations>
<Locations> ::= <Location> | <Locations>
<DetectionTime> ::= '' | <DateTime>
<DetectedUnit> ::= 'SQUAD' | 'PLATOON' | 'COMPANY' |
'BATTALION' | 'BRIDAGE' | 'DIVISION' | 'CORPS' | 'ARMY'
<Coordinate> ::= '' | <Location>
<EngagementTime> ::= '' | <DateTime>
<CoordinateFound> ::= '' | <Location>
DetectedAmmunition ::= 'BULLET' | 'BOMB' | 'MISSILE' | 'WARHEAD' | 'MINE'

```

Ek B FULL VERSION OF THE GRAMMAR IN XTEXT FORMAT

```

grammar mydsl.MyDsl with org.eclipse.xtext.common.Terminals
generate myDsl "http://www.xtext.org/example/mydsl/MyDsl"

```

```

Combat :
    "Combat" name=ID
    operationDate = DateTime

```

```

    mission = Mission
    (troops += Troop)* ;
Mission:
    title=STRING
    description = STRING
    tasks += Task+;
Task:
    title=STRING
    description = STRING
    taskID = ID;
Troop :
    "Troop" name=ID
    location = Location
    tasks += Task+
    "Trooper" trooper+=Personnel+
    "Commander" commander=Personnel
    ("Equipment" equipments += (Vehicle | Weapon | Ammunition))*
    msgManSys = MsgManSys
    (subTroops += Troop)*;
Personnel:
    name=STRING
    id = ID;
Equipment:
    id = ID
    name = STRING;
Vehicle:
    "Vehicle"
    Equipment
    weapons += Weapon*;
Weapon :
    Equipment
    ammunitions += Ammunition*;
Ammunition:
    Equipment;
Location :
    "Location"
    "{"
    latitude = Latitude ", "
    longitude = Longitude
    "}";
Latitude :
    degrees=INT ("N"|"S") minutes=INT seconds=INT;
Longitude :
    degrees=INT ("E"|"W") minutes=INT seconds=INT;
MsgManSys :
    "MsgManSys" name=ID
    (messages += (Order|Report))*;
Message :
    "Message"
    id = ID
    messageHeader = MessageHeader
    messageInfo = MessageInfo
    attachments += Attachment*;
Order :
    Message
    "Order" order = OrderType;
Report :

```

```

Message
  "Report" report=ReportType;
MessageHeader:
  "Header"
  "Title" title = STRING
  "Sender" senderID=ID
  "Receiver" (receiverID += ID)+
  "SendDate" sendDate=DateTime
  "ExpirationDate" expirationDate=DateTime
  "Urgency" urgency=Urgency
  "EncryptionType" encryptionType = EncryptionType;
Attachment:
  "Attachment" fileName=ID;
DateTime :
  date=Date "-" time=Time;
Date :
  year=INT "." month=INT "." day=INT;
Time :
  ht=INT ":" mt=INT ":" st=INT;
MessageInfo :
  "Info"
  route += Route?
  region += Region?
  fireType = FireType?
  startTime += DateTime?
  stopTime += DateTime?
  nbctype = NBCType?
  coordinate += Location?
  impactTime+=DateTime?
  detectionTime+=DateTime?
  detectedUnit=DetectedUnit?
  engagementTime+=DateTime?
  detectedAmmunition=DetectedAmmunition?
  coordinateFound+=Location?
  borders=Borders?
  content=STRING; //message content
Route :
  "Route"
  (points+=Location)+ ;
Region :
  "Region"
  "Center" center+=Location
  "Radius" radius=INT;
Borders :
  (borders+=Location)+;
enum OrderType :
  MOVE | FIRE | FREE | MISSION ;
enum ReportType :
  NBC | ENEMY | AMMUNITION | MINEFIELD | FREE;
enum FireType :
  IMMEDIATE|DEFANSIVE|CONTINUOUS;
enum EncryptionType :
  ALG1|ALG2|ALG3;
enum Urgency :
  IMMEDIATE|HIGH|NORMAL|MILD;
enum NBCType :
  NUCLEER|BIOLOGICAL|CHEMICAL;

```

