

39534

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

39534

**MİMARİ TASARLAMA EĞİTİMİNDE YAZLIK KONUT TASARIMI AMACIYLA
KULLANILABİLECEK BİR UZMAN SİSTEM 'ZEUS'**

YÜKSEK LİSANS TEZİ

Mimar Gün ALEV

Ana Bilim Dalı: Mimarlık

Programı: Bina Bilgisi

TEMMUZ 1994

**MİMARİ TASARLAMA EĞİTİMİNDE YAZLIK KONUT TASARIMI AMACIYLA
KULLANILABİLECEK BİR UZMAN SİSTEM 'ZEUS'**

YÜKSEK LİSANS TEZİ

Mimar Gün ALEV

Tezin Enstitüye Verildiği Tarih : 13 Haziran 1994

Tezin Savunulduğu Tarih : 4 Temmuz 1994

Tez Danışmanı : Prof. Dr. Nigan BAYAZIT

Diğer Jüri Üyeleri : Prof. Dr. Mete ÜNÜGÜR

Doc. Dr. Belkıs ULUOĞLU

TEMMUZ 1994

Önsöz

Değişen zamanla birlikte mimarlık ta değişmektedir. Problemler farklılaşmaktadır, çözümleri çok çeşitli ve karmaşık hale gelmektedir. Bu çözüm arayışında problemlere birçok değişik açıdan bakma ihtiyacı duyulmaktadır. Bu yüzden farklı açılardan problemlere yaklaşmak, mimarlık eğitiminin ilk günlerinde öğrenilmelidir. Bunun yanında tasarım sırasında birçok yönetmelik ve kurala uyulması gerekir. Hesaplamaların hızla yapılması ve farklı olasılıkların ilk tasarım aşamasında öğrenciye verilmesi, öğrencinin başlangıçta tasarıma daha hızlı girmesini ve sahip olduğu süreyi daha verimli değerlendirmesine imkan tanıyarak, tasarım sürecine faydalı olacaktır. Gelişen programlama teknikleri ve çizim programları sayesinde üretilen ilk eskizlerin tasarımcının yorumuna hazır halde kullanıcıya verilmesi böylece tasarım yapmayı yeni öğrenen öğrenci mimarları desteklenme imkanı olduğunu düşünülmektedir.

Sayın Prof. Dr. Nigan Bayazıt'ın teşviki ve desteği ile bu uzman sistem programı yapılmıştır. Bu araştırma ile ilgili bilgiler ve elde edilen sonuçlar ilerideki sayfalarda anlatılmıştır.

Bu çalışmanın oluşmasına yardımcı olan Sayın Metin Hepgüler'e ve Mar Mimarlık'a teşekkür ediyorum. Ayrıca bütün hayatım boyunca bana destek veren aileme de teşekkür etmek istiyorum.

Haziran 1994

Mimar Gün Alev

İçindekiler

Önsöz.....	ii
İçindekiler	iii
Şekil Listesi	v
Tablo Listesi	vii
Özet	viii
Summary.....	ix
Bölüm 1. Giriş.....	1
Bölüm 2. Uzman Sistemlerden Tasarlama	
Yararlanma.....	3
2.1. Bilgisayar Destekli Tasarım	3
2.2. Uzman Sistemler.....	4
Bölüm 3. Çalışmada Uygulanan Sistemin Modeli.....	8
3.1. Literatür Çalışmaları	9
3.2. Tasarım Problemi Konusunda Bilgiler ve	
Kabuller,.....	11
3.3. Uzman'dan Bilgi Çıkarımı	12
3.4. Bulgular ve Bulguların Analizi (Protokol Analizi).....	14
3.5. Programın Hazırlanması.....	16
Bölüm 4. Uygulama	19
4.1. Uzman'dan Bilgi Çıkarımı	24
4.2. Bulgular ve Bulguların Analizi (Protokol Analizi).....	25
4.3. Tasarım Stratejileri (Bulgular)	28
4.4. Programın Hazırlanması.....	45
Bölüm 5. Sonuç	84
Kaynaklar	88

Ekler	90
Ek A. Tasarım Stratejisi A ve B için, Mekan Alanları ve Kenar Uzunlukları Oranları	90
Ek B. Konut Yönlendirme Şeması	91
Ek C. Programlama Dilleri Klavuzu	92
Ek D. UM'a verilen açıklayıcı föyler.....	103
Ek E. Oturumun Metni ve Yapılan Örnek Tasarımlar	107
Ek F. Kurallar Tablolarında Kullanılan Kısaltmalar.....	122
Ek H. Bilgi Girişi için Program Denemesi	123
Özgeçmiş	124



Şekil Listesi

Şekil 3.1. Uzman Sistem Oluşturma Modeli.....	8
Şekil 3.2. Literatür Çalışmaları.....	9
Şekil 3.3. Tasarım Stratejisi A Örneği.....	10
Şekil 3.4. Tasarım Stratejisi B Örneği.....	11
Şekil 3.5. Uzman'dan Bilgi Çıkarımı Çalışmaları.....	12
Şekil 3.6. Bulguların Analizi Çalışmaları	14
Şekil 3.7. Uzmandan Bilgi Alma Modeli.	14
Şekil 3.8. Programın Hazırlanması.	16
Şekil 3.9. Zeus Programı Akış Diyagramı	17
Şekil 4.1. Uygulama	19
Şekil 4.2. Literatür Bilgileri	20
Şekil 4.3. Uzmandan Bilgi Çıkarımı Bulguları	24
Şekil 4.4. Bulguların Analizi.....	25
Şekil 4.5. Eskiz 1	30
Şekil 4.6. Eskiz 1'in Bilgisayar Çizimi.	32
Şekil 4.7. Eskiz 2	33
Şekil 4.8. Eskiz 2'in Bilgisayar Çizimi.	33
Şekil 4.9. Eskiz 4	34
Şekil 4.10. Eskiz 4'ün Bilgisayar Çizimi.....	36
Şekil 4.11. Eskiz 5	37
Şekil 4.12. Eskiz 5'in Bilgisayar Çizimi.	38
Şekil 4.13. Eskiz 8	39
Şekil 4.14. Eskiz 8'in Bilgisayar Çizimi.	40
Şekil 4.15. Eskiz 9	41
Şekil 4.16. Eskiz 9'un Bilgisayar Çizimi.	42
Şekil 4.17. Eskiz 19	43

Şekil 4.18. Eskiz 19'un Bilgisayar Çizimi.	44
Şekil 4.19. Programın Hazırlanması.	45
Şekil 4.20. Standartlar Sorgu Kutusu.	46
Şekil 4.21. Kuzey Yönü Sorgu Kutusu.	47
Şekil 4.22. Hakim Rüzgar Sorgu Kutusu	50
Şekil 4.23. Manzara Yönü Sorgu Kutusu.....	52
Şekil 4.24. Mekan Alanları Sorgu Kutusu.....	53
Şekil 4.25. ZEUS Ana Sorgu Kutusu	59
Şekil 4.26. ZEUS Ana Sorgu Kutusu	60
Şekil B.1. Konut Yönlendirme Şeması.	91
Şekil D.1. Arsa Krokisi.	105
Şekil D.2. Vaziyet Planı.	106
Şekil E.1. Eskiz 1.	107
Şekil E.2. Eskiz 2.	108
Şekil E.3. Eskiz 3.	109
Şekil E.4. Eskiz 4.	109
Şekil E.5. Eskiz 5.	110
Şekil E.6. ve 7. Eskiz 6 ve 7.	111
Şekil E.8. Eskiz 8.	111
Şekil E.9. Eskiz 9.	113
Şekil E.10. Eskiz 10	114
Şekil E.11. Eskiz 11 Eskiz 10'daki evin perspektifi.	114
Şekil E.12. Eskiz 12 Eskiz 10'daki evin ön görünüşü.....	115
Şekil E.13. Eskiz 13 Çatı planı.....	116
Şekil E.14. Eskiz 14 Çatı Görünüşü.....	116
Şekil E.15. Eskiz 15	117
Şekil E.16. Eskiz 16	117
Şekil E.17. Eskiz 17	118
Şekil E.18. Eskiz 18	119
Şekil E.19. Eskiz 19	119
Şekil E.20. Eskiz 20_ Banyo örneği.....	120

Tablo Listesi

Tablo 4.1. Mimarla yapılan oturumda çıkarılan kuralların bazıları	27
Tablo 4.2. Eskiz 1 ve 2'nin Kural Dökümü.	30
Tablo 4.3. Eskiz 1'in Tasarım Stratejisi	32
Tablo 4.4. Eskiz 1'in Alan ve Mekan Boyutları.	33
Tablo 4.5. Eskiz 2'nin Tasarım Stratejisi.	33
Tablo 4.6. Eskiz 2'nin Alan ve Mekan Boyutları.	34
Tablo 4.7. Eskiz 4'ün Kural Dökümü.....	35
Tablo 4.8. Eskiz 4'ün Tasarım Stratejisi.	36
Tablo 4.9. Eskiz 4'ün Alan ve Mekan Boyutları.	36
Tablo 4.10. Eskiz 5'in Kural Dökümü.....	37
Tablo 4.11. Eskiz 5'in Tasarım Stratejisi.	38
Tablo 4.12. Eskiz 5'in Alan ve Mekan Boyutları.	38
Tablo 4.13. Eskiz 8'in Kural Dökümü.....	39
Tablo 4.14. Eskiz 8'in Tasarım Stratejisi.	40
Tablo 4.15. Eskiz 8'in Alan ve Mekan Boyutları.	41
Tablo 4.16. Eskiz 9'un Kural Dökümü.....	42
Tablo 4.17. Eskiz 9'un Tasarım Stratejisi.	42
Tablo 4.18. Eskiz 9'un Alan ve Mekan Boyutları.	43
Tablo 4.19. Eskiz 9'un Kural Dökümü.....	43
Tablo 4.20. Eskiz 19'un Tasarım Stratejisi.	44
Tablo 4.21. Eskiz 19'un Alan ve Mekan Boyutları.	45
Tablo A.1. Konutun Strateji A ve B için, Toplam Mekan ve Kenar Alanları Oranları.	90
Tablo F.1. Kurallarda Kullanılan Kısaltmalar.	122

Özet

Zeus Paket Programı, mimarlık öğrencilerine yönelik olarak hazırlanan bir Uzman Sistem'dir. Program, kullanıcılarına tasarım aşamasında destek vermek ve uyulması gereken kurallara, yönetmelik ve kanunlara dayanarak yön göstermek amacıyla tasarlanmış bir yardımcı araçtır. Tezin amacı yerine getirilirken, mevcut kurallardan, yönetmeliklerden ve kanunlardan başka yazlık konut tasarımı ve yapımı konusunda deneyimli bir mimarın bilgilerinden ve deneyimlerinden yararlanılmaktadır. Bu program mimarlık öğrencisinin tasarım sürecinin süresini kısaltacak ve onu destekleyecektir.

Tezde ilk olarak uzman sistemlerden tasarlama nasıl yararlanılacağı konusu üzerinde durulduktan sonra bilgisayar destekli tasarım veri tabanlarının (databases) kullanımı olarak tanımlanmaktadır. Uzman sistemler geleneksel programlardan farklıdır ve çeşitli tipleri vardır. Bunlar çevirme, tahmin, tanı, tasarlama, planlama ve izleme sistemleridir. Biz bu tez kapsamında tasarlama sistemlerini incelenmiştir.

Uzman Sistem'in incelenmesinden sonra tezin planı ve yapılan çalışmaların modeli kurulmuştur. Daha sonra bu plan ve program dahilinde yapılan incelemeler ve bulgular göz önüne alınmıştır.

ZEUS kapsamında, uzman mimar olarak, Mimar Metin Hepgüler'den yardım istenmiştir. Uzman mimarla yapılan 30 dakikalık oturum süresince, mimarın yaptıkları videoya kaydedilmiş ve mimarın yaptığı eskizler alınmıştır. Yapılan bu çalışmaların ardından, ZEUS uzman sisteminin bilgisayar ortamına aktarılmasına başlanmıştır.

Oluşturulan ZEUS US, hedeflenen amaca ulaşmıştır. Fakat bu örnek çalışma başta yapılan sınırlamalar ve kabuller doğrultusunda çok çeşitli konut tiplerine hizmet verememektedir. Bundan sonraki amaç, bu tezde yapılan sınırların kaldırılarak veya en azından hafifletilerek seçilecek konut tiplerinin daha detaylı olarak UM'dan alınması ve bunun konut tiplerinde kullanılması olabilir. Bunun gerçekleştirilmesiyle US daha geniş bir konut tipi ve arsa şartların da kendisinden beklenenleri yerine getirebilecektir. Eğer bu tür bir çalışma gerçekleştirilirse, hedef kullanıcılar sadece mimarlık öğrencileriyle sınırlı kalmayacaktır. Böylece ülkemiz mimarlığına destek verici uzman sistemlerde destek verir duruma gelecektir.

Summary

'An Expert System is based on to Design Summer Houses which can be used at the Architectural Design Education' (ZEUS=Mimari Tasarlama Eğitiminde Yazlık Konut Tasarımı Amacıyla Kullanılabilecek bir Uzman Sistem).

ZEUS is an expert system packet program which has been developed to be used by the architecture students while designing, not only helping them with it's design rules and regulations but introducing them a certain design environment and to make use of the knowledge of an experienced architect on a summer house design. This program will shorten the designing period of a student and will support him or her.

In the first section, after we consider how one can use the expert systems while designing later computer aided design, databases use is defined. Declarative knowledge and processing information are used to transform under certain conditions. Through database experience, declarative and processing knowledge are transformed to computer. Transformed regulations are the samples of the complied knowledge and experiences to computer. Regulations which are based on knowledge based systems occur as 'IF condition THEN action'. This kind of regulations are used in expert systems. Transition from artificial intelligence to expert systems is the result of the studies of intelligency put into computer programs. The computer programs take form with high level expert knowledge in one area. In order to be successful in such programs, different type disciplines are required. With the help of such

disciplines one should gather high quality knowledge from an expert and can analyze it. Expert systems differ from traditional programs and have different types. These are interpretation, prediction, diagnosis, designing, planning and monitoring systems. In this research we will examine the designing expert systems carefully.

The second section contains the method. At first expert systems take form and later each method is explained in detail. The knowledge taken from literature has been placed under three sub-titles. The first type of acknowledgment which is used for expert system design, creation information. The second type gives information on ZEUS. In this section samples which can be found in literature are examined. As a result of these studies, specific restrictions are determined and related to summer house design. Conserving the design information of the two types of design strategies, space orientation regulations and space area and dimensions have taken place.

In the light of the information gathered, a sample field has been brought up keeping the needs of the client in mind in order to get dependable information from an expert architect. The program on expert system and program language information have been gathered under this third topic from an expert architect, the duration of the project location and the dimensions of the building are gathered.

After doing the research in literature, information on design process, space orientation and space dimensions are gathered from an expert architect. In order to gather the above mentioned information an expert architect has been carefully examined while designing; 'think aloud' technique has been used for the preparation of the plan. The aim of this technique is to clarify what the expert architect thinks and what sort of regulation he uses while designing.

In order to achieve the ZEUS program, Architect Metin Hepgüler has been asked to help as an expert. In a 30 minutes session architect sketches were taken to video. This study was later examined further in detail and the verbal information was put into written text.

The third part of the research is used for the explanation of the experiment. During the 'think-aloud' technique session, it has been recognized that restrictions and regulations are means of limitations for the architect.

During the 'think-aloud' technique session the restrictions and regulations were handy_cape for the expert so he was asked to develop his designs without some of the regulations given. In conclusion, the expert solved six different design strategies.

After the 'Think-aloud' session, the interview was put into text and his sketches were gathered and analyzed. In the first step of the analysis the rules (regulations) used by the expert were put down.

The interview text was classified as procedures, regulations and preference sentence groups. Later it was again classified by design regulations. The list of regulations has been organized such away that it could be used later in the program. The series of regulations have been classified according to the concerned group of sketches. In this way each sketch design strategy is obtained. Further more, these sketches have been calculated according to their dimensions and have been put into a form which could be used in the program.

After this study, ZEUS expert system was transferred to computer to computer system circumstances.

ZEUS program starts with an inquiry dialogue box for architecture student as the user. The inquiry main dialogue box asks the user about the field, the house, directions, house settlement, which he would use. ZEUS controls the knowledge which is gathered from the user. If there is undesired design regulations, it would state them, would ask for new values and would also state reasoning. After entering the values, it asks the user to enter the design strategies. At this stage the user must choose at least one of the design strategies which has been coded before.

ZEUS materializes the chosen design strategy either from the literature or from the expert point of view of the values and regulations. After the program materializes the desired design, it will form sample sketch projects.

This program and the sketches are in the format of AutoCAD R12 for DOS and/or AutoCAD R12 for Windows packet program. Hence, the user will be able to study the designed samples and will be able to change it according to the user's wishes.

After designing the sketch strategy the program closes itself. Thus the user gets the values according to the given sample designs.

ZEUS Expert System has reached to the goal which is desired. But this sample study can not meet the desired assents of different types of housing projects. In addition to this, it shows that it will help the efficiency of different types of housing projects in the future. Here, the next goal should be either to remove the limits or at least to relieve the strict rules of housing types. Details can be taken from expert architect and can be used at different type of housing.

Through the realization of expert systems, larger type of housing projects and field can meet the expectations. If such study can be realized, it will not be limited by only the expected architecture students. Thus, the architecture in our country will also receive to get help from Expert Systems.



BÖLÜM 1. GİRİŞ

Mimarlıkta tasarım süreci geliştikçe ve karmaşıklıklaştıkça, tasarım süreci üzerinde yapılan çalışmalar genişlemekte ve tasarım süreci hakkında bilgilerimiz fazlalaştıkça, tasarım sürecinin benzetimlerinin yapılması çalışmaları gün geçtikçe artmaktadır. Bu çalışmaların ışığında yürütülen bu tasarım araştırması bir mimarın tasarlama sürecini bütünüyle kopyalamak yerine onun tasarım sürecine nasıl başladığının ve süreci nasıl kullandığının açılımını çıkararak, elde edilen sonuçlardan mimarlık öğrencilerinin kullanabileceği bir tasarım aracı oluşturmak, öğrencilere tasarım aşamasında seçenekler sunmak ve deneyimli mimarların tasarıma nasıl yaklaştığını aktarmaya çalışmaktadır.

Tezin amacı doğrultusunda tez konusunun 'Yazlık Konut' olmasına karar verilmiştir. Konu üzerinde yapılan araştırmalarda, konut hakkında biçimsel ve mekan sayıları ile ilgili bazı kısıtlamaların ve kabullerin yapılmasının hedeften uzaklaşmamak için gerekli olduğu görülmüş ve bu konularda kabuller yapılarak konunun sınırları belirlenmiştir.

Tezin adı 'Mimari Tasarlama Eğitiminde Yazlık Konut Tasarımı Amacıyla Kullanılabilecek bir Uzman Sistem'dir. Kısaca ZEUS adı verilmiştir.

Çağın yoğun çalışma temposu sonucu ortaya çıkan zaman yetersizliği, tasarım ve yapım sürecine ayrılan zamanı da kısaltmıştır. Bu zaman yetersizliği tasarımcıların hata yapma olasılıklarını çoğaltmış, yapılan hataların düzeltilme fırsatını azaltmıştır. Bu tezin amacı, bina tasarlayıcılarının karşılaştığı sorunları minimuma indirmek ve tasarım sürecini kısaltarak tasarımcının zamanını daha verimli kullanmasını sağlamaktır. Bilgisayar kullanımı mimari tasarımda her geçen gün daha fazla yer almaktadır. Amaçlanan, bilgisayar kullanımını geliştirmek ve bilgisayarı sadece çizim aracı olmaktan çıkarıp, tasarıma destek veren ve gelişimine olanak sağlayan bir araç olarak mimarlık mesleğine tekrar ve daha yararlı bir biçimde kazandırmaktır.



BÖLÜM 2. UZMAN SİSTEMLERDEN TASARLARMADA YARARLANMA

2.1. Bilgisayar Destekli Tasarım

Bilgisayar destekli tasarım, veritabanlarının (databases) kullanımıyla tanımlanmıştır. Bu veritabanları belirli şart altında ele alınan nesnelerin topolojik ve geometrik özelliklerine ait sintatik bilgilerdir ^[1].

Deneyimin tasarımda etkili bir yeri vardır. Deneyimli tasarımcılar mevcut sorunların çözülmesinde geçmiş tasarımlarındaki standart detaylar gibi verimli çalışan çözümlerinden de yararlanırlar. Tasarımın incelenmesinde kullanılan mevcut yaklaşımlar, tasarım sürecinin prototipleri gibi kurallar, çerçeveler ve şemalardan toplanmış bilgilerin (compiled knowledge) tekrardan sunulmasına dayanmaktadır ^[2].

Toplanan veya genel bilgi, tek tek toplanan deneyim bilgilerinden oluşur. Tasarımcı da, verilen probleme kendi bilgileri yardımıyla, o duruma uygun çözümler getirir. Toplama bilgileri, bir dizi durumları kapsayan bilginin genel bir biçimidir ^[3].

Kurallarsa, toplanmış bilgilerin örnekleridir. Bilgiye dayanan sistemlerde kurallar,

Eğer 'şart' ise 'sonuç'

olarak yer alırlar. Mevcut durum, 'Eğer' arkasındaki şartı 'Eğer' yürürlükteyken sağlıyorsa, 'sonuç' devreye girer ve o durumda yapılması gereken işlemler yapılır. Bu şartlar belirli durumlara uyar ve sonuçlar gerçekleri belirttiği gibi, yapılması gereken hareketleri de içerebilir. Uzman Sistemler (US=Expert System) bu tip kuralları genellikle keşfetme yöntemi veya bütünün kuralları olarak kullanırlar [4].

Durum tabanlı akıl yürütmeler (Case based reasoning) Yapay Zeka (ZY=Artificial Intelligence) içinde iyi tanımlanmış sistemlerdir [5].

Durum tabanlı akıl yürütmeler mevcut sorunu hafızadaki en uygun deneyime bağlar ve deneyimin problemi çözme işlemini kullanır. Bu yaklaşım, bilgisayarın yaklaşımdan çok bellek tabanlı yaklaşım olmasına rağmen, yeniden çözüm yerine daha önce çözülmüş duruma göre sonuç getirmedi. Bu yaklaşım bu tez kapsamında kullanılan işleme daha çok uymaktadır.

2.2. Uzman Sistemler

US'ler, YZ olarak adlandırılan alanın bir alt bölümüdür. 1960'larda bilim adamları çok geniş problem gruplarını çözmek için genel metotlar bularak, karmaşık düşünme sürecini benzeştirmeye çalışmışlar ve bu metotları genel amaçlı programlarda kullanmışlardır. 1970'li yılların sonlarına doğru bilim adamları YZ içinde bazı önemli konular olduğunu anladılar. Uygulamalı yapay zekanın alanı yapay zekanın alanından daha geniş olup, biliş bilimini (BB=Cognitive Science) , toplum ve yönetim bilimlerini de içine alır [6]. Bunun gerçekleştirilmesiyle birlikte özel amaçlı, çok dar bir problem alanında uzman olan bilgisayar programlarının ve bilgisayar sistemlerinin gelişmesine olanak sağlandı.

YZ alanında US'lerin ortaya çıkmasını sağlayan fikir Aterman'ın 1986'da söylediği gibi 'Bir programı zeki kılmak istiyorsanız, onu bazı problem alanlarında yüksek kalitede ve özel bir bilgi yoğunluğuyla destekleyiniz.' [7] sözüdür. Kısıtlı bir problem alanında yüksek düzeyde bir performans elde etmek için uzman bilgisini kullanan bilgisayar programlarına US denir. Bu uzman sistem oluşturma işlemi de BİLGİ MÜHENDİSLİĞİ (BM=Knowledge Engineering) olarak adlandırılmaktadır. Bir başka deyişle, bilgi mühendisliği uzmandan bilgisayara bilgi aktarma işidir ve bazı problem alanlarında uzman kişilerle, bilgi mühendisi arasındaki özel bir etkileşim şeklini içerir.

Sonuç, bir bilgisayar programı olup, bir alanda bir uzmanın başardıklarını belirli bir düzeyde başaran bir sistemdir. Yalnız US'lerin programları bilinen türdeki geleneksel bilgisayar programlarından farklıdır ve bazı ek gereksinimleri vardır. Bilgi tabanlı sistemler, insan bilgisinin incelenebilir belli bir konuda ele alındığı ve yeniden sunulduğu bilgisayar programlarıdır. Burada hedef, bilgiyi hem bilgisayarın hem de insanın anlayabileceği sunuş biçimine ulaştırmaktır. Kendine has, egemen elemanların oluşturduğu bilgi ile çalışan US'ler genellikle Uzman Sistemler olarak adlandırılırlar. Tasarımdaki bilgi bir çok kuraldan oluşmaktadır. US, işlem gösterici bilgilerin sunulması ile yaratılmalıdır.

US tipleri; US'ler, üzerinde çalışılan problemlerin tiplerine göre farklı amaçlara yönelirler ve yapılanmaları da ona göre değişir. US tiplerini aşağıdaki gibi ayırabiliriz [8].

Çevirme Sistemleri; Bu sistemler gözlenenlerden durum tanımı çıkarır. Bu kategoriye keşif, konuşma, anlama, görüntü analizi, kimyasal yapıları açıklama, sinyal anlama gibi entelektüel analizlerin herhangi biri girebilir. Bu sistemde gözlenen verilere sembolik anlamlar vererek ve elde edilenler üzerinde hesap yapılarak durum açıklanır.

Tahmin Sistemleri; Bazı durumlar sonucunda çıkabilecek olanları bulur. Bu kategoriye, hava tahminleri nüfus tahminleri, trafik tahminleri, askeri gelecek tahminleri gibi konular girer. Tahmin sistemi parametrik dinamik bir model kullanır. Bu modelde parametre değerleri, verilen duruma göre değişir. Modelden elde edilen sonuçlar tahmin için esas oluştururlar. Bu tahmin sistemleri çok sayıda senaryoyu ortaya koyabilirler.

Tanı Sistemleri; Gözlemlere dayanarak daha çok bozuklukları bulan sistemlerdir. Tıbbi, elektronik, mekanik ve yazılım tanıları bu kategoriye girer. Tanı sistemleri gözlenen davranış bozukluklarının altında yatan nedenleri bulmakla ilgilidir. Bu amaçla iki teknikten biri kullanılır. Birincisinde esas olarak, davranış ve tanı arasındaki bağlantılara dayanan tablolar kullanılır. İkincisi gözlemlerle, bozukluğa neden olma olasılığı olan tasarım, yapım ya da bileşenlerdeki, olası hata bilgileriyle, sistem tasarım bilgilerini birleştirir.

Tasarlama Sistemleri; Tasarlama sistemi, tasarlama kanıtlamalarını sağlayan nesne konfigürasyonlarını geliştirir. Bu programlar elektronik devre planları, bina tasarımı ve bütçe ile ilgilidir. Tasarlama sistemleri nesnelerin birbirleriyle çeşitli ilişkilerinin tanımını yapar. Ek olarak bir çok tasarlama sistemi olasılık yasasının maliyet ve istenmeyen özellikleri ölçen amaç fonksiyonunu minimize etmeye çalışır. Tasarım problemine bu bakış açısı, onun amaca yönelik davranışını da ortaya koyar.

Planlama Sistemli Tasarım Eylemleri; Bu sistemler, fonksiyona sahip nesnelerin tasarım problemleri üzerinde özelleşmişlerdir. Bunlar otomatik programlama, robotlar, projeler, deney ve askeri planlama problemleriyle ilgilidir. Planlama sistemleri, planlanan eylemlerin etkilerini görmek için eylem modellerinden yararlanırlar.

İzleme Sistemleri; Bu sistemler, sistem davranışları gözlemleriyle, planlama sonuçlarının başarısı için, zorunlu görülen

özellikleri karşılaştırır. Bu önemli özellikler ya da zayıflıklar planın olası bozuklukları ile ilgilidir ve sistemin gidişi konusunda fikir verir.

Bilgi tabanlı uzman sistemler bir bilgisayar programı olup, bir uzmanın uzmanlık alanındaki nesneler, olaylar, durumlar ve hareket tarzları konusundaki bilgilerini kapsar. Bu bilgiler insan uzmanların bir alandaki muhakeme süreçleriyle ilgilidir. Bu sistem dışlaştırılmış olarak kodlanan bilgiye dayanan bir bilgi tabanı olmaktadır. Kullanılan çıkarım metodunda 'zeka' ya da 'kontrollü arama' yolları kullanılmaktadır.

US'ler istatistiksel bulguları ya da problemle ilgili olarak insan uzmanların bilgilerini daha verimli kullanabilirler. Bu bilgi kaynaklarından herhangi biri olmadıkça bu sistemler verimli çalışamazlar. Bir uzman sistem belirli parçalardan oluşur:

- a) Bilgi Tabanı,
- b) Çıkarım mekanizması,
- c) Kullanıcı programı.

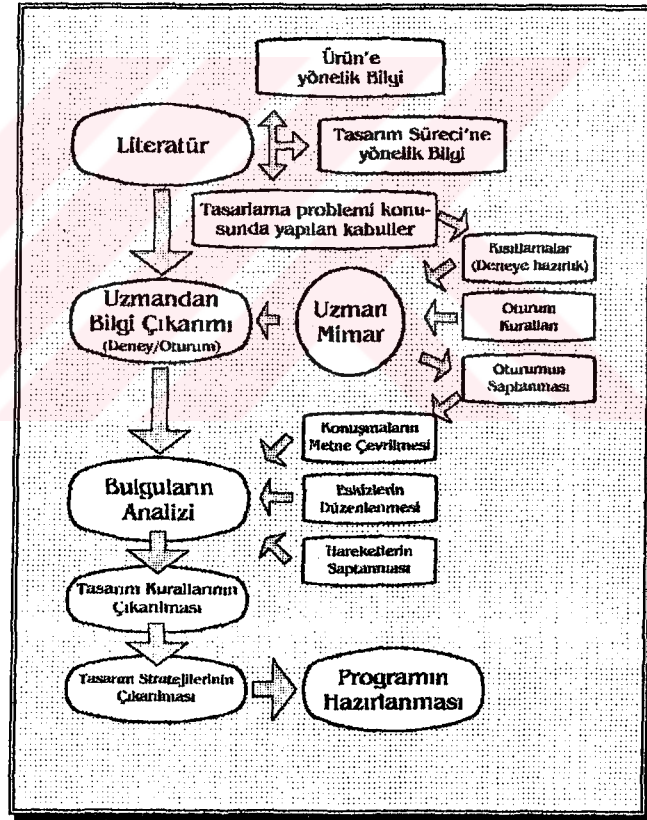
Ayrıca uzman sistemler kullandıkları yaklaşım şekillerine göre ikiye ayrılırlar:

- a) Kural tabanlı uzman sistemler,
- b) Model tabanlı uzman sistemler.

Mimari tasarım için gerekli bilgiler, değişik formlardan gelen bilgilerdir. BM'nin yaptığı son araştırmalar da bilginin iki katagoriye ayrıldığını öne sürülmektedir: Deklaratif ve süreçle yönelik bilgiler. Deklaratif bilgiler nesnelerin nasıl olduğunu anlatan bilgilerdir.^[9] Bu bilgiler nesneleri oluşturan bilgileri kapsar. Bunlar ofis, ışık, bina, kalem gibi nesnelerin özellikleri (işlevsel, pahalı, çekici) ve kendileri arasındaki ilişkiyi kapsar (yeşil araba, ilginç bina, işlevsel olmayan fakat ilginç bina). Bunun yanında işlemci bilgi, hareketleri tanımlar ve önceden haber veya bir hareketin programını verir ^[10].

BÖLÜM 3. ÇALIŞMADA UYGULANAN SİSTEMİN MODELİ

Bu çalışmanın amacı, bir uzmandan alınacak bilgilere dayanarak, tasarlama eğitimi amaçlı bir program hazırlamaktır. Bu amaçla bir çalışma modeli hazırlanarak, araştırma bu çerçevede yürütülmüştür. Bu model Şekil 3.1.'de verilmektedir. Modelin bileşenleri aşağıda sırayla açıklanmaktadır. Modelin sol sütunu yapılan işlemlerin gruplanması ve sınıflandırılmasına yer vermektedir.



Şekil 3.1. Uzman Sistem Oluşturma Modeli

hazırlanması hakkında araştırma yapılması gerçekleştirilmektedir. US'in çerçevesine de bu aşamada karar verilmektedir.

2. Tasarım Sürecine Yönelik Bilgiler,

US'in tasarlama sürecinin nasıl gerçekleştireceği ile, bilgiler bu aşamada derlenmiştir. Bu bilgiler kaynaklarda belirtilen literatürlerden toplanan Tasarım Stratejilerini de kapsamaktadır. Ayrıca bu aşamada UM'dan bilgilerin nasıl alınacağına ve analiz edileceğine karar verilmiştir.

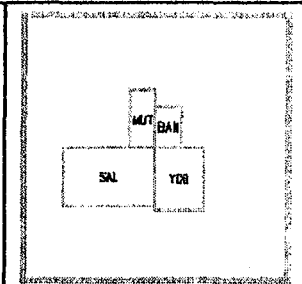
3. Tasarım Stratejileri

Uzman Mimar'dan bilgi alma aşamasının verimliliğini arttırmak ve olası tasarım stratejilerini belirlemek amacıyla mevcut literatür incelenmiştir. Mevcut tez kapsamında bilgisayara aktarılması uygun bulunan iki adet tasarım stratejisi ortaya çıkarılmıştır.

A. Tasarım Stratejisi

Bu Tasarım Stratejisi Ulrich Flemming, Robert F. Coyne, Timothy, Hung Hsi ve Micheal D. Rychener'in 'A Generative Expert System for the Design of Building Layouts' Carnegie Mellon Üniversitesindeki raporlarından çıkarılmıştır.

- Arazi koordinat sistemi şeklinde 4 parçaya bölünmelidir.
- Oluşan her parçaya bir mekan yerleştirilir. Bu mekanların bir köşesi daima koordinat sisteminin 0,0 noktasında olmalıdır.
- Hiçbir mekan kesişmeyecektir.
- Mekan alanları Literatürden çıkarılan ortalama metrekare oranlarından alınmaktadır. (Ek A)
- Mekanların hangi parçada olacağı Ek B'deki yönlendirme kurallarına göre yapılmaktadır.
- Mekanların parçalara yerleştirilmesinde izlenecek sıra; Salon, Mutfak, Banyo ve Yatak Odası'dır.

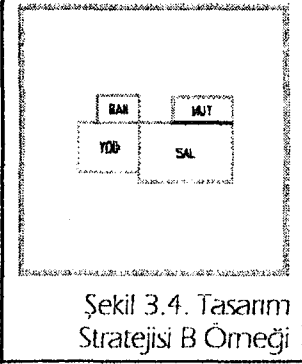


Şekil 3.3. Tasarım Stratejisi A Örneği

B. Tasarım Stratejisi

B. Tasarım stratejisi, literatür çalışmaları sırasında tez yürütücüsü ile üretilen tasarım stratejisidir.

- Mekanlar birbirine en az bir kenar doğrultusunda birleşmelidirler.
- Mekanlar Ek B'deki yönlerde en uygun doğrultuda yerleştirilmelidir.
- Mekan alanları Literatürden çıkarılan ortalama metrekare oranlarından alınmaktadır. (Ek A)
- Mekanların parçalara yerleştirilmesinde izlenecek sıra; Salon, Mutfak, Banyo ve Yatak Odası'dır.



3.2. Tasarım Problemi Konusunda Bilgiler ve Kabuller,

Yapılan deneysel çalışmanın amacı göz önünde bulundurularak; hedeften uzaklaşmamak için literatür taranması, mevcut benzer örneklerden uygun bilgilerin toplanması bu aşamada gerçekleştirilmektedir. Ancak, tezin amacı program oluşturmak olduğu için tasarlama problemi konusunda kabuller yapılmıştır.

UM'la yapılacak oturum öncesinde yapılan örnek incelemesi çalışmalarında tez konusunun genişliği tezin kapsamının sınırlandırılması yönünde adımlar atılmasına neden olmuştur. Bu tezin çıkış noktası Mimarlık öğrencilerine, konusunda UM'ların tasarım olgusuna nasıl yaklaştığını göstermek olduğuna göre, burada amaç, mevcut yazlık konutların formlal veya alansal analizi değildir. Ayrıca söz konusu olan yazlık konutun da biçimsel ve/veya mekansal olarak sınırlanması gerektiği araştırmalar sırasında ortaya çıkmıştır.

Üzerinde kabul yapılan konular; arazi, alan, boyut, biçim ve koşullar gibi tasarımı etkileyen alt problemlerdir.

Bunun için UM'dan randevu alınarak kendisiyle yapılmak istenenler anlatılmıştır. Literatür araştırmaları sırasında tasarlama problemi olan küçük tatil konutuna ilişkin elde edilen bilgilere dayanarak tasarlama kısıtlamaları (arsa boyutları, mekan sayısı, mekan formları vb.) konularında kabuller yapılmıştır. Bu kısıtlamalar tasarım verileri olarak UM'a verilmiştir. UM'dan öce bu kısıtlamalara uyması istenmiştir. Oturumun ilerleyen aşamalarında kısıtlamaların özellikle bina programına ve biçimine ilişkin olanları kaldırılarak tasarımın iki türlü yapılmasına olanak sağlanmıştır. Tezin ilerleyen aşamalarında anlatılan UM'dan alınan tasarım stratejileri iki alt başlığa bölünmüştür. Bunlar;

1) Kısıtlamalara göre UM'ın yaptığı tasarımlar,

2) Kısıtlamalar kaldırıldıktan sonra UM'ın yaptığı tasarımlardır.

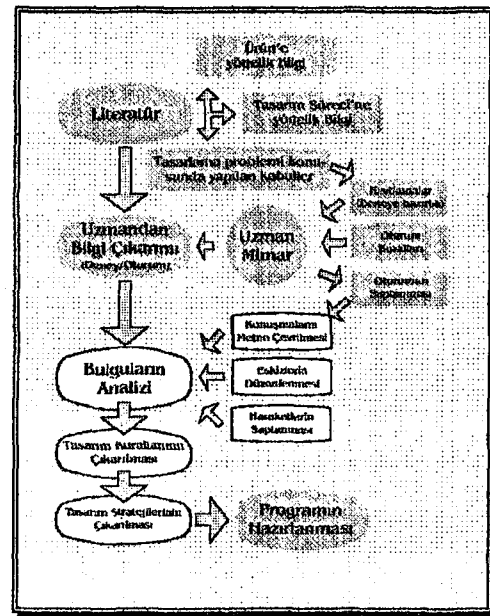
Oturum kuralları; oturum sırasında kullanılacak tespit araçları, oturumun süresi, oturumun nasıl yapılacağı ile ilgilidir. Tasarımcıya oturum sırasında kendisinin ve tesbit yapan ekibin neler yapacağı anlatılmıştır.