```
enum DetectedUnit :
    SQUAD|PLATOON|COMPANY|BATTALION|BRIDAGE|DIVISION|CORPS|ARMY;
enum DetectedAmmunition :
    BULLET|BOMB|MISSILE|WARHEAD|MINE;
```

EK C

FULL VERSION OF THE SAMPLE VALIDATION

```
@Check
public void checkLatitude( Latitude latitude){
    if( latitude.getDegrees() <0 || latitude.getDegrees() >=90){
        warning("Degrees of Latitude should be in 0-90", MyDslPackage.LATITUDE__DEGREES);
    }

    if( latitude.getMinutes() <0 || latitude.getMinutes() > 59){
        warning("Minutes of Latitude should be in 0-59", MyDslPackage.LATITUDE__MINUTES);
    }

    if( latitude.getSeconds() <0 || latitude.getSeconds() > 59){
        warning("Seconds of Latitude should be in 0-59", MyDslPackage.LATITUDE__SECONDS);
    }
}
```

EK D

SAMPLE CUSTOM JAVA VALIDATION AS OCL

```
*** Validates the valueCheck constraint of '<em>Location</em>'.
* <!-- begin-user-doc -->
* <!-- end-user-doc -->
* @generated NOT
*/
public boolean validateLocation_valueCheck(Location location, DiagnosticChain diagnostics, Map<Object, Object> context) {
    // TODO implement the constraint
    // -> specify the condition that violates the constraint
    // -> verify the diagnostic details, including severity, code, and message
    // Ensure that you remove @generated or mark it @generated NOT
    // violation conditions
    boolean latd = location.getLatitude().getDegrees()->90 && location.getLatitude().getDegrees()<90;
    boolean longd = location.getLongitude().getDegrees()->-180 && location.getLongitude().getDegrees()<180;
    boolean llmin = location.getLatitude().getMinutes()->0 && location.getLatitude().getMinutes()<59;
    boolean llsec = location.getLatitude().getSeconds()->0 && location.getLatitude().getSeconds()<59;
    if (!latd || !longd || !llmin || !llsec) {
        if (diagnostics != null) {
            diagnostics.add(createDiagnostic(Diagnostic.ERROR,
            DIAGNOSTIC_SOURCE,0,"_UI_GenericConstraint_diagnostic",
            new Object[] { "valueCheck", getObjectLabel(location, context) },
            new Object[] { location },context));
        }
        return false;
    }
    return true;
}
```

EK E

FULL VERSION OF THE MODEL

```
<?xml version="1.0" encoding="UTF-8"?>
<tbms:Combat xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:tbms="http://tbms/1.0" operationDate="2455565654">
  <troops name="BlackHawk">
    <location>
      <latitude degrees="44" minutes="32" seconds="44"/>
      <longitude minutes="33" degrees="32" seconds="34"/>
    </location>
    <msgManSys>
      <messages xsi:type="tbms:Order">
        <contains/>
        <header receiverID="1" title="requiring more troopers and ammunition" sendDate="1" expirationDate="2"/>
        <attachments fileName="ammoList.data_enc"/>
      </messages>
      <messages xsi:type="tbms:Report">
        <contains/>
        <header senderID="1" title="no more troopers and ammunition" sendDate="12345678977" expirationDate="12345678989"/>
      </messages>
    </msgManSys>
    <task title="mine field destruction task" description="destroy the mine field"/>
    <equipments xsi:type="tbms:Vehicle" name="Cobra"/>
    <equipments xsi:type="tbms:Weapon" name="MG-3"/>
    <equipments xsi:type="tbms:Weapon" name="G-3" ID="1"/>
    <equipments xsi:type="tbms:Ammunition" name="MG-3 bullets"/>
    <equipments xsi:type="tbms:Ammunition" name="G-3 bullets" ID="1"/>
    <equipments xsi:type="tbms:Weapon" name="Mine Destroyer" ID="2"/>
    <troopers name="trooper0"/>
    <troopers name="trooper1" ID="1"/>
  </troops>
</Combat>
```

```
<commander name="commander" ID="2"/>
</troops>
<mission title="Capture the island" description="Capture the island and destroy all the bases of the enemy"/>
</tbms:Combat>
```