'Think-Aloud' yöntemi oturumun teyp veya video gibi araçlarla kaydedilmesi ve UM'ın oluşturduğu örneklerin alınmasıdır. Bu amaçla UM tasarım sırasında kağıt, kalem ve cetvel kullanmış, araştırma ekibi de video ile tesbit yapmıştır.

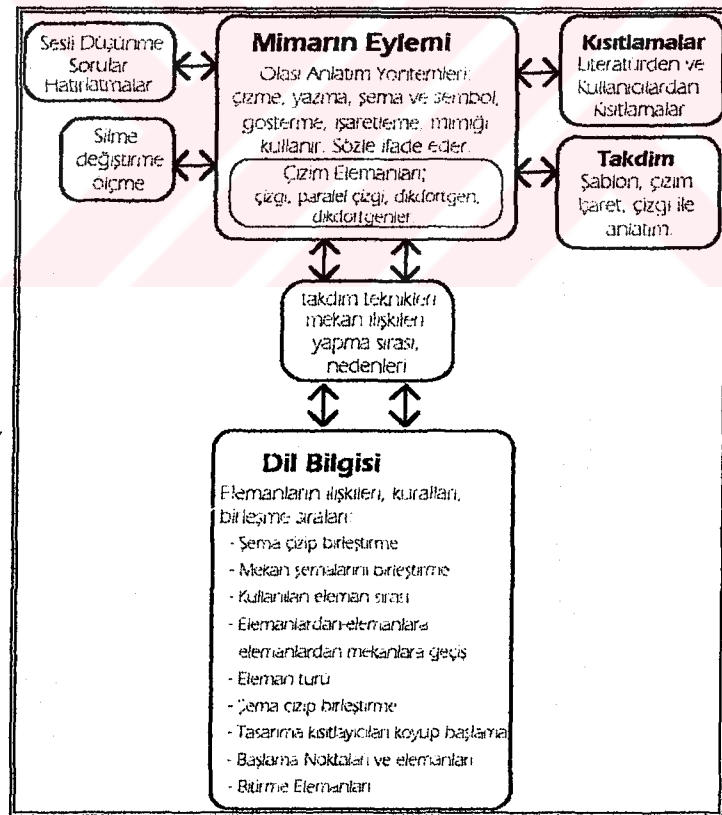
Oturum aşaması UM'ın daha önce belirlenen kurallarla deneye katıldığı aşamadır. UM'a belirlenen tasarlama problemi konusunda kısıtlamalar, oturumun kuralları ve oturumun nasıl yapılacağı anlatıldıktan sonra, oturuma başlanmıştır.

UM'ın oturum sırasında yaptıklarının saptanması sonucunda US oluşturulması için gereken bilgilerin derlenmesi sürecinin başlangıcıdır. Bu aşamadan sonra bilgiler analiz edilerek US'de kullanılabilinecek kurallar oluşturulacaktır.

3.4. Bulgular ve Bulguların Analizi (Protokol Analizi)



Şekil 3.6. Bulguların Analizi Çalışmaları



Şekil 3.7. Uzmandan Bilgi Alma Modeli.

Bu aşamada protokol analizinin tam yapılabilmesi için gerekli ön hazırlıklar yapılmıştır. Bunlar aşağıda anlatıldığı gibi çeşitli aşamalardan oluşmuştur. Bu aşamalar konuşma metninin çıkarılması, eskizlerin sıralarına ve gruplarına göre ayrılması ve oturum sırasında mimarın neyi ne zaman yaptığının ve eskizlerdeki mekanların ele alınış sıralarının saptanmasıdır.

Daha sonra oturumda saptanan ses kayıtları metne çevrilmiştir. Metne çevirim sırasında tez ile ilgili olmayan konuşmalar çıkarılır. Metnin anlam bütünü ifade eden cümlelere ve sözcük gruplarına ayrılması işlemi yapılır.

Oturum sırasında UM'ın ürettiği örnekler birbirlerinden ayrılmış ve bunlar tek tek eskizler haline getirilmiş ve eskizlerin bilgisayara aktarılması ve kodlanması yapılmıştır.

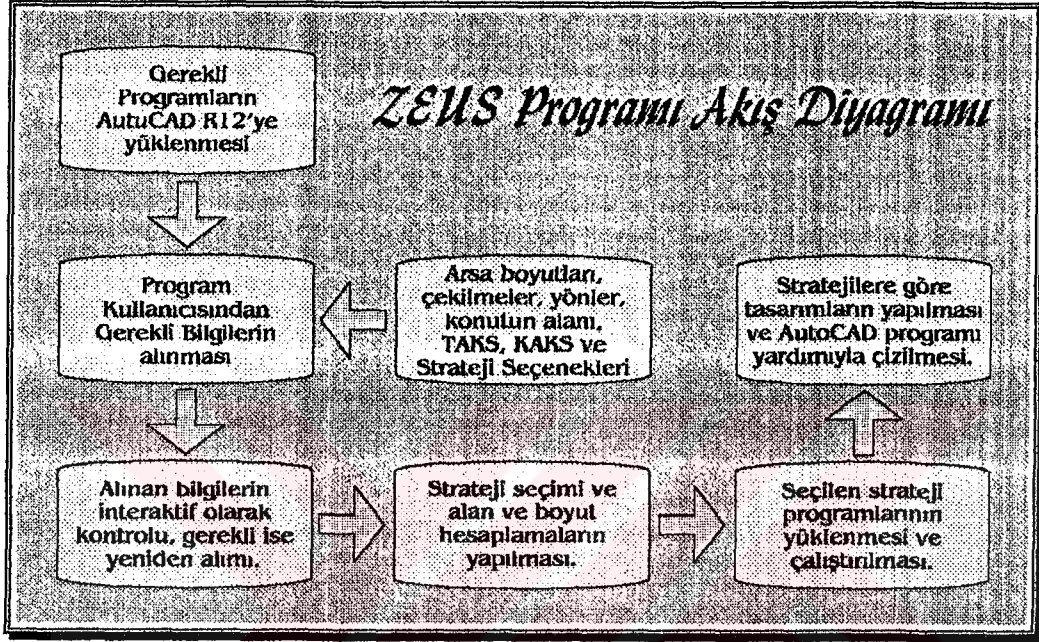
Video'ya kaydedilen oturum da UM'ın tasarım süreci ve tasarım sırasında kullandığı çözüm ve işlem sıraları saptanmıştır.

Daha sonra tasarım stratejilerini oluşturmak üzere, UM'ın oluşturduğu örneklerden söylediği, yazdığı ve/veya çizdiği kurallar çıkarılmıştır. Kurallar alan, oran, yönlendirme ve yerleştirme konularında olabilmektedir. (Mekanların yorumu farklı bir sistemin formundaki tasarım cümleleridir ^[11].)

Oluşturulan örneklere bağlı olarak çıkarılan kurallar eskizlere bağlı olarak düzenlenmiş ve her örneğe bağlı olarak, örnekleri oluştururken UM'ın kullandığı tasarım stratejileri çıkarılmıştır. Yani tasarımı yaparken UM'ın çizim işlemine başlama noktası ve takip ettiği yol, mekanları yerleştirme sırası, mekan çözümlerini yapma yaklaşımları, vb. strateji olarak belirlenmektedir. İki adet literatürden tasarım stratejisi, yedi adet de UM'ın tasarım stratejisi saptanmıştır.

komutlarının kullanılması yapılırken ayrıca hesaplama ve şart cümlecikleri de yazılabilmektedir.

Ek C’de bu dillerde kullanılan komutların açıklamaları bulunmaktadır.



Şekil 3.9. Zeus Programı Akış Diyagramı

ZEUS paket programı, "Mimari Tasarlama Eğitiminde Yazlık Konut Tasarımı amacıyla kullanılabilecek bir Uzman Sistem" tezinin sonucunda ortaya çıkarılan programdır. Program, tezin içinde çıkarılan bilgilere dayanarak bir US oluşturmaktadır.

Program, tezin ilk aşamalarında karar verildiği gibi AutoCAD R12 paket programı ve onun üzerinden çalışan AutoLISP, Diesel ve DCL programlama dilleri yardımıyla yazılmıştır. Oluşturulan paket program AutoCAD R12 for DOS ve AutoCAD R12 for WINDOWS programları üzerinden çalışmaktadır.

ZEUS, programı LSP, HLP, SLD ve DCL uzantılı dosyalar içinde programın kabul edeceği formatlarda yazılmış ve program, uygulama bölümünde işlem aşamalarına ayrılarak anlatılmıştır.

ZEUS programının girdileri iki tipe ayrılmıştır. Bunlar

1) Sabit tutulan değerler,

2) Kullanıcıdan alınan değerlerdir (Burada bahsedilen kullanıcı Mimarlık Öğrencisidir=MÖ).

MÖ, ZEUS'un çalışabilmesi için gerekli dosyaları HardDisk'ine yükeledikten sonra AutoCAD paket programını çalıştırır.

ZEUS'u yüklemek için MÖ HardDisk'ine yüklediği dosyalar arasından THESIS.DWG'i AutoCAD'in 'Open' komutuyla açar.

Dosyanın açılmasıyla ZEUS'un Menü'süde yüklenmiş olur. Menüden ilk önce ZEUS programını yüklemelidir.

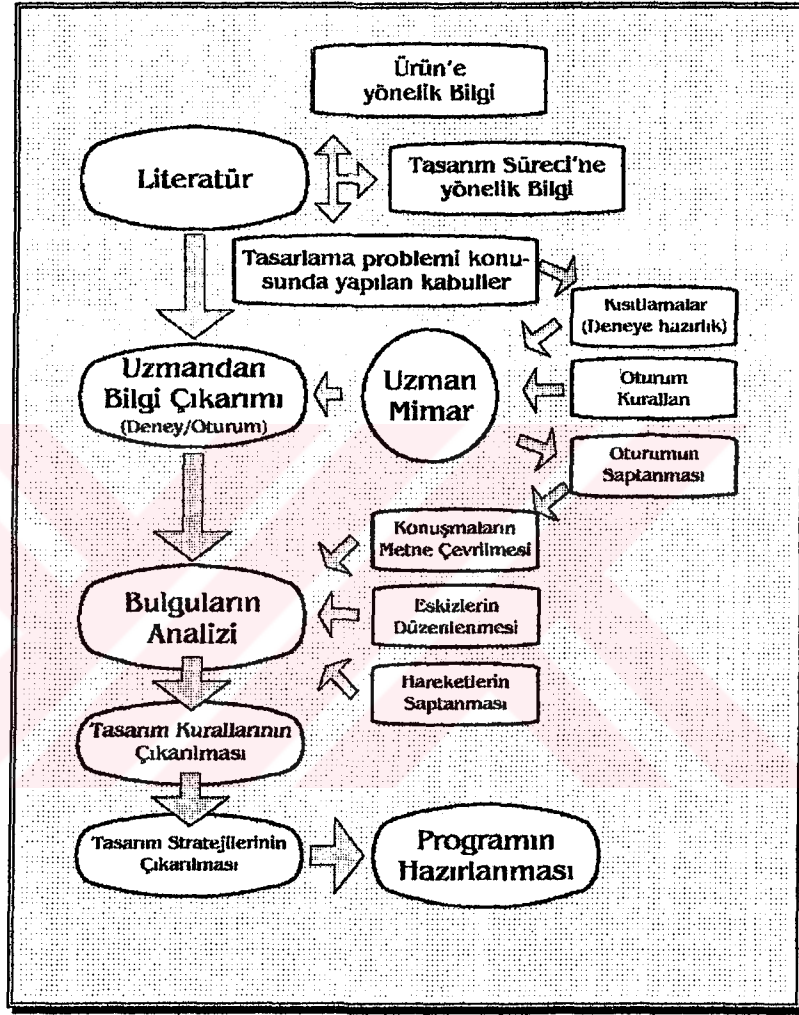
ZEUS paket programı yüklendikten sonra ZEUS'un ana giriş sorgu kutusu çıkar. MÖ burada giriş verilerini istediği gibi atadıktan sonra, program bu bilgileri analiz eder. Eğer kabullere veya formüllere uymayan veriler varsa, bu verilerin tekrardan girilmesini ister.

Program gerekli verileri aldıktan sonra MÖ'den hangi tasarım stratejilerinin verilen veriler ışığında değerlendirilmesinin istendiğini sorar. Bu aşamada MÖ'si en az bir tasarım stratejisi seçmek zorundadır.

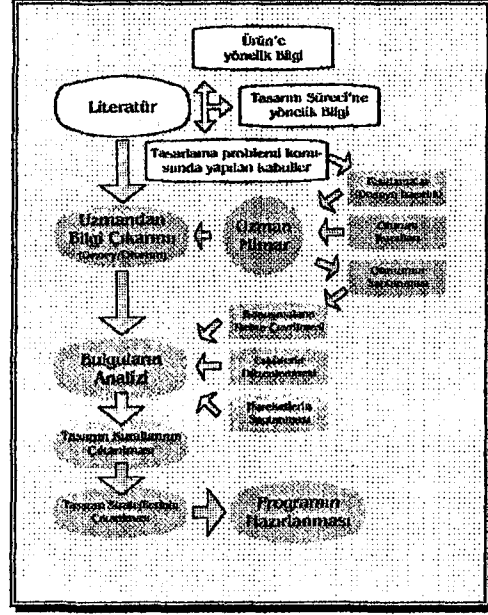
Daha sonra seçilen tasarım stratejilerine göre ZEUS örnek konutların yaratılmasına başlar.

İstenen tasarım stratejileri aynı çizim içerisinde çizilir. Bunun amacı MÖ'nin istediği örneklerin tümünü aynı çizim üzerinde görebilmesi ve UM'ın verilen şartlardaki bir yazlık konutun tasarımına nasıl başladığını daha rahat algılayabilmesini sağlamaktır.

BÖLÜM 4. UYGULAMA



Şekil 4.1. Uygulama



Şekil 4.2. Literatür Bilgileri

Uygulama yapılırken literatürden yararlanılarak çıkarılan bilgilere dayanan kısıtlamalar, UM'a problem açıklanırken verilmek üzere hazırlanmıştır. Bu verilen kısıtlamalar aynı zamanda programın kullanıcısı olarak MÖ'lerine etkileşimli olarak sorulan ve programa girdi oluşturacak olan bilgilerle aynı niteliktedir.

ZEUS, tasarlama sürecinde mimarlık öğrencilerine yardımcı olmayı hedefleyen bir US'dır. Amacına ulaşabilmek için tasarım probleminde, tasarım sürecine, araçlara ve çevre koşullarına ait bilgilere ihtiyaç duyulmaktadır.

Ek A'de T.C. Başbakanlık Toplu Konut İdaresi'nin gerçekleştirdiği konutların programları tezin konusuyla ilgili olan konutlardan alınarak ortalama değerleri çıkarılmış, konut alan oranları ve boyutlarıyla ilgili bir tablo hazırlanmıştır. Bu tablo, literatür araştırması sırasında çıkarılan tasarlama stratejilerinde kullanılmıştır.

Ek B'de İ.T.Ü. Mimarlık Fakültesi, Mimarlık Bölümün'de Bina Bilgisi Dersi kapsamında öğrencilere dağıtılan mekan yönlendirme şeması vardır. Bu şema, literatürden çıkarılan tasarlama şemalarında

ve UM'ın tasarlamada yönü belirtilmeyen mekanların yönlendirilmesinde kullanılmıştır.

Mimara verilen kısıtlamalar tezin hedefine ulaşılması için yapılmıştır. Aksi halde örneğin mekan sayılarının artması veya arsa bilgilerinin sınırlanmaması halinde oluşacak olan ZEUS içinde bulunması gereken kural sayısı artacak ve programlama zorlukları bu aşamada çok fazla olacaktır. Yukarıda sayılan nedenler doğrultusunda, aşağıda dökümü verilen kısıtlamalar ve/veya kabuller yapılmıştır.

UM'a bu kısıtlamalar içinde kalan örnek arsa ve kullanıcı istekleri verilmiştir. Ancak, verilen bazı kısıtlamalar ve kullanıcı istekleri daha sonra UM tarafından tasarımı sınırlayıcı olduğu belirtilecek örnek oluşturması açısından kaldırılmıştır. UM tarafından kaldırılan kısıtlamalar veya kullanıcı istekleri UM'ın tasarım stratejileri incelenirken belirtilmiştir.

Hedeflenen "Mimari Tasarlama Eğitiminde Yazlık Konut Tasarımı Amacıyla Kullanılabilecek Bir Uzman Sistem" paket programında aşağıdaki şartları sabit tutmaktadır. Bunun yanısıra ZEUS programı çalışması sırasında bazı bilgileri de kullanıcısı olan mimarlık öğrencilerinden alacaktır. Araştırma sırasında kullanıcıların belirlemesi istenecek değişken değerleri aralığında kalmak koşuluyla, tasarım problemini benzeştirmek için UM'dan, kabul edilen bir değer grubunda tasarım yapması istenmiştir.

Yazlık Konutun sabit kabul edilen özellikleri ve UM'a verilen bilgiler aşağıda sırayla belirtilmektedir.

1. Arsa İle İlgili Kabuller

o Arsanın yeri: Arsa İstanbul yakınlarında deniz kıyısına yakın fakat deniz kenarında değildir.

o Arsa düz ve dikdörtgendir.

- o Yazlık Konut müstakil bir yapı olacaktır.
- o Arsanın Alanı: 442 m².dir. Eni 17m. ve Boyu 26 m. dir.
- o Yoldan çekilme 5m., Önden çekilme 10 m. ve Yanlardan çekilme mesafesi 3 m. olarak kabul edilmiştir.
- o Konutun net alanı minimum 35 m². olabilir.
- o Arsanın KAKS' 1, TAKS' ise 0.20 dır.
- o Arsanın Kuzey Yönü 45° Hakim Rüzgar Yönü 135°'e Manzara Aksı 270°'dir. (Manzara Aksı; Manzara Yayının açısı ortaydır.)
- o Arsa üzerinde yapılaşmayı etkileyebilecek düzeyde bitki örtüsü yoktur.
- o Su, gaz ve kanalizasyon bağlantıları yol cephesinden sağlanmaktadır.

2. Yazlık Konut ve Kullanıcılarının Sabit Kabul Edilen Özellikleri veya İstekleri

- o Konut tek katlıdır.
- o Konut 4 mekandan oluşmaktadır. Bunlar 1 salon, 1 yatak odası, 1 banyo ve 1 mutfaktır.
- o Salon oturma, yemek ve dinlenme eylemlerine izin vermelidir.
- o Yatak Odası fazla güneş almamalıdır.
- o Banyo hem tuvalet hemde yıkanma ihtiyaçlarına cevap vermelidir.
- o Mutfak salona veya diğer bir mekana açık olmamalıdır. Fakat girişi salon veya varsa giriş holünden olabilir.

(Not: Bu istek UM tarafından örnek oluşturması amacıyla kaldırılmıştır.)

- o Konutta koridor veya benzer mekanlarla yer kaybı istenmemektedir.
- o Konutun çatı çözümü mekan yerleşimini etkilemeyecektir.
- o Konut 2 kişilik bir aile için tasarlanacaktır.
- o Yatıya misafir gelmeyeceği kabul edilmektedir.
- o Ailenin çocukları yoktur.

o Aile bireylerinin ortalama yaşı 25'dir.

Yukarıda belirtilen kriterler sabit tutulduğunda, kanunlar, yönetmelikler, standartlar veya imar kanunları değişmediği sürece programın algoritmasının yenilenmesi gerekmeyecektir. Ancak yeni tasarım stratejileri eklenebilir veya mevcutlar geliştirilebilir.

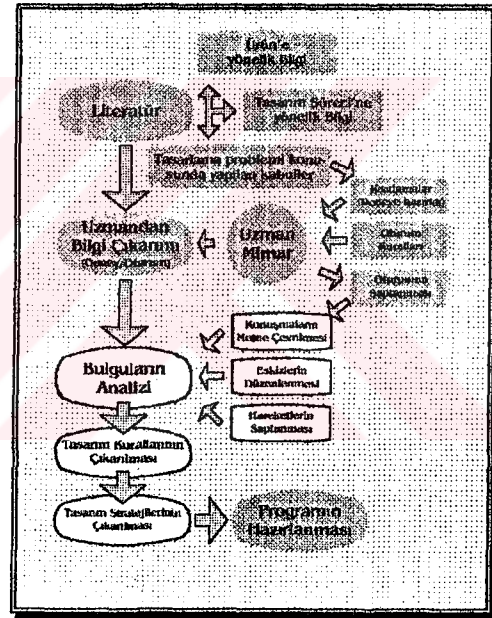
3. ZEUS programın MÖ olan kullanıcılarından alacağı bilgiler:

- o Arsanın Alanı,
- o Arsanın eni ve boyu,
- o KAKS (Konut tek katlı olduğundan tasarımı etkilemeyecektir.),
- o TAKS,
- o Arazideki önden ve yanlardan çekilmeler,
- o Kuzey Yönü,
- o Hakim Rüzgar Yönü,
- o Manzara Yönü.

düşüncelerle yaptığını açıklaması istenmiştir. Bu nedenle; oturumda 'Sesli Düşünme' yöntemi UM'dan bilgi alma yöntemi olarak benimsenmiştir.

Oturum sırasında yapılan tasarım eylemi, konuşma ve hareketler, yaptığı eskizler videoya kaydedilmiştir. Ayrıca oturum sırasında UM'ın yaptığı eskizler alınmıştır. Oturumun kuralları, bunların saptanma şekli ve Ek D'deki bilgilerin verilmesinden sonra UM tasarıma başlamıştır.

4.2. Bulgular ve Bulguların Analizi (Protokol Analizi)



Şekil 4.4. Bulguların Analizi

Protokol Analizinin yapılabilmesi için oturum metninin oluşturulması gerekmektedir. Bunun için görüntü, ses ve çizilen eskizler gibi oturum tutanaklarının kağıt üzerine dökülmesi gerekmektedir. Bu işlem için konuşmalar metne çevrilmiş, eskizler düzenlenmiş ve videodaki görüntülerden tasarımın mekanlarının hangi sıra ile birleştirildiği ortaya çıkarılmıştır. Bu metin Ek E'de bulunabilir.

Oturum da geçen konuşmaların tam metni, yapılan eskizlerin konuşmalara göre metne yerleştirilmiş ve gereksiz çizgilerden ayıklanmış hali ve tasarımda hangi mekanların sırasıyla çizildiği Ek E'de yer almaktadır.

Tasarım Kurallarının çıkarılması işlemine Ek E'deki metnin incelenmesiyle başlanmıştır. Metin içinde bulunan her işlem ayrı bir satır haline çevrilmiştir. İşlem sahibinin her cümlecğini takiben içerdiği kural veya tercih olarak ayrılmıştır. Bunlar daha sonra hangi örneğin içinde yer aldığına ve içerdiği bilgi tipine göre ayrılmıştır.

Aşağıdaki şekillerden eskizlerin ayrı ayrı analizi yer almaktadır. Bu listedeki her cümlecik için yapılan işlem bir örnek cümle için incelenecek olursa;

UM'ın oturum sırasında kullandığı cümle:

“Belki yine köşesel bir giriş, ana hacmi ısırın sert bir eleman,”

Bu arada UM, beş numaralı eskizdeki iki numaralı dikdörtgeni çizmektedir.

UM'ın bu cümlesinden şu kural eskiz 5 için ortaya çıkarılır.

“- Giriş, hakim rüzgara bakan A köşesinden dikdörtgen olarak yerleştirilir.”

Ve eskiz 5'in Tasarım Stratejisinde;

- Giriş köşeden ve Hakim Rüzgar yönünde olmalıdır.

kural olarak kullanılır.

Tablo 4.2.1.'de Oturum ardından çıkarılan kuralların benzerlerinin birleştirilerek oluşturduğu kurallar listesi bulunmaktadır.

Tablo 4.1. Mimari yapılan oturma da çıkarılan kuralların bazıları (Not: Kural tablolarında kullanılan kısaltmaların açıklamalarını Ek F'de bulabilirsiniz.)

UZ'ın Verdiği Kural	Yorumu
Eğer alan 45-60 ise merkezi form.	
Yatak Odası kuzeye bakar.	YOD → KA
Salon Manzaraya bakar.	SAL → MA
Mutfak merkezi form içinde.	
Tek katlı yapı.	
Giriş, Yola bakar ve manzaraya bakmaz.	GİR → YOL & -MA
AB'de ıslahacım kapalı mutfak. Mutfak ve banyo yanyana olur.	BAN & MUT → -MA
giriş (HA). Giriş köşeden olur.	GİR → HA & Köşeden
Eğer 45 m ² . ise, giriş YOD, ıslahacım arasında olabilir.	
Eğer verilen evin alanı 45 m ² ise Evin eni 7m. ve boyu 6.5m. olur.	Eğer 45m ² . ise, 7 x 6.5m olur.
Salon dışındaki alanlar için boy 2.80m. olur.	Eğer 7 x 6.5m.ise, MUT, BAN & YOD eni 2.80m.
Salonun eni 4m., alanı 28 m ² .	Eğer 7 x 6.5m.ise, SAL eni 4 m, alanı 28m ² .
Eğer MUT açık olursa, MUT önünde yemek yeme imkanı olur.	
Açık MUT'ta olabilir.	Eğer 45m ² . ise, MUT açık olabilir.
Açılabilir YOD olabilir.	Eğer 45m ² . ise, SAL açık olabilir.
Eğer sıra evler ise YOD Kuzeye veya Hakim Rüzgara bakabilir.	Eğer sıra ev ise, YOD → KA & MA
Eğer sıra ev değilse YOD kuzeye bakmalı. YOD; Kuzey Doğu, Kuzey Batı arasında da olabilir.	YOD → KA; YOD → (KA -45°) - (KA +45°)
Eğer 2 katlı olursa, YOD yukarı çıkabilir.	
Açık mekanlı ev düşüncesi ile tasarıma başlandı.	
B köşesinde yan açık mutfak olabilir.	MUT → KA
Eğer alan 45 m ² . gibi küçük olursa yüzen elemanlarla mekanı serbest ve rahat bırakılmalıdır.	Eğer 45 m ² . ise, açık mekanlı olmalı.
Eğer alan 60m ² .nin altında ise açık MUT.	Eğer <60 m ² . ise, açık MUT
Eğer program iki kişilik aile içinse, YOD açık mekan olabilir.	
Yatak Odasına minimum alan ayrılmalı.	Minimum alan YOD
'ABC' ÜG; SAL; 'C' köşesi dik açı	
Eğer arsada yan çekilmeler 3m. ise, ÜG evin eni 11 olabilir.	
ÜG'de taban, h'ye eşit alırsak taban 9 oluyor.	'ÜG'de [AB]=h

Tablo 4.1. 'in devamı

Eğer 'ABC' ÜG' ninde alan $45m^2$. ise $h=taban=9.5$ m.	Eğer 'ABC' $45m^2$. ise, $h= AB =9.5m$.
Eğer yarlarda boşluk kalırsa, sol veya sağa bina çekilebilir.	
Eğer arsada manzara doğrultusunda çekilmelerden boşluk kalırsa, boşluk manzara yönüne verilmeli.	
Evin oturumunu yarlardan 5'te 2 veya 5'te 3 yaparak yerleştirmek bahçe taranımı için daha iyi.	
GİR ÜG'de ortada olmamalı.	
Eğer alan $60-70m^2$, olursa YOD ana yapıdan ayrılmalı.	
HA'da GİR, KA'da YOD, BAN ve MUT -MA'da	
MUT ve BAN yine 'AB' ortasında. GİR yine HA'da	
MUT; $1.2m \times 2m$, BAN; $2m \times 2m$, YOD; $3m \times 3.2m$, SAL $3.5m. \times 10m$.	
Ev alanı sınırlı olduğu için ana form ÜG veya DD olmalı.	
YOD, salondan direkt görünmez.	
Yuvarlak YOD nişi için 2.80 m. veya 3.00 çap yeterli.	
YOD'un salonu bakan taralı duvar ile kapalı, diğer yönleri açık olabilir.	
YOD'ı sirkülasyondan çıkarılmalı.	

4.3. Tasarım Stratejileri (Bulgular)

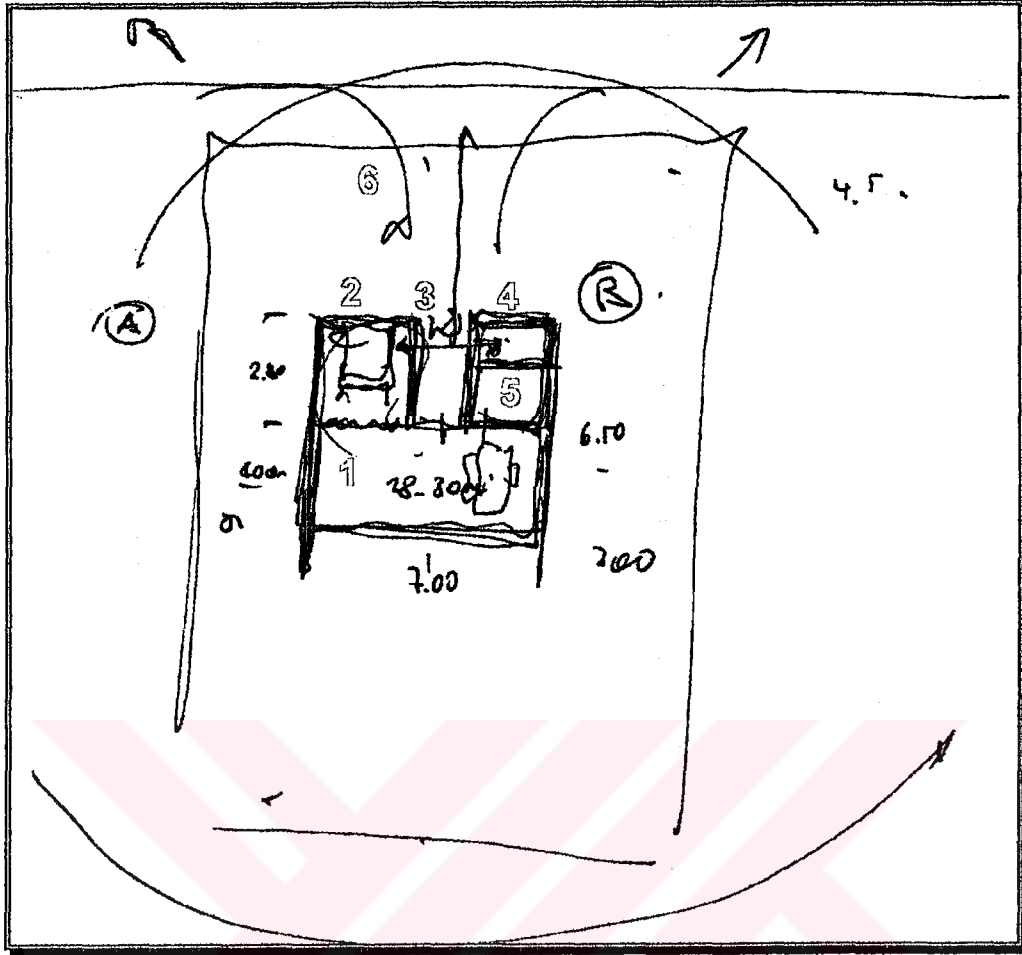
UM'la yapılan görüşme sırasında, önceden belirlenen sınırlayıcı kullanıcı ihtiyaçları, boyutlar ve kısıtlamalardan hareket edilmiştir. Bundan sonra biçimsel kısıtlamalar kaldırılmıştır. Bu düzenlemenin yapılış nedeni mimarla yapılan görüşme sırasında ortaya çıkan form ve biçim farklılıklarının da teze dahil edilmesidir. Değiştirilen biçimsel kısıtlamaların bazıları UM tarafından oturum sırasında kaldırılmış, sonuçta oluşan şartların incelenmesi sonucunda bazı kısıtlamaların kaldırılmasına karar verilmiştir. Ayrıca, tezin dahilinde göz önünde tutulmayacak olan *çatı formu* gibi yapı öğelerinin incelenmesine de karar verilmiştir. Bu tip tasarım elemanlarının da incelenmesinin, daha sonra ki çalışmalarda da kullanılabileceği ve tezin kapasitesinin genişletilebileceği göz önüne alınarak yapılmıştır.

Bu yüzden ařağıdaki tasarım stratejilerin de tasarım kurallarında bazı farklılıklar olmaktadır. UM'dan bilgi alımı sırasında stratejiler de farklılıklar yapılmıř ve stratejilerden sınırlayıcı olan bazıları kaldırılmıřtır. Ařağıdaki maddelerde sınırlayıcı kurallar kaldırıldıkça ve/veya deęiřtirildikçe belirtilecektir.

Tasarım stratejilerinin çıkarılması için, her eskizin metni ierisinden eskizle ilgili kurallar çıkarılmıřtır. Ayrıca programın kullanması için eskizlerden oranlar veya UM'ın verdięi ölçülerle mekan alanları ve boyutları hesaplanmıřtır. Bu kuralların incelenmesi, ortaya çıkarılan tasarım stratejileri ve oranlar ařağıda tek tek anlatılmıř ve bulguların analizi detaylı olarak sunulmuřtur.

Eskiz 1.

Oturumda mimarın ilk yaptıęı eskizler ařağıda görölmektedir. İlk iki eskizle ilgili UM'ın söyledikleri, çizdikleri ve kurallar Tablo 4.3.1.'de incelenmektedir. İlk iki eskizde UM, eskizler arasında sürekli olarak geiřler yaptıęı için ekizlerin ilk incelemeleri beraber yapılmıřtır. Tasarım stratejileri ise ayrı ayrı ele alınmıřtır.



Şekil 4.5. Eskiz 1

Tablo 4.2. Eskiz 1 ve 2'nin Kural Dökümü.

Eskiz 1 ve 2 (Bu örnekler beraber düşünüldüğü için kurallar beraber incelenmiştir.)				
	Uzman'ın söyledikleri	Bilgi Tipi	Bilgi açılımı	Uzman çizimi ve kuralı
1	Konut basit bir konut planı olarak merkezi bir form teşekkül ediyor.	UB	Eğer alan 45-60 ise konut merkezi form olmalıdır.	
2	Yatak odasının fazla güneş almaması isteniyor.	UB	Yatak Odası güneye bakmalıdır. (-KA)	
3	Manzaraya açık bir salon,	UB	Salon Manzaraya bakmalıdır. (MA)	Eskiz 1 'EHCD' DD; SAL. 'AIFE' DD; YOD. 'IKGE' DD; GİR. 'KBHG' DD; BAN ve MUT. 'KBL' DD; BAN. 'JLHG' DD; MUT.

Tablo 4.2.'nin devamı

4	kapalı bir mutfak isteniyor	Uİ	Mutfak merkezi form içinde kalmalıdır.	
5	tek katlı olması şart koşuluyor.	Uİ	Tek katlı yapı	
6	öne gelen salon	UB	Salon manzaraya bakmalıdır.	Eskiz 2. 'EHCD' DD; SAL.
7	bir taraftan girdiğiniz takdirde	UB	Giriş, yola bakmalı ve manzaraya bakmamalıdır.	'AFE' DD; GİR.
8	bu tarafa göre inkişaf etme imkanınız	UB	'IF'den salon dışındaki mekanlar	'IF'den 'FH' doğrultusunda MUT, BAN ve YOD.
9	arkada ıslakhacimler kapalı mutfak	UB	Mutfak ve banyo yanyana olmalı ve Manzara açısına bakmamalıdır.	'ILNM' DD; BAN, 'MNGF' DD; MUT.
10	Yatak odası buraya geliyor.	UB	Yatak odası Kuzeye bakmalıdır.	'LBHG' DD; YOD.
11	Giriş yanda	UB	GİR HR'de olmalıdır.	'AFE' DD; GİR.
12	giriş ortaya gelirse bu ihtimal ortaya çıkıyor.	UB	veya Giriş AB'nin ortasında olur, YOD, ıslakhacimler arasında olmalıdır.	Eskiz 1. 'IKGH' DD; GİR.
	Konut 45 m ² ise, 7 x 6.5 olduğu takdirde	Uİ	Eğer verilen konutun alanı 45 m ² ise Konutun eni 7m. ve boyu 6.5m. olmalıdır.	'ABCD' DD 7m. x 6.5 m.
13	ıslakhacimler ve yatak odası için 2.75-2.80 (m) bir aks alırsak,	AH	Salon dışındaki alanlar için boy 2.80m. olmalıdır.	'ABHE' DD eni 2.80m.
14	geriye 4 duvarlar hariç kalır. 28-30 m ² . bir oturma odası olur.	AH	Salonun eni 4m., alanı 28 m ² . olmalıdır.	'EHCD' DD eni 4, boyu 7. Alanı 28m ² .
15	Buraya mutfak açılır önünde aynı yemek yeme imkanı var.	UB	Eğer MUT açık olursa, MUT önünde yemek yeme imkanı olur.	Açık MUT'ta olabilir.
16	Yatak odası gerektiğinde açılabilmesi kullanma alanını genişletebilir.	UB	Eğer YOD açılabilirse, SAL genişler.	Açılabilir YOD olabilir.
17	Girişin ortadan olması banyo ve tuvaletin karşıya geçmesi konutun küçük olmasıyla büyük bir problem doğurmuyor. Bunlara ek bunların ekonomisi ve bağlantısı rahatlıyor.	UB	Ortadan giriş olabilir. YOD - BAN geçişi sorun yaratmaz.	Giriş YOD ve BAN, MUT arasında olabilir.

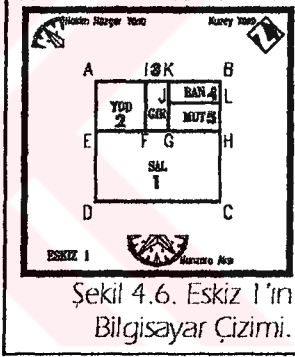
Tablo 4.2.'nin devamı

19	Eğer sıradan arsalar yanyana ise bu yatak odasının burada veya diğer yönde olmasının bir mahsuru yok.	UB	Eğer sıra evler ise YOD Kuzeye veya Hakim Rüzgara bakabilir.	Eğer sıra konut ise, KA veya MA olabilir.
	En basit şekliyle bu.		Eskiz 1 ve 2'nin sonu.	
20	Belki burada baş aşağı gelebilir. buraya gelebilir.	UB	YOD kuzeye bakmalı.	YOD; Kuzey Doğu, Kuzey Batı arasından olmalı.
21	Eğer Katlı olsaydı başka imkanlar çıkabilirdi. belki o zaman yatak odası yukarı alanırdı.	UB	Eğer 2 katlı olursa, YOD yukarı çıkmalıdır.	Eğer 2 kat olursa Yatak odası yukarı çıkar.

M.1. Eskiz 1'in Tasarım Stratejisi

Tablo 4.3. Eskiz 1'in Tasarım Stratejisi

- Yatak Odası kuzeye bakmamalıdır. Hakim Rüzgara bakabilir.
- Salon Manzaraya bakmalıdır.
- Mutfak merkezi form içinde kalmalıdır.
- Mutfak Hakim Rüzgara bakmamalıdır.
- Salon manzaraya bakmalıdır.
- Giriş, yola bakmalı ve manzaraya bakmamalıdır.
- AB'de islahhacım kapalı mutfak. Mutfak ve banyo yanyana olmalıdır.
- Giriş A köşesinden olmalı ve HR'de olmalıdır.
- Eğer verilen konutun alanı 45 m^2 ise Konutun eni 7m. ve boyu 6.5m. olur.
- Salon dışındaki alanlar için boy 2.80m. olur.
- Salonun eni 4m., alanı 28 m^2 olmalıdır.
- Ortadan giriş olabilir. Yatak Odası, Banyo geçişi sorun yaratmaz.
- Mekanların yerleştirilme sırası; Salon, Yatak Odası, Giriş, Banyo ve Mutfak'tır.



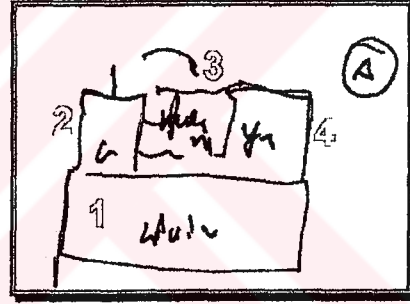
M.1. Eskiiz 1. Tasarım Stratejisinde US kullanacağı alan ve uzunluk oranları (Ölçüler Mimar tarafından verilmiştir.)

Tablo 4.4. Eskiiz 1'in Alan ve Mekan Boyutları.

Mekan No	1	2	3	4	5
Adı	Salon	Y. Odası	Giriş	Banyo	Mutfak
Eni (m)	4.00	2.80	2.80	1.20	1.60
Boy (m)	7.00	2.80	1.30	2.90	2.90
Alanı (m ²)	28.00	7.00	3.60	3.48	4.64
Alan Oranı	59.93	14.98	7.70	7.44	9.93

Evin Net Alanı 46.72 m².dir.

Eskiiz 2.

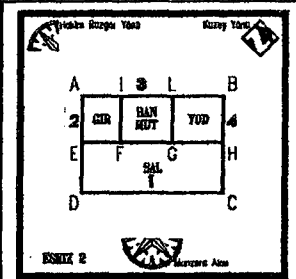


Şekil 4.7. Eskiiz 2

M-2. Eskiiz 2'nin Tasarım Stratejisi kuralları

Tablo 4.5. Eskiiz 2'nin Tasarım Stratejisi.

- Yatak Odası güneye bakmalıdır.
- Salon Manzaraya bakmalıdır.
- Mutfak merkezi form içinde kalmalıdır.
- Salon manzaraya bakmalıdır.
- Giriş, yola bakmalı ve manzaraya bakmamalıdır.
- Mutfak Hakim Rüzgara bakmamalıdır.
- AB'de ıslakhacim kapalı mutfak. Mutfak ve banyo yanyana olmalıdır.



Şekil 4.8. Eskiiz 2'nin Bilgisayar Çizimi.

Tablo 4.5.'in devamı

- Giriş A köşesinden olmalı ve Hakim Rüzgar'a bakmalıdır.
- Eğer verilen konutun alanı 45 m^2 ise Konutun eni 7m. ve boyu 6.5m. olur.
- Salon dışındaki alanlar için boy 2.80m. olur.
- Salonun eni 4m., alanı 28 m^2 olmalıdır.
- Mekanların yerleştirilme sırası; Salon, Giriş, Banyo, Mutfak ve Yatak Odası'dır.

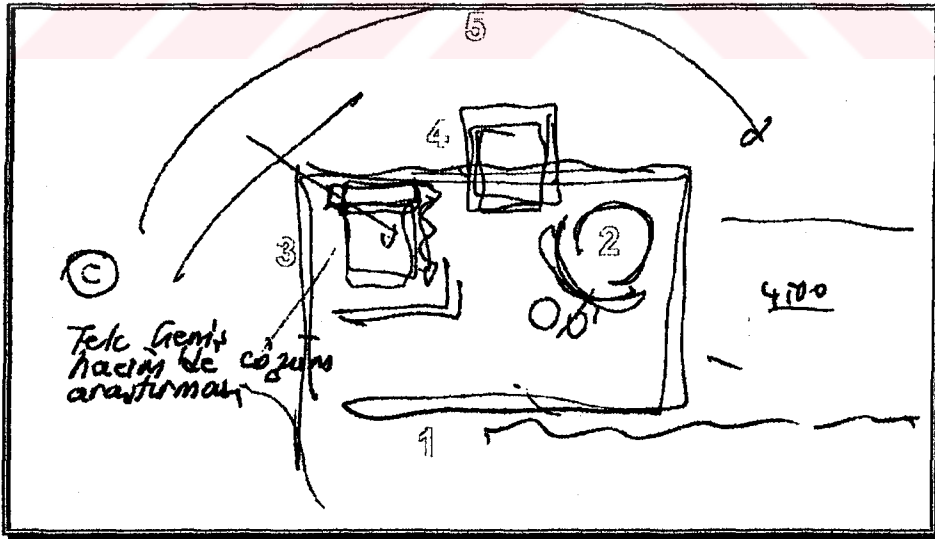
M.2. Eskiz 2'de US'in kullanacağı alan ve uzunluk oranları

Tablo 4.6. Eskiz 2'nin Alan ve Mekan Boyutları.

Mekan No	1	2	3	4
Adı	Salon	Giriş	Banyo / Mutfak	Y. Odası
Eni (m)	8.03	2.19	2.92	2.92
Boy (m)	2.92	2.62	2.62	2.62
Alanı (m ²)	23.65	5.73	7.65	7.65
Alan Oranı	52.93	12.82	17.12	17.12

Evin Net Alanı 44.68 m^2 dir.

Eksiz 4.



Şekil 4.9. Eskiz 4

Tablo 4.7. Eskiz 4'ün Kural Dökümü.

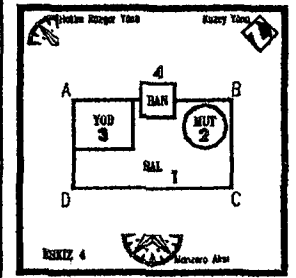
Eskiz 4 (Not: Eskiz 3 konu dışında olduğu için kapsama alınmamıştır.)				
	Uzman'ın söyledikleri	Bilgi Tipi	Bilgi açılımı	Uzman çizimi ve kuralı
1	Şimdi bütün bu klasik formlardan kurtularak, başka araştırmalara gelindiği takdirde,	UB	Açık mekanlı konut düşüncesi ile tasarıma başlanmıştır.	'ABCD' DD konutun sınırları olarak çizildi.
2	mutfağın kapalı olması isteniyor eğer yarı açık olabilseydi.	UB	Ana mekan içinde 2 nolu daire mutfak olarak çizilmiştir.	B köşesinde yarı açık mutfak olabilir.
3	O zaman tek hacim içerisinde, 45 m ² yine bunları kapsar ve tavana değmeyen yüzer elemanlarla bu üç boyutu tutardı ve o zaman ışıktaki yardımcı olurdu 4. boyuttada büyürdü ve yayılırdı etrafa.	UB	Eğer alan 45 m ² . gibi küçük olursa yüzer elemanlarla mekanı serbest ve rahat bırakılmalıdır.	
4	O zaman bir çerçeve kısmen transparan, kısmen dış iç entegrasyonuna imkan veren.	UB	Eğer yüzer ve/veya geçici elemanlarla mekan yaratılıyorsa saydam dış duvarlarla mekan dışarıya açılmalıdır.	
5	Yine giriş belki köşesel bir giriş her şey olabilir.	UB	Giriş HA'dan veya ortadan ('AB'nin ortasından) olmalıdır.	'A' veya 'AB' ortasından 4 DD; GİR.
6	Tek hacim içerisinde ve mutfak servis imkanı bankoda veya tekrar masada,	UB	Eğer açık mekan ise mutfak açık olmalı, banko veya masa yemek için kullanılabilir.	Eğer alan 60m ² .nin altında ise açık MUT.
7	yatak icabında açılabilir, kaldırılabilir bir yatak maden ki iki kişiler çocuksuz,	UB	Eğer program iki kişilik aile içinse, YOD açık mekan olabilir.	Eğer iki kişi içinse YOD açık olabilir.
8	Mümkün mertebe yaşama sahasını günlük ihtiyaçları rahatlatma yönüne gidilebilirdi.	UB	Eğer alan küçükse, ana mekan içinde alanlar serbest olmalıdır.	

M-3. Eskiz 4'de UM'ın kullandığı tasarım kurallarının cümle formuna çevrilmiş hali aşağıda bulunmaktadır.

Tasarım kurallarından mutfakın kapalı olmasının UM'la yapılan konuşmalar sonucu kaldırılmasına görüşme sırasında karar verilmiştir. Ayrıca tasarımda kullanılacak form dikdörtgen olarak sınırlandırılmıştı, bununla kaldırılmasına karar verilmiştir.

Tablo 4.8. Eski 4'ün Tasarım Stratejisi.

- Yatak Odası kuzeye bakmamalıdır. Hakim Rüzgara bakabilir.
- Salon Manzaraya bakmalıdır.
- Ana mekan içinde 2 nolu daire mutfak olarak B köşesine çizilmelidir.
- Mutfak Kuzeye bakmalıdır.
- Giriş A köşesinden Hakim Rüzgar yönüne bakmalıdır.
- Mutfak Hakim Rüzgara bakmamalıdır.
- Mekanların yerleştirilme sırası; Salon, Mutfak, Yatak Odası, Banyo'dur.



Şekil 4.10. Eski 4'ün Bilgisayar Çizimi.

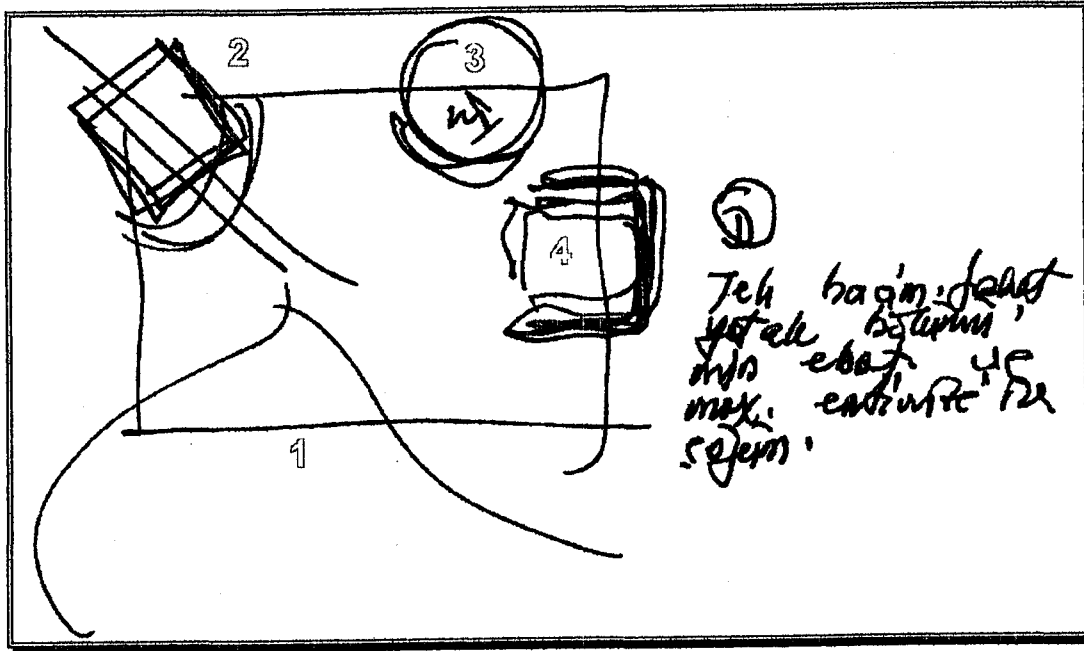
M.3. Eski 4.'den çıkarılan alan ve uzunluk oranları. Evin tek bir hacim içinde yüzücü elemanlarla mekanlara ayrılmasından dolayı Salon alanı yoktur.

Tablo 4.9. Eski 4'ün Alan ve Mekan Boyutları.

Mekan No	1	2	3	4
Adı	Ev	Mutfak	Y. Odası	Banyo
Eni (m)	5.03	Daire	2.79	1.90
Boy (m)	8.94	2.45 Çapı	3.35	1.90
Alanı (m ²)	44.96	4.67	9.34	3.61
Alan Oranı	100	10.38	20.77	8.02

Evin Net Alanı 44.96 m².dir.

Eskiz 5.



Şekil 4.11. Eskiz 5

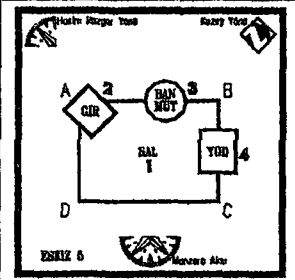
Tablo 4.10. Eskiz 5'in Kural Dökümü.

Eskiz 5				
	Uzman'ın söyledikleri	Bilgi Tipi	Bilgi açılımı	Uzman çizimi ve kuralı
1	Formu hep böyle çiziyorum ama böyle olması şart değil.	UB		'ABCD' DD: konutun formu.
2	Yine ıslakhacimler ve mutfak burada.	UB	BAN ve MUT; Manzara dışında (-MA) olmalıdır.	BAN ve MUT; 'AB' üzerinde
3	Belki ona yakın bir yerde yatak hacmi, mesela şöyle bir şeyde olabilir yatak kadar yatak hacmi olabilir.	UB	Yatak Odasına minimum alan ayrılmalıdır.	YOD; 4 DD 'BC' üzerinde
4	Belki yine köşesel bir giriş, ana hacmi ısıran sert bir eleman,	UB	Giriş HA'dan olmalıdır.	'A' köşesinde GİR.
5	Bu form yine dikdörtgen çünkü bu çok küçük bir bina olduğu için ama yine başka formlarda olabilir.	UB	Eğer alan küçükse YOD; DD formda olmalıdır.	

M-4. Eskiz 5. Tasarım Stratejileri

Tablo 4.11. Eskiz 5'in Tasarım Stratejisi.

- Salon manzaraya bakmalıdır.
- Banyo ve Mutfak; manzaraya bakmaz.
- Yatak Odasına minimum alan ayrılmalıdır.
- Giriş köşeden ve Hakim Rüzgar yönünde olmalıdır.
- Eğer alan küçükse Yatak Odası; dikdörtgen formda olmalıdır.
- Mekanların yerleştirilme sırası; Salon, Giriş, Banyo, Mutfak ve Yatak Odası'dır.



Şekil 4.12. Eskiz 5'in Bilgisayar Çizimi.

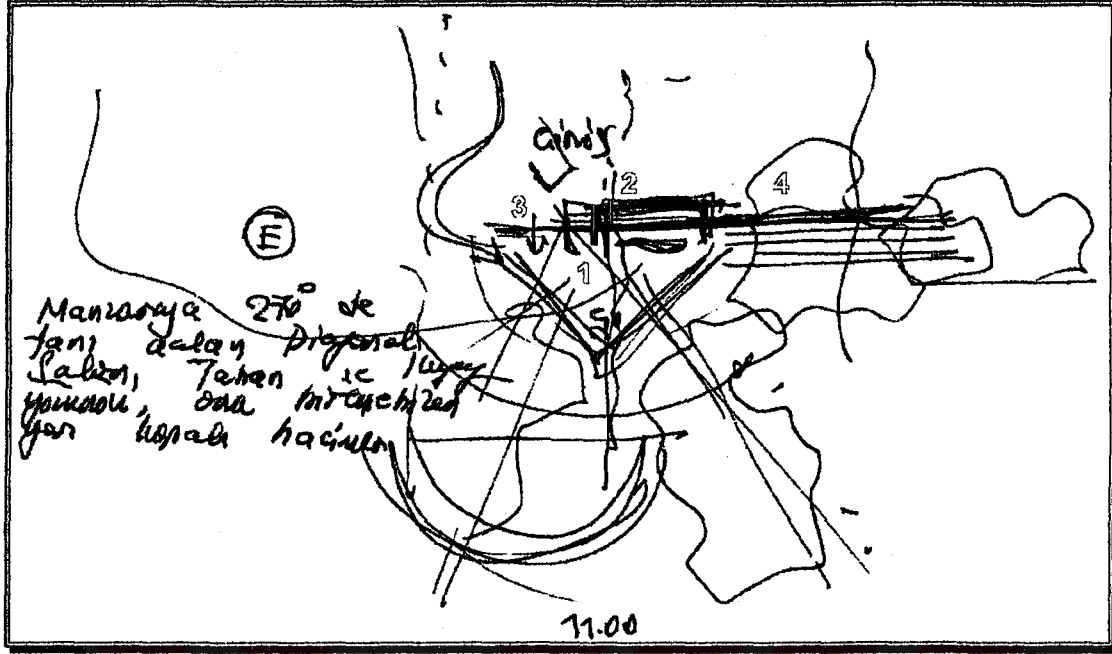
M.4. Eskiz 5. Alan ve uzunluk oranları. Evin tek bir hacim içinde yüzücü elemanlarla mekanlara ayrılmasından dolayı Salon alanı yoktur.

Tablo 4.12. Eskiz 5'in Alan ve Mekan Boyutları.

Mekan No	1	2	3	4
Adı	Ev	Giriş	Mutfak/Banyo	Y. Nişi
Eni (m)	7.88	2.18	Daire	2.06
Boy (m)	5.70	1.94	2.30 Çapı	2.42
Alanı (m ²)	44.91	4.22	4.15	4.98
Alan Oranı	100	9.39	9.24	11.08

Evin Net Alanı 44.91 m².dir.

Eskiz 8.



Şekil 4.13. Eskiz 8

Tablo 4.13. Eskiz 8'in Kural Dökümü.

Eskiz 8				
	Uzman'ın söyledikleri	Bilgi Tipi	Bilgi açılımı	Uzman çizimi ve kuralı
1	Burada manzara simetrik gibi bir yarım daire çiziyor, ama bu bir üçgenede dönüşebilir.	UT	Ana form ÜG içinde çözülmelidir.	'ABC' ÜG
2	Salon, mutfak veya yatak yine aynı mekan içerisinde	UB	SAL, MUT ve YOD 'AB' üzerinde olmalıdır.	'DEFG' DD'de 'A' dan 'B'ye MUT, BAN ve YOD.
3	giriş	UB	GİR HA'da olmalıdır.	GİR 'AB' üzerinde 'A'ya yakın
4	180 derecelik açılıma kuşak açan dik açılı bir salon,	UB	Salon Manzaraya bakmalıdır.	'ABC' ÜG; SAL 'C' köşesi dik açı
5	Yine üçer metre minimum fakat 3 metre çekince 11 metre kalıyor o fazlasıyla yeterli bu tür bir yerleşime,	AH	Eğer arsada yan çekilmeler 3m. ise, konutun eni 11m. olabilir.	
6	taban çarpı yüksekliğin yarısı 9 metre alsak yine 11 metre içinde kalıyor.	AH	ÜG'de taban, h'ye eşit alırsak taban 9 olmalıdır.	Eğer 'ABC' ÜG' ninde alan 45m ² . ise h=taban=9.5 m.

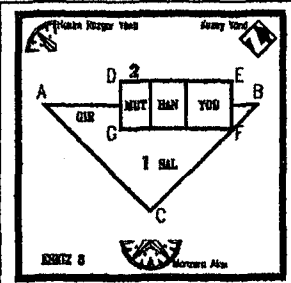
Tablo 4.13.'ün devamı

7	Hatta bir tarafa çekip bir tarafı bol bırakmak imkanı var	UT	Eğer yanlarda boşluk kalırsa, sol veya sağa bina çekilmelidir.	
8	birde mümkün mertebe geriye aldığımız için önde çok güzel bir açılım kalıyor.	UT	Eğer arsada manzara doğrultusunda çekilmelerden boşluk kalırsa, boşluk manzara yönüne verilmelidir.	
9	Bu bahçeleri formal yan formal düzenlemek imkanı veriyor	UB	Bahçeleri evi merkeze yerleştirmeyerek, yan formal yapılmalıdır.	
10	Proporsiyon olarak 5'te 2 5'te 3 gibi kalıyor bir yana çekersek,	UT	Konutun oturumunu yanlardan 5'te 2 veya 5'te 3 yaparak yerleştirilmelidir.	
11	hatta bir güzel imkanda bu servisin ve yatak odasının ayrı bahçeleri olabiliyor	UT	Konutun oturumunu yanlardan 5'te 2 veya 5'te 3 yaparak arka bahçeyi ayırmak gerekir.	
12	Duvar vesaireyi uzatarak bölmek imkanı doğuyor.	UT	Arka bahçe duvarla bölünebilir.	
13	girişi köşeden arıyoruz	LIB	GİR HA'da olmalıdır.	
14	Tabii yerindeki şartlarla bu diğer köşeye kayabilir.	UB	'AB' üzerindeki mekanlar 'BA' şeklinde dizilebilir.	
15	Ama böyle bir planda herhalde ortadan girmemeli.	UB	GİR ÜG'de ortada olmamalıdır.	

M-5. Eski 8. Tasarım Stratejileri

Tasarımda Kullanılacak ana form dikdörtgen olarak belirlenmişti, bu kuralın Uzman Mimar'la yapılan konuşmalar sırasında kaldırılmasına karar verilmiştir.

Tablo 4.14. Eski 8'in Tasarım Stratejisi.

- Ana form üçgen olmalıdır. - Salon, Mutfak ve Yatak Odası üçgenin hipotenüsü üzerinde yer almalıdır. - Üçgen Salonun dik açısı Manzaraya bakmalıdır. - Giriş Hakim Rüzgar yönünden olmalıdır. - Giriş Üçgen'de ortada olmamalıdır. - Eğer yanlarda çekilmelerden sonrada boşluk kalırsa yapı bir yöne çekilmelidir.	 Şekil 4.14. Eski 8'in Bilgisayar Çizimi.
---	---

Tablo 4.14. 'un devamı

- Eğer arsada manzara doğrultusunda çekilmelerden boşluk kalırsa, boşluk manzara yönüne verilmeli.
- Konutun oturumunu yanlardan 5'te 2 veya 5'te 3 yaparak yerleştirmek bahçe taranımı için daha iyi olmaktadır..
- Mekanların yerleştirilme sırası; Salon, Giriş, Banyo, Mutfak ve Yatak Odası'dır.

M.5. Eskiz 8. Alan ve uzunluk oranları. Evin tek bir hacim içinde yüzücü elemanlarla mekanlara ayrılmasından dolayı Salon alanı yoktur.

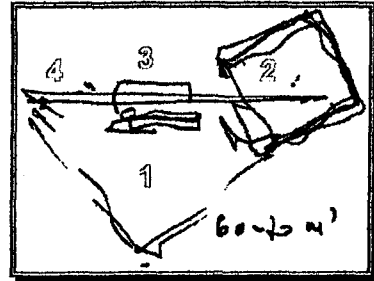
Tablo 4.15. Eskiz 8'in Alan ve Mekan Boyutları.

Mekan No	1	2	3	4
Adı	Salon	Banyo	Mutfak	Y. Odası
Eni (m)	Taban 12.00	2.00	1.70	2.60
Boy (m)	Yükseklik 6.00	2.68	2.68	2.68
Alanı (m ²)	33.74	5.36	4.55	6.96
Alan Oranı	66.65	10.58	8.98	13.74

Evin Net Alanı 50.62 m².dir.

Not: Salonun alanı üçgenin alanından Yatak Odası, Mutfak ve Banyo'nun alanlarının yarısının çıkarılmasıyla elde edilmiştir.

Eskiz 9



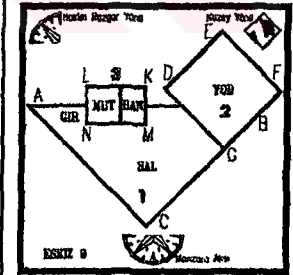
Şekil 4.15. Eskiz 9

Tablo 4.16. Eskiz 9'un Kural Dökümü.

Eskiz 9				
	Uzman'ın söyledikleri	Bilgi Tipi	Bilgi açılımı	Uzman çizimi ve kuralı
1	tek katlı olduğunda bu ebatlar 15 m ² . büyük olsa yavaş yavaş yatak odasını koparmaya gidebilirsin. Kendi içerisinde ayrı bir bölüm haline getirmeye başlayabilirsin.	UB	Eğer alan 60-70m ² , olursa YOD ana yapıdan ayrılmalıdır.	'ABC' ÜG'nine 'B' köşesinden 'DEFG' DD YOD olarak eklenir.
2	burada mutfak yapıp yine girişi buradan yapınca yine 60 70 m ² .de çok büyük bir faydası olmayacağı kanatındesim.	UB	MUT ve BAN 'AB' ortasında, GIR HA'da olmalıdır.	'LKMN' DD; MUT ve BAN. GIR 'A' köşesinde.
3	metrekare enteresan, kritik olduğu için, jeometrik form içerisinde halledebilirsek. Ne kadar temiz olursa o kadar üçgen, dikdörtgen vs. yani hatta yamuk bile olmamalı	UB	Konut alanı sınırlı olduğu için ana form ÜG veya DD olmalıdır.	

M.6. Eskiz 9. Tasarım Stratejisi

Tablo 4.17. Eskiz 9'un Tasarım Stratejisi.

- Ana form üçgen olmalıdır. - Eğer alan 60-70m², olursa Yatak Odası ana yapıdan ayrılmalıdır. - Salon ve Mutfak üçgenin hipotenüsü üzerinde yer almalıdır. - Üçgen Salonun dik açısı Manzaraya bakmalıdır. - Giriş Hakim Rüzgar yönünden olmalıdır. - Giriş Üçgeni'nde ortada olmamalıdır. - Eğer yanlarda çekilmelerden sonrada boşluk kalırsa yapı bir yöne çekilmelidir. - Eğer arsada manzara doğrultusunda çekilmelerden boşluk kalırsa, boşluk manzara yönüne verilmeli. - Mekanların yerleştirilme sırası; Salon, Yatak Odası, Banyo, Mutfak ve Giriştir.	 Şekil 4.16. Eskiz 9'un Bilgisayar Çizimi.
---	--

M.6. Eskiz 9.'un US'de kullanılacak alan ve uzunluk oranları.

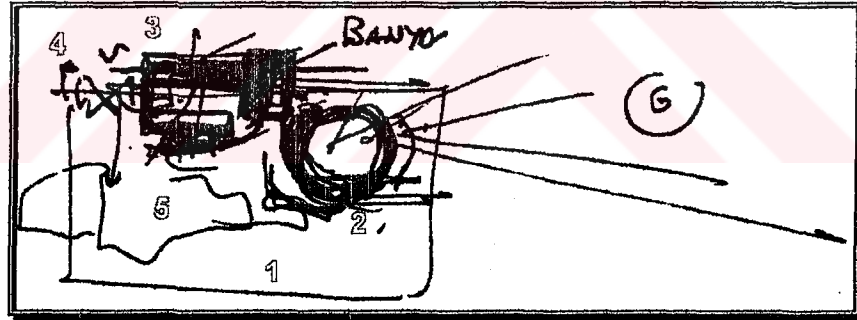
Tablo 4.18. Eskiz 9'un Alan ve Mekan Boyutları.

Mekan No	1	2	3	4
Adı	Salon	Y. Odası	Mutfak	Banyo
Eni (m)	Taban 13.50	4.55	1.9	1.40
Boyu (m)	Yükseklik 6.75	4.76	2.07	2.07
Alanı (m2)	38.82	21.65	3.93	2.89
Alan Oranı	57.69	32.17	5.84	4.29

Evin Net Alanı 67.29m².dır.

Not: Salonun alanı üçgenin alanından Mutfak ve Banyo'nun alanlarının yarısının Yatak Odasının alanın dörtte birinin çıkarılmasıyla elde edilmiştir.

Eskiz 19



Şekil 4.17. Eskiz 19

Tablo 4.19. Eskiz 9'un Kural Dökümü.

Eskiz 19 (Eskiz 9 ile 19 arasındaki eskizler konu dışında kalmaktadır.)				
	Uzman'ın söyledikleri	Bilgi Tipi	Bilgi açılımı	Uzman çizimi ve kuralı
1	Arsa düz ama, o kadar güzel bir mehtap da olur ki adam ayda bir iki gece görmek ister, bu durumda yatak veya mutfakın orayı yükseltmek gerekebilir.	UT	DD form içinde, YOD çatısı yükseltilebilir.	'ABCD' DD; KONUT,

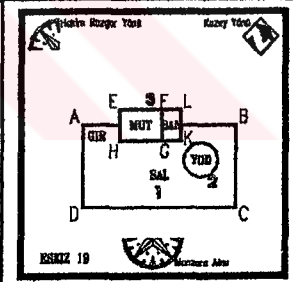
Tablo 4.19.'un devamı

2	Planda buradan alsak yatağı bu şekilde böyle halletsek, bunun üstüne gelecek bir eleman bunula beraber yuvarlak bir yatakda olabilir.	UT	YOD yuvarlak çatısı ve tabanı yüksek olmalıdır.	2 daire; YOD,
3	Başı bu tarafa kapalı. yandan girilen odadan bakıldığında gözükmeyen fakat ters istikamete açık.	UB	YOD, salondan direkt görünmemelidir.	YOD'un salonu bakan tarafı duvar ile kapalı, diğer yönleri açık.
4	O zaman çok az yer kaplayacak 2.80 - 3.00 m çaplı bir yer fazlasıyla kafi bütün mekan oturma hacmine kalacak.	UB	Yuvarlak YOD nişi için 2.80 m. veya 3.00 çap yeterlidir.	
5	Burayada yine kapalı veya açık mutfak gelecek, kapalı olur çünkü öyle isteniyor.	UT	Kapalı veya açık MUT kullanılabilir.	'AB' üzerinde 'EFGH' DD; MUT.
6	Rüzgarlıklarda bir giriş gelecek.	UB	GİR HA'da olmalıdır.	'A' köşesinden GİR

M-7. Eskiz 19. Tasarım Stratejisi

Tablo 4.20. Eskiz 19'un Tasarım Stratejisi.

- Giriş Hakim Rüzgara bakmalıdır.
- Yatak Odası Kuzeye bakmalıdır.
- Banyo ve Mutfak Manzarayı görmemelidir.
- Yatak Odası sikülasyondan çıkarılmalıdır.
- Yatak Odası Daire formu olmalıdır.
- Mutfak açık yada kapalı olabilir.
- Mekanların yerleştirilme sırası; Salon, Yatak Odası, Banyo, Mutfak ve Giriş'tir.



Şekil 4.18. Eskiz 19'un Bilgisayar Çizimi.

M.7. Eskiz 19.'deki US'ın kullanacağı alan ve uzunluk oranları
(Ölçüler Mimar tarafından verilmiştir.)

THESIS.LSP

;Starting Values

```
(setq aral 225.0
      arx 15.0
      ary 15.0
      aralmin 225.0
      arxcek 0.0
      arycek 0.0
      otalmin 35.0
      KAKS 1.0
      TAKS 1.155
      kuz_deg 45.0
      ruz_deg 135.0
      man_deg 270)
```

;End of Starting Values Başlangıç Değerlerini ata.

- aral: arazi alanı 225m2
- arx: arazinin eni 15 m.
- ary: arazinin boyu 15m2.
- aralmin: minimum arazi alanı 225m2.
- arxcek: endeki çekilme değeri 0.
- arycek: boydaki çekilme değeri 0m.
- otalmin: evin minimum oturma alanı 35 m2.
- KAKS, 1.
- TAKS, 1.155
- kuz_deg: kuzey yönü 45 derece.
- ruz_deg: hakim rüzgar yönü 135 der.
- man_deg: manzara açısı 270 derece.

Min., Maks. ve Standart Değerler			
Arazi		Diger	
Alanı:	225.0	Min. Ev Net Alanı:	35.0
Min. :	225.0	KAKS :	1.0
Boy :	15.0	TAKS :	0.2
Eni :	15.0	Kuzey Yonu :	45.0
Boydaki Çekilme:	0.0	Rüzgar Yonu :	135.0
Endeki Çekilme :	0.0	Manzara Yonu:	270.0
Tamam		Yardım	

Şekil 4.20. Standartlar Sorgu Kutusu.

STANDART'S

```
(defun c:STA ()
  (setq dcl_id (load_dialog "thesis.dcl"))
  (if (not (new_dialog "standarts" dcl_id))
      (exit))
  (set_tile "t_aral" (rtos aral))
  (set_tile "t_arx" (rtos arx))
)
```

```

(set_tile "t_ary" (rtos ary))
(set_tile "t_aralmin" (rtos aralmin))
(set_tile "t_arxcek" (rtos arxcek))
(set_tile "t_arycek" (rtos arycek))
(set_tile "t_otalmin" (rtos otalmin))
(set_tile "t_kaks" (rtos kaks))
(set_tile "t_taks" (rtos taks))
(set_tile "t_kuz_yon" (rtos kuz_deg))
(set_tile "t_ruz_yon" (rtos ruz_deg))
(set_tile "t_man_yon" (rtos man_deg))
(action_tile "ok" "(done_dialog)")
(action_tile "help" "(acad_helpdlg \"thesis.hlp\" \"standart\")")
(start_dialog)
(unload_dialog dcl_id)

```

Menüdeki Standartlar seçeneği seçildiğinde harekete geçecek

program.

- 'standart' sorgu kutusunu çıkarır.
- ve sorgu kutusunda çıkacak değerleri oranlara göre hesapla.
- sorgu kutusunda çık.



Şekil 4.21. Kuzey Yönü Sorgu Kutusu.

```

KUZEYDLG_ACT
(defun KUZEYDLG_ACT ()
  (defun kuz_msg ()
    (set_tile "error" (strcat "Mevcut Kuzey Yonu: " (rtos kuz_deg)))
  )
  : SOL_UST_ACT
(defun SOL_UST_ACT ()
  (setq KUZ_DEG 135.0)
  (kuz_msg))
  : UST_ACT
(defun UST_ACT ()
  (setq KUZ_DEG 90.0)
  (kuz_msg))
  : SAG_UST_ACT

```

```

(defun SAG_UST_ACT ()
  (setq KUZ_DEG 45.0)
  (kuz_msg))
; SAG_ACT
(defun SAG_ACT ()
  (setq KUZ_DEG 0.0)
  (kuz_msg))
; SAG_ALT_ACT
(defun SAG_ALT_ACT ()
  (setq KUZ_DEG 315.0)
  (kuz_msg))
; ALT_ACT
(defun ALT_ACT ()
  (setq KUZ_DEG 270.0)
  (kuz_msg))
; SOL_ALT_ACT
(defun SOL_ALT_ACT ()
  (setq KUZ_DEG 225.0)
  (kuz_msg))
; SOL_ACT
(defun SOL_ACT ()
  (setq KUZ_DEG 180.0)
  (kuz_msg))

```

- Ana sorgu kutusundan çıkan Kuzey yönü belirlemek için kullanılan sorgu kutusunun tanımlanması ve tuşlarına fonksiyonlarının atanması.

```

(if (not (new_dialog "kuzey_yon" dcl_id))
  (exit))
(setq x (dimx_tile "ev"))
(setq y (dimy_tile "ev"))
(start_image "ev")
  (slide_image 0 0 x y "ev")
  (end_image)
  (setq x (dimx_tile "y_sol_ust"))
  (setq y (dimy_tile "y_sol_ust"))
(start_image "y_sol_ust")
  (slide_image 0 0 x y "ok-5")
  (end_image)
  (setq x (dimx_tile "y_ust"))
  (setq y (dimy_tile "y_ust"))
(start_image "y_ust")
  (slide_image 0 0 x y "ok-6")
  (end_image)
  (setq x (dimx_tile "y_sag_ust"))
  (setq y (dimy_tile "y_sag_ust"))
(start_image "y_sag_ust")
  (slide_image 0 0 x y "ok-7")
  (end_image)
  (setq x (dimx_tile "y_sag"))
  (setq y (dimy_tile "y_sag"))
(start_image "y_sag")

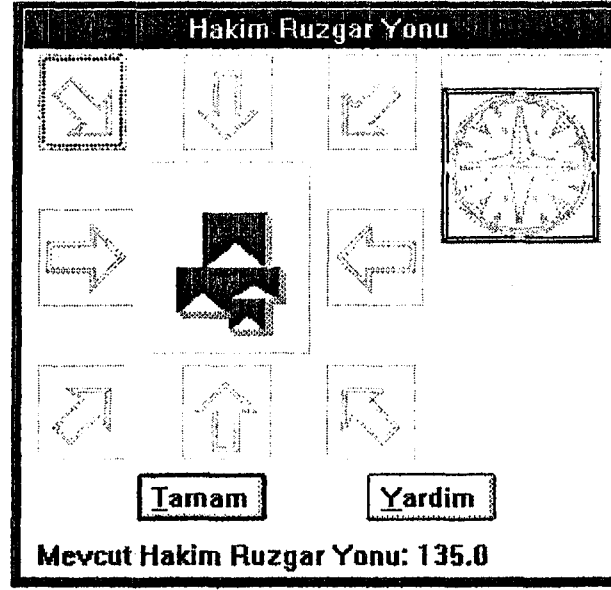
```

```

        (slide_image 0 0 x y "ok-8")
        (end_image)
        (setq x (dimx_tile "y_sag_alt"))
        (setq y (dimy_tile "y_sag_alt"))
        (start_image "y_sag_alt")
        (slide_image 0 0 x y "ok-1")
        (end_image)
        (setq x (dimx_tile "y_alt"))
        (setq y (dimy_tile "y_alt"))
        (start_image "y_alt")
        (slide_image 0 0 x y "ok-2")
        (end_image)
        (setq x (dimx_tile "y_sol_alt"))
        (setq y (dimy_tile "y_sol_alt"))
        (start_image "y_sol_alt")
        (slide_image 0 0 x y "ok-3")
        (end_image)
        (setq x (dimx_tile "y_sol"))
        (setq y (dimy_tile "y_sol"))
        (start_image "y_sol")
        (slide_image 0 0 x y "ok-4")
        (end_image)
        (setq x (dimx_tile "kuz_yon"))
        (setq y (dimy_tile "kuz_yon"))
        (start_image "kuz_yon")
        (slide_image 0 0 x y "kuzey")
        (end_image)
        (kuz_msg)
        (action_tile "y_sol_ust" "(SOL_UST_ACT)")
        (action_tile "y_ust" "(UST_ACT)")
        (action_tile "y_sag_ust" "(SAG_UST_ACT)")
        (action_tile "y_sag" "(SAG_ACT)")
        (action_tile "y_sag_alt" "(SAG_ALT_ACT)")
        (action_tile "y_alt" "(ALT_ACT)")
        (action_tile "y_sol_alt" "(SOL_ALT_ACT)")
        (action_tile "y_sol" "(SOL_ACT)")
        (action_tile "ok" "(Tamam2)")
        (action_tile "help" "{acad_helpdlg \\thesis.hlp\\ \"kuzey\"}")
        (start_dialog)
)

```

- Kuzey yönü belirleme sorgu kutusundaki resimleri yerine yerleştir.



Şekil 4.22. Hakim Rüzgar Sorgu Kutusu

```

HAKIMDLG_ACT
(defun HAKIMDLG_ACT ()
  (defun ruz_msg ()
    (set_tile "error" (strcat "Mevcut Hakim Rüzgar Yonu: " (itos ruz_deg)))
  )
  ; SOL_UST_ACT
  (defun SOL_UST_ACT ()
    (setq RUZ_DEG 135.0)
    (ruz_msg))
  ; UST_ACT
  (defun UST_ACT ()
    (setq RUZ_DEG 90.0)
    (ruz_msg))
  ; SAG_UST_ACT
  (defun SAG_UST_ACT ()
    (setq RUZ_DEG 45.0)
    (ruz_msg))
  ; SAG_ACT
  (defun SAG_ACT ()
    (setq RUZ_DEG 0.0)
    (ruz_msg))
  ; SAG_ALT_ACT
  (defun SAG_ALT_ACT ()
    (setq RUZ_DEG 315.0)
    (ruz_msg))
  ; ALT_ACT
  (defun ALT_ACT ()
    (setq RUZ_DEG 270.0)
    (ruz_msg))
  ; SOL_ALT_ACT
  (defun SOL_ALT_ACT ()
    (setq RUZ_DEG 225.0)
    (ruz_msg))
  ; SOL_ACT
  (defun SOL_ACT ()

```

```
(setq RUZ_DEG 180.0)
(ruz_msg))
```

- **Ana sorgu kutusundan çıkan hakim rüzgarı belirlemek için kullanılan sorgu kutusunun tanımlanması ve tuşlarına fonksiyonlarının atanması.**

```
(if (not (new_dialog "hakim_ruz" dcl_id))
    (exit))
(setq x (dimx_tile "ev"))
(setq y (dimy_tile "ev"))
(start_image "ev")
(slide_image 0 0 x y "ev")
(end_image)
(setq x (dimx_tile "y_sol_ust"))
(setq y (dimy_tile "y_sol_ust"))
(start_image "y_sol_ust")
(slide_image 0 0 x y "ok-1")
(end_image)
(setq x (dimx_tile "y_ust"))
(setq y (dimy_tile "y_ust"))
(start_image "y_ust")
(slide_image 0 0 x y "ok-2")
(end_image)
(setq x (dimx_tile "y_sag_ust"))
(setq y (dimy_tile "y_sag_ust"))
(start_image "y_sag_ust")
(slide_image 0 0 x y "ok-3")
(end_image)
(setq x (dimx_tile "y_sag"))
(setq y (dimy_tile "y_sag"))
(start_image "y_sag")
(slide_image 0 0 x y "ok-4")
(end_image)
(setq x (dimx_tile "y_sag_alt"))
(setq y (dimy_tile "y_sag_alt"))
(start_image "y_sag_alt")
(slide_image 0 0 x y "ok-5")
(end_image)
(setq x (dimx_tile "y_alt"))
(setq y (dimy_tile "y_alt"))
(start_image "y_alt")
(slide_image 0 0 x y "ok-6")
(end_image)
(setq x (dimx_tile "y_sol_alt"))
(setq y (dimy_tile "y_sol_alt"))
(start_image "y_sol_alt")
(slide_image 0 0 x y "ok-7")
(end_image)
(setq x (dimx_tile "y_sol"))
(setq y (dimy_tile "y_sol"))
(start_image "y_sol")
(slide_image 0 0 x y "ok-8")
(end_image)
```

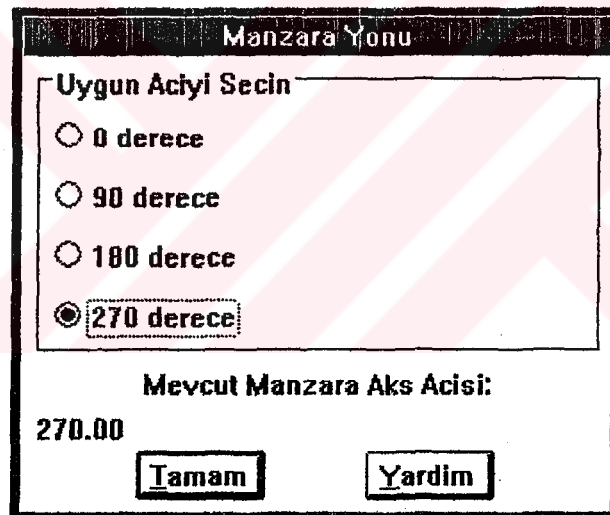
```

(setq x (dimx_tile "hak_yon"))
(setq y (dimy_tile "hak_yon"))
(start_image "hak_yon")
(slide_image 0 0 x y "ruzgar")
(end_image)
(ruz_msg)

(action_tile "y_sol_ust" "(SOL_UST_ACT)")
(action_tile "y_ust" "(UST_ACT)")
(action_tile "y_sag_ust" "(SAG_UST_ACT)")
(action_tile "y_sag" "(SAG_ACT)")
(action_tile "y_sag_alt" "(SAG_ALT_ACT)")
(action_tile "y_alt" "(ALT_ACT)")
(action_tile "y_sol_alt" "(SOL_ALT_ACT)")
(action_tile "y_sol" "(SOL_ACT)")
(action_tile "ok" "(Tamam2)")
(action_tile "help" "(acad_helpdlg \"thesis.hlp\" \"ruzgar\")")
(start_dialog)
)

```

- Hakim rüzgar sorgu kutusundaki resimleri yerine yerleştir.



Şekil 4.23. Manzara Yönü Sorgu Kutusu

```

MAN_DLG_ACT
(defun MAN_DLG_ACT ()
  (if (not (new_dialog "man_yon" dcl_id))
    (exit))
  (mode_tile "b_1A" 1)
  (mode_tile "b_3A" 1)
  (mode_tile "b_5A" 1)
  (mode_tile "b_7A" 1)
  (set_tile "error" (rtos man_deg))
  (cond
    ((= man_deg 0)
     (set_tile "b_1" "I"))
  )
  ((= man_deg 90)

```

```

(set_tile "b_3" "1")
)
( (= man_deg 180)
(set_tile "b_5" "1")
)
( (= man_deg 270)
(set_tile "b_7" "1")
)
)
(defun m_a (aks)
(setq man_deg aks)
(set_tile "error" (rtos man_deg))
)
(action_tile "b_1" "{m_a 0}")
(action_tile "b_3" "{m_a 90}")
(action_tile "b_5" "{m_a 180}")
(action_tile "b_7" "{m_a 270}")
(action_tile "ok" "(Tamam2)")
(action_tile "help" "(acad_helpdlg \"thesis.hlp\" \"manzara\")")
(start_dialog)
)

```

- Ana sorgu kutusundan çıkan manzara yönü sorgu kutusunu çiz, değerlerini ata.

Mekan Net Alanları			
Salon	Mutfak	Banyo	Yatak Odası
Alanı: 17.2	Alanı: 4.5	Alanı: 3.6	Alanı: 9.7
Boyu : 3.3	Boyu : 1.4	Boyu : 1.5	Boyu : 2.7
Eni : 5.2	Eni : 3.3	Eni : 2.4	Eni : 3.5
Evin Net Alanı: 34.9			
Tamam		Yardım	

Şekil 4.24. Mekan Alanları Sorgu Kutusu

```

MEK_DEGER_DLG_ACT
(defun MEK_DEGER_DLG_ACT ()
(MEK_HESAP)
(defun MEK_DEGER_T_OTAL_ACT ()
(set_tile "error" "Bu surum'de Evin Net Alanı bu sorgu kutusundan
degistirilemez!")
(set_tile "t_otal" (rtos otal))
(mode_tile "ok" 2)
)
(if (not (new_dialog "mek_deger" dcl_id))
(exit))
(set_tile "t_sal" (rtos sal))
(set_tile "t_salx" (rtos salx))
(set_tile "t_saly" (rtos saly))
(set_tile "t_mut" (rtos mut))
(set_tile "t_mutx" (rtos mutx))
)

```

```
(set_tile "t_muty" (rtos muty))
(set_tile "t_yat" (rtos yat))
(set_tile "t_yatx" (rtos yatx))
(set_tile "t_yaty" (rtos yaty))
(set_tile "t_ban" (rtos ban))
(set_tile "t_banx" (rtos banx))
(set_tile "t_bany" (rtos bany))
(set_tile "t_otal" (rtos otal))
(action_tile "t_otal" "(MEK_DEGER_1_OTAL_ACT)")
(action_tile "ok" "(Tamam2)")
(action_tile "help" "(acad_helpdlg \"thesis.hlp\" \"mekan\")")
(start_dialog)
```

) - Ana sorgu kutusu altından çıkan evin mekan alan ve boyutları gösteren sorgu kutusunu çağır ve değerlerini göster.

```
; Tasarım Stratejileri Sorgu Kutusu
(defun STRA_DIALOG_ACT ()
  (if (not (new_dialog "strateji" dcl_id))
    (exit))
  (if (<= OTAL 55.00)
    (mode_tile "stra_5" 1)(set_tile "stra_5" "1")
  )
  (set_tile "stra_1" "1")
  (set_tile "stra_2" "1")
  (set_tile "stra_3" "1")
  (set_tile "stra_4" "1")
  (action_tile "stra_1" "(setq tastr_1 1)")
  (action_tile "stra_2" "(setq tastr_2 1)")
  (action_tile "stra_3" "(setq tastr_3 1)")
  (action_tile "stra_4" "(setq tastr_4 1)")
  (action_tile "stra_5" "(setq tastr_5 1)")
  (action_tile "ok" "(Tamam2)")
  (action_tile "help" "(acad_helpdlg \"thesis.hlp\" \"strateji\")")
  (start_dialog)
)
```

- Kullanıcının hangi tasarım stratejileri istediğini öğren.

THESIS

```
(defun c:the (/ sal, ermsg, salx, saly, mut, evxmin, mutx, evymin, muty, ban, otal, banx,
bany, yat, yatx, yaty)
  (setvar "regenmode" 1)
```

- Ana programı çalıştırıp değişkenleri sıfırla.

```
; Mekan Alan Hesapları
(defun MEK_HESAP ()
  (setq otal (* aral TAKS)
```

```
salor 0.4918
mutor 0.12915
```

```

banor 0.10211
yator 0.27694

sal (* otal salor)
salx (sqrt (/ SAL 1.59))
saly (/ SAL salx)
mut (* otal mutor)
mux (sqrt (/ MUT 2.46))
muty (/ MUT mux)
ban (* otal banor)
banx (sqrt (/ BAN 1.60))
bany (/ BAN banx)
yal (* otal yator)
yatx (sqrt (/ YAT 1.29))
yaty (/ YAT yatx)

```

```

evxmin (* 2.0 salx)
evymin (* 2.0 salx)

```

)) - daha sonra literatürededen alınan tasarım stratejilerinde mekan alanlarını hesaplamak için gerekli olacak programı yükle.

```

; CHECK_ARAL
(defun ARALedit_ACT (ARAL_NEW)
  (MEK_HESAP)
  (setq aral_new (distof aral_new))
  (if (and (/= ARAL_NEW ARAL) (> (* ARAL_NEW TAKS) otalmin) (> aral_new (* (+ (*
2.0 arxcek) evxmin) (+ (* 2.0 aryce) evymin))))
    (progn
      (set_tile "error" "")
      (setq aral aral_new)
      (setq arx (sqrt aral) ary arx)
      (MEK_HESAP)
      (set_tile "edit_aral" (rtos aral))
      (set_tile "edit_arx" (rtos arx))
      (set_tile "edit_ary" (rtos ary))
      (set_tile "t_otal" (rtos otal))
      (mode_tile "edit_aral" 2)
    )
    (progn
      (if (/= aral_new aral)
        (progn
          (setq errmsg "Arazi için Kucuk Deger.")
          (set_tile "error" errmsg)
          (set_tile "edit_aral" (rtos aral))
          (mode_tile "edit_aral" 2)
        )
      )
    )
  )
)

```

- Ana sorgu kutusunda arazi alanı değeri değiştirilirse kontrol et. Eğer verilen değer kabul edildenden küçükse, mevcut TAKS değeri ile evin oturma

alanı minimumdan aşağı düşüyorsa veya mevcut arazideki çekilme değerleriyle evin oturması için yeterli genişlik kalmıyorsa verilen değeri kabul etme ve yeni değeri alana kadar eski değeri göster.

```

; CHECK_ARX
(defun ARXedit_ACT (ARX_NEW)
  (setq arx_new (distof arx_new))
  (if (and (/= arx_new arx) (> (* arx_new ary TAKS) otalmin) (> arx_new (+ evxmin (*
2.0 arxcek))))
    (progn
      (set_tile "error" "")
      (setq arx arx_new)
      (setq aral (* arx ary))
      (MEK_HESAP)
      (set_tile "edit_aral" (rtos aral))
      (set_tile "edit_arx" (rtos arx))
      (set_tile "t_otal" (rtos otal))
      (mode_tile "edit_arx" 2)
    )
    (progn
      (if (/= arx_new arx)
        (progn
          (setq errmsg "Arazinin Boyu için Kucuk Deger.")
          (set_tile "error" errmsg)
          (set_tile "edit_arx" (rtos arx))
          (mode_tile "edit_arx" 2)
        )
      )
    )
  )
)

```

- Ana sorgu kutusunda arazinin eni değeri değiştirilirse kontrol et. Eğer verilen değer kabul edileden küçükse, mevcut TAKS değeri ile evin oturma alanı minimumdan aşağı düşüyorsa veya mevcut arazideki çekilme değerleriyle evin oturması için yeterli genişlik kalmıyorsa verilen değeri kabul etme ve yeni değeri alana kadar eski değeri göster.

```

; CHECK_KAKS
(defun KAKSedit_ACT (KAKS_NEW)
  (if (/= kaks_new kaks)
    (progn
      (set_tile "error" "Deger atandi fakat Ev bir katli kabul edilmistir.")
      (setq KAKS (distof KAKS_new))
      (set_tile "edit_kaks" (rtos KAKS))
      (mode_tile "edit_taks" 2)
    )
  )
)

```

```

; CHECK_TAKS
(defun TAKSedit_ACT (TAKS_NEW)
  (setq taks_new (distof taks_new))
  (if (/= taks_new taks)
    (progn
      (setq TAKS_OLD TAKS TAKS_NEW)
      (MEK_HESAP)
      (if (and (>= (* aral TAKS_NEW) otalmin) (<= evxmin (- arx (* arxcek
2.0))) (<= evymin (- aiy (* arycek 2.0))) )
        (progn
          (set_tile "error" "")
          (MEK_HESAP)
          (set_tile "edit_taks" (rtos TAKS))
          (set_tile "t_otal" (rtos otal))
          (mode_tile "edit_taks" 2)
        )
        (progn
          (setq errmsg (strcat "Min. TAKS: " (rtos (/ otal aral))
" Maks. TAKS: " (rtos (/ (* (+ evxmin (* arxcek 2.0)) (+ evymin (* arycek 2.0)) ) aral)) ))
          (set_tile "error" errmsg)
          (setq TAKS TAKS_OLD)
          (set_tile "edit_taks" (rtos TAKS_OLD))
          (mode_tile "edit_taks" 2)
        )
      )
    )
  )
)

```

- Ana sorgu kutusunda arazinin TAKS değeri değiştirilirse kontrol et. Eğer verilen TAKS değeri ile evin oturma alanı minimumdan aşağı düşüyorsa verilen değeri kabul etme ve yeni değeri alana kadar eski değeri göster.

```

; CHECK_ARXCEK
(defun ARXCEKedit_ACT (ARXCEK_NEW)
  (setq arxcek_new (distof arxcek_new))
  (if (/= arxcek_new arxcek)
    (progn
      (if (<= (+ (* 2.0 arxcek_new) evxmin) arx)
        (progn
          (set_tile "error" "")
          (setq ARXCEK ARXCEK_NEW)
          (MEK_HESAP)
          (set_tile "edit_arxcek" (rtos arxcek))
          (mode_tile "edit_arxcek" 2)
        )
      )
      (progn
        (setq errmsg (strcat "Arazini Boydaki cekilmesi "
(rtos (/ (- aral evxmin) 2.0)) "dan kucuk olamaz."))
        (set_tile "error" errmsg)
        (set_tile "edit_arxcek" (rtos arxcek))
        (mode_tile "edit_arxcek" 2)
      )
    )
  )
)

```

- Ana sorgu kutusunda arazinin enindeki çekilme değeri değiştirilirse kontrol et. Eğer verilen değerle oturması için gerekli değerden daha küçük bir mesafe kalıyorsa arazide verilen değeri kabul etme ve yeni değeri alana kadar eski değeri göster.

```

CHECK_ARYCEK
(defun ARYCEKedit_ACT (ARYCEK_NEW)
  (setq aryceck_new (distof aryceck_new))
  (if (/= aryceck_new aryceck)
    (progn
      (if (<= (+ (* 2.0 aryceck_new) ewymin) ary)
        (progn
          (set_tile "error" "")
          (setq ARYCEK ARYCEK_NEW)
          (MEK_HESAP)
          (set_tile "edit_arycek" (rtos aryceck))
          (mode_tile "edit_arycek" 2)
        )
      )
    )
  )
  (progn
    (setq errmsg (strcat "Arazini Endeki cekilmesi " (rtos
      (/ (- aral ewymin) 2.0)) "dan kucuk olamaz."))
    (set_tile "error" errmsg)
    (set_tile "edit_arycek" (rtos aryceck))
    (mode_tile "edit_arycek" 2)
  )
)

```

- Ana sorgu kutusunda arazinin enindeki çekilme değeri değiştirilirse kontrol et. Eğer verilen değerle oturması için gerekli değerden daha küçük bir mesafe kalıyorsa arazide verilen değeri kabul etme ve yeni değeri alana kadar eski değeri göster.

```

CHECK_OTAL
(defun OTALedit_ACT (OTAL_NEW)
  (setq OTAL_new (distof OTAL_new))
  (if (/= OTAL_new OTAL)
    (progn
      (cond
        ((< OTAL_NEW OTALMIN)
          (progn
            (setq errmsg (strcat "Konutun Alani " (rtos OTALMIN)
              "m2'den kucuk olamaz."))
            (set_tile "error" errmsg)
            (set_tile "edit_otal" (rtos otal))
          )
        )
      )
    )
  )
)

```

```

(mode_tile "edit_otat" 2)
))
(> OTAL_NEW OTALMAX)
(progn
(setq errmsg (strcat "Konutun Alanı " (rtos OTALMAX)
"m2'den büyük olamaz."))
(set_tile "error" errmsg)
(set_tile "edit_otat" (rtos otal))
(mode_tile "edit_otat" 2)
))
(> OTAL_NEW OTALHEP)
(progn
(setq errmsg (strcat "Konutun Alanı " (rtos OTALHEP)
"m2'den büyük olamaz."))
(set_tile "error" errmsg)
(set_tile "edit_otat" (rtos otal))
(mode_tile "edit_otat" 2)
))
(set_tile "error" "")
(setq OTAL_OTAL_NEW)
(MEK_HESAP)
(set_tile "edit_otat" (rtos otal))
(mode_tile "edit_otat" 2)
)
)
)
)
)

```

- Ana sorgu kutusunda konutun alan değeri değiştirilirse kontrol et. Eğer verilen değer, minimum alandan küçükse, arazinin taksitla çarpımından büyükse veya kabullerden büyükse yeni değer al.

Şekil 4.25. ZEUS Ana Sorgu Kutusu

```

; TAMAM_1
(defun Tamam1 ()
  (STRA_DLG_ACT)
  (done_dialog)
  (unload_dialog dcl_id)
)

```

- Ana sorgu kutusundaki tamam seçilirse ana sorgu kutusunu kapat ve Zeus'un Uzman Sistemini Çalıştır.

```

; TAMAM_2
(defun Tamam2 ()
  (done_dialog)
)

```

- Ana sorgu kutusunun alt sorgu kutularındaki tamamların tanımlanması.

The screenshot shows a window titled "Arsa ve Yazlik Konut Alanlari Hesaplari". It contains several input fields and buttons:

- Arsanın** section:
 - Alani: 225.00
 - Boy: 15.00
 - Eni: 15.00
- Oranlar, Konut Alani** section:
 - KAKS: 1.00
 - TAKS: 0.16
 - K. Alan: 34.88
- Cekilmeler** section:
 - Boydaki: 0.00
 - Endeki: 0.00
- Buttons on the right:
 - Kuzey Yonu...
 - Hakim Ruzgar...
 - Manzara Yonu...
 - Mekan Alanlari...
- Buttons at the bottom:
 - Tamam
 - Cikis
 - Yardim

Şekil 4.26. ZEUS Ana Sorgu Kutusu

```

; ANA=PROGRAM
(setq dcl_id (load_dialog "thesis.dcl"))
(if (not (new_dialog "ana_deger" dcl_id))
  (exit))
(MEK_HESAP)

```

- Ana sorgu kutsunu yükle ve mekan hesaplarını yap.

```

; SET_TILE
(set_tile "edit_aral" (rtos aral))
(set_tile "edit_arx" (rtos arx))
(set_tile "edit_ary" (rtos ary))
(set_tile "edit_kaks" (rtos kaks))
(set_tile "edit_taks" (rtos taks))
(set_tile "edit_arxcek" (rtos arxcek))
(set_tile "edit_arycek" (rtos arycek))
(set_tile "t_otal" (rtos otal))

```

- Ana sorgu kutusundaki değer alanlarına değerlerini yaz.

```

; ACTIONS
(action_tile "edit_aral" "(ARALedit_ACT $value)")
(action_tile "edit_arx" "(ARXedit_ACT $value)")
(action_tile "edit_ary" "(ARYedit_ACT $value)")
(action_tile "edit_kaks" "(KAKSedit_ACT $value)")
(action_tile "edit_taks" "(TAKSedit_ACT $value)")
(action_tile "edit_arxcek" "(ARXCEKedit_ACT $value)")
(action_tile "edit_arycek" "(ARYCEKedit_ACT $value)")

```

```
(action_tile "but_kuz" "(KUZEYDLG_ACT)")
(action_tile "but_hak" "(HAKIMDLG_ACT)")
(action_tile "but_man" "(MAN_DLG_ACT)")
(action_tile "but_mek" "(MEK_DEGER_DLG_ACT)")
(action_tile "help" "(acad_helpdlg \"thesis.hlp\" \"zeus\")")
(action_tile "ok" "(Tamam!)")
```

- Ana sorgu kutusundaki tuşlara basılınca programın hangi alt programı çalıştıracağını tanımla.

```
(start_dialog)
(unload_dialog dcl_id)
```

Ana sorgu kutusunu çalıştır.

- Böylece kullanıcıdan programın çalışması için gerekli değerlerin alınması işlemi sona ermiş ve kullanıcının hangi tasarım stratejilerini istediği öğrenilmiştir. Bu aşamadan sonra seçilen tasarım stratejilerine göre örnek çizimler yapılır.

```
(Defun Yazı_yaz (s h p x y / sn sb m i sp)
  (Setq
    sn (Length s) i 0
    sb (/ (* h (1 - (* 4 sn))) 6.0)
    m (List (+ (Car p) (/ x 2.0)) (+ (Cadr p) (/ y 2.0) sb)))
  (Repeat sn
    (Setq sp (List (Car m) (- (Cadr m) (* i (/ (* 4 h) 3.0)) (/ h 2.0))))
    (Entmake
      (List
        (Cons 0 "TEXT")
        (Cons 1 (Nth i s))
        (Cons 10 sp)
        (Cons 11 sp)
        (Cons 40 h)
        (Cons 72 4)
        (Cons 73 0)))
      (Setq i (1 + i)))
  (Princ))
(Defun Dikdort_Ciz (s h p x y / p2 p3 p4)
  (Setq
    p2 (List (+ (Car p) x) (Cadr p))
    p4 (List (Car p) (+ (Cadr p) y))
    p3 (List (Car p2) (Cadr p4)))
  (Command "PLINE" p p2 p3 p4 "C")
  (Yazı_yaz s h p x y))
(Defun Strj2 (salon_x salon_y yatak_x yatak_y
  giris_x giris_y banyo_x banyo_y
  mutfak_x mutfak_y
  m_yon_d h_yon_d k_yon_d yazı_yuk /
  mekan_merkez mm_x mm_y mekan_x mekan_y
  salon_p giris_p yatak_p banyo_p mutfak_p)
```

```

(Setvar "LIMCHECK" 0)
(Setq
  mekan_merkez (Getvar "VIEWCTR")
  mm_x (Car mekan_merkez) mm_y (Cadr mekan_merkez))
(If (Equal k_yon_d 0 0.01))(Progn
  (Setq
    mekan_x (+ salon_x yatak_x) mekan_y (+ salon_y mutfak_y)
    salon_p (List (- mm_x (/ mekan_y 2.0)) (- (+ mm_y
(/ mekan_x 2.0)) salon_x))
    yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) (- salon_y yatak_y)) (-
mm_y (/ mekan_x 2.0)))
    banyo_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) (- yatak_x banyo_x)))
    mutfak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) salon_x)))
    (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_y salon_x)
    (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
    (Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_y banyo_x)
    (Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_y mutfak_x)
  ))
  (If (Equal k_yon_d 45 0.01))(Progn
    (Setq
      mekan_x salon_x mekan_y (+ salon_y mutfak_y)
      salon_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
      yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (+ (-
mm_y (/ mekan_y 2.0)) salon_y))
      banyo_p (List (- (+ mm_x (/ mekan_x 2.0)) (+ yatak_x banyo_x)) (+
(- mm_y (/ mekan_y 2.0)) salon_y))
      mutfak_p (List (- mm_x (/ mekan_x 2.0)) (+ (- mm_y
(/ mekan_y 2.0)) salon_y)))
      (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
      (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
      (Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_x banyo_y)
      (Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_x mutfak_y)
    ))
    (If (Equal k_yon_d 90 0.01))(Progn
      (Setq
        mekan_x (+ salon_x yatak_x) mekan_y (+ salon_y mutfak_y)
        salon_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
        yatak_p (List (+ (- mm_x (/ mekan_x 2.0)) salon_x) (+ (-
mm_y (/ mekan_y 2.0)) (- salon_y yatak_y)))
        banyo_p (List (+ (- mm_x (/ mekan_x 2.0)) salon_x) (+ (-
mm_y (/ mekan_y 2.0)) salon_y))
        mutfak_p (List (+ (- mm_x (/ mekan_x 2.0)) (- salon_x mutfak_x)) (+
(- mm_y (/ mekan_y 2.0)) salon_y)))
        (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
        (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
        (Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_x banyo_y)
        (Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_x mutfak_y)
      ))
      (If (Equal k_yon_d 135 0.01))(Progn
        (Setq

```

```

mekan_x (+ salon_x mutfak_x) mekan_y (+ salon_y yatak_y)
salon_p (List (- (+ mm_x (/ mekan_x 2.0)) salon_x) (- mm_y
(/ mekan_y 2.0)))
yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) yatak_y))
banyo_p (List (- (+ mm_x (/ mekan_x 2.0)) (+ yatak_x banyo_x)) (-
(+ mm_y (/ mekan_y 2.0)) yatak_y))
mutfak_p (List (- mm_x (/ mekan_x 2.0)) (+ (- mm_y
(/ mekan_y 2.0)) (- salon_y mutfak_y))))
(Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_x banyo_y)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_x mutfak_y)
))
(If (Equal k_yon_d 180 0.01) (Progn
(Setq
mekan_x (+ salon_x yatak_x) mekan_y (+ salon_y mutfak_y)
salon_p (List (+ (- mm_x (/ mekan_y 2.0)) mutfak_y) (-
mm_y (/ mekan_x 2.0)))
yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) mutfak_y) (+ (-
mm_y (/ mekan_x 2.0)) salon_x))
banyo_p (List (+ (- mm_x (/ mekan_y 2.0)) (- mutfak_y banyo_y))
(+ (- mm_y (/ mekan_x 2.0)) salon_x))
mutfak_p (List (- mm_x (/ mekan_y 2.0)) (+ (-
mm_y (/ mekan_x 2.0)) (- salon_x mutfak_x))))
(Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_y salon_x)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_y banyo_x)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_y mutfak_x)
))
(If (Equal k_yon_d 225 0.01) (Progn
(Setq
mekan_x (+ salon_x yatak_x) mekan_y (+ salon_y mutfak_y)
salon_p (List (- (+ mm_x (/ mekan_x 2.0)) salon_x) (- (+
mm_y (/ mekan_y 2.0)) salon_y))
yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) salon_y))
banyo_p (List (+ (- mm_x (/ mekan_x 2.0)) (- yatak_x banyo_x)) (-
(+ mm_y (/ mekan_y 2.0)) (+ salon_y banyo_y)))
mutfak_p (List (- (+ mm_x (/ mekan_x 2.0)) mutfak_x) (-
mm_y (/ mekan_y 2.0))))
(Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_x banyo_y)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_x mutfak_y)
))
(If (Equal k_yon_d 270 0.01) (Progn
(Setq
mekan_x (+ salon_x yatak_x) mekan_y (+ salon_y mutfak_y)
salon_p (List (- (+ mm_x (/ mekan_x 2.0)) salon_x) (- (+ mm_y (/
mekan_y 2.0)) salon_y))
yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) salon_y))

```

```

banyo_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) (+ salon_y banyo_y)))
mutfak_p (List (- (+ mm_x (/ mekan_x 2.0)) mutfak_x) (- mm_y (/
mekan_y 2.0))))
(Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_x banyo_y)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_x mutfak_y)
))
(If (Equal k_yon_d 315 0.01) (Progn
(Setq
mekan_x (+ salon_x mutfak_x) mekan_y (+ salon_y yatak_y)
salon_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) salon_y))
yatak_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
banyo_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (+ (- mm_y (/
mekan_y 2.0)) (- yatak_y banyo_y)))
mutfak_p (List (+ (- mm_x (/ mekan_x 2.0)) salon_x) (- (+ mm_y (/
mekan_y 2.0)) salon_y)))
(Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_x banyo_y)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_x mutfak_y)
))

```

- Tasarım Stratejisi L.2. Literatürden çıkarılan tasarım stratejisinin programı

```

(Defun Yazı_yaz (s1 h p x y / sn sb m i sp)
(Setq
sn (Length s1) i 0
sb (/ (* h (1- (* 4 sn))) 6.0)
m (List (+ (Car p) (/ x 2.0)) (+ (Cadr p) (/ y 2.0) sb)))
(Repeat sn
(Setq sp (List (Car m) (- (Cadr m) (* i (/ (* 4 h) 3.0)) (/ h 2.0))))
(Entmake
(List
(Cons 0 "TEXT")
(Cons 1 (Nth i s1))
(Cons 10 sp)
(Cons 11 sp)
(Cons 40 h)
(Cons 72 4)
(Cons 73 0)))
(Setq i (1+ i)))
(Princ))
(Defun Dikdort_Ciz (s h p x y / p2 p3 p4)
(Setq
p2 (List (+ (Car p) x) (Cadr p))
p4 (List (Car p) (+ (Cadr p) y))
p3 (List (Car p2) (Cadr p4)))
(Command "PLINE" p p2 p3 p4 "C")

```

```

(Yazi_yaz s h p x y))
(Defun Strj4 (salon_x salon_y yatak_x yatak_y
  giris_x giris_y banmut_x banmut_y
  m_yon_d h_yon_d k_yon_d yazi_yuk /
  mekan_merkez mm_x mm_y mekan_x mekan_y
  salon_p giris_p yatak_p banmut_p)
  (Setvar "LIMCHECK" 0)
  (Setq
    mekan_merkez (Getvar "VIEWCTR")
    mm_x (Car mekan_merkez) mm_y (Cadr mekan_merkez)
    mekan_x salon_x mekan_y (+ salon_y giris_y))
  (If (Equal m_yon_d 0 0.01) (Progn
    (Setq salon_p (List (+ (- mm_x (/ mekan_y 2.0)) giris_y) (- mm_y (/
mekan_x 2.0))))
    (If (Or (Equal h_yon_d 45 0.01) (Equal h_yon_d 90 0.01) (Equal h_yon_d
135 0.01))) (Progn
      (Setq
        giris_p (List (- mm_x (/ mekan_y 2.0)) (- (+ mm_y (/
mekan_x 2.0)) giris_x))
        yatak_p (List (- mm_x (/ mekan_y 2.0)) (- mm_y (/
mekan_x 2.0)))
        banmut_p (List (- mm_x (/ mekan_y 2.0)) (+ (- mm_y (/
mekan_x 2.0)) yatak_x)))
      ))
    (If (Or (Equal h_yon_d 225 0.01) (Equal h_yon_d 270 0.01) (Equal
h_yon_d 315 0.01))) (Progn
      (Setq
        giris_p (List (- mm_x (/ mekan_y 2.0)) (- mm_y (/ mekan_x
2.0)))
        yatak_p (List (- mm_x (/ mekan_y 2.0)) (- (+ mm_y (/
mekan_x 2.0)) yatak_x))
        banmut_p (List (- mm_x (/ mekan_x 2.0)) (+ (- mm_y (/
mekan_x 2.0)) giris_x)))
      ))
    (If (Or (Equal h_yon_d 0 0.01) (Equal h_yon_d 180 0.01)) (Progn
      (If (Or (Equal k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal
k_yon_d 135 0.01))) (Progn
        (Setq
          giris_p (List (- mm_x (/ mekan_y 2.0)) (- (+ mm_y
(/ mekan_x 2.0)) giris_x))
          yatak_p (List (- mm_x (/ mekan_y 2.0)) (- mm_y (/
mekan_x 2.0)))
          banmut_p (List (- mm_x (/ mekan_y 2.0)) (+ (-
mm_y (/ mekan_x 2.0)) yatak_x)))
        ))
      (If (Or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 225 0.01) (Equal k_yon_d 270 0.01) (Equal k_yon_d 315 0.01))) (Progn
        (Setq
          giris_p (List (- mm_x (/ mekan_y 2.0)) (- mm_y (/
mekan_x 2.0)))
          yatak_p (List (- mm_x (/ mekan_y 2.0)) (- (+ mm_y
(/ mekan_x 2.0)) yatak_x))
          banmut_p (List (- mm_x (/ mekan_y 2.0)) (+ (-
mm_y (/ mekan_x 2.0)) giris_x)))
        ))
      ))
  )

```

```

    ))
    ))
    (Dikdort_Ciz (List "SALON") yazi_yuk      salon_p salon_y salon_x)
    (Dikdort_Ciz (List "GIRIS") yazi_yuk      giris_p giris_y giris_x)
    (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
    (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_y
banmut_x)
    ))
    (If (Equal m_yon_d 90 0.01))(Progn
      (Setq salon_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/ mekan_y 2.0))
salon_y)))
      (If (Or (Equal h_yon_d 315 0.01) (Equal h_yon_d 0 0.01) (Equal h_yon_d
45 0.01)))(Progn
        (Setq
          giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- mm_y (/
mekan_y 2.0)))
          yatak_p (List (- mm_x (/ mekan_x 2.0))      (- mm_y (/
mekan_y 2.0)))
          banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (-
mm_y (/ mekan_y 2.0))))
        ))
        (If (Or (Equal h_yon_d 135 0.01) (Equal h_yon_d 180 0.01) (Equal
h_yon_d 225 0.01)))(Progn
          (Setq
            giris_p (List (- mm_x (/ mekan_x 2.0))      (- mm_y (/
mekan_y 2.0)))
            yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- mm_y
(/ mekan_y 2.0)))
            banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (-
mm_y (/ mekan_y 2.0))))
          ))
          (If (Or (Equal h_yon_d 90 0.01) (Equal h_yon_d 270 0.01)))(Progn
            (If (Or (Equal k_yon_d 315 0.01) (Equal k_yon_d 0 0.01) (Equal
k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal k_yon_d 270 0.01)))(Progn
              (Setq
                giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (-
mm_y (/ mekan_y 2.0)))
                yatak_p (List (- mm_x (/ mekan_x 2.0))      (-
mm_y (/ mekan_y 2.0)))
                banmut_p (List (+ (- mm_x (/ mekan_x 2.0))
yatak_x) (- mm_y (/ mekan_y 2.0))))
              ))
              (If (Or (Equal k_yon_d 135 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 225 0.01)))(Progn
                (Setq
                  giris_p (List (- mm_x (/ mekan_x 2.0))      (-
mm_y (/ mekan_y 2.0)))
                  yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x)
(- mm_y (/ mekan_y 2.0)))
                  banmut_p (List (+ (- mm_x (/ mekan_x 2.0))
giris_x) (- mm_y (/ mekan_y 2.0))))
                ))
                ))
            ))
          (Dikdort_Ciz (List "SALON") yazi_yuk      salon_p salon_x salon_y)

```

```

(Dikdort_Ciz (List "GIRIS") yazi_yuk      giris_p giris_x giris_y)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_x
banmut_y)
))
  (If (Equal m_yon_d 180 0.01)) (Progn
    (Setq salon_p (List (+ (- mm_x (/ mekan_y 2.0)) (- mm_y (/ mekan_x 2.0))))
    (If (Or (Equal h_yon_d 45 0.01) (Equal h_yon_d 90 0.01) (Equal h_yon_d
135 0.01))) (Progn
      (Setq
        giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) giris_x))
        yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (-
mm_y (/ mekan_x 2.0)))
        banmut_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) yatak_x)))
      ))
    (If (Or (Equal h_yon_d 225 0.01) (Equal h_yon_d 270 0.01) (Equal
h_yon_d 315 0.01))) (Progn
      (Setq
        giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- mm_y
(/ mekan_x 2.0)))
        yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) yatak_x))
        banmut_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) giris_x)))
      ))
    (If (Or (Equal h_yon_d 0 0.01) (Equal h_yon_d 180 0.01))) (Progn
      (If (Or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal k_yon_d 135 0.01))) (Progn
        (Setq
          giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- (+ mm_y (/ mekan_x 2.0)) giris_x))
          yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- mm_y (/ mekan_x 2.0)))
          banmut_p (List (+ (- mm_x (/ mekan_y 2.0))
salon_y) (+ (- mm_y (/ mekan_x 2.0)) yatak_x)))
        ))
      (If (Or (Equal k_yon_d 225 0.01) (Equal k_yon_d 270 0.01) (Equal
k_yon_d 315 0.01))) (Progn
        (Setq
          giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- mm_y (/ mekan_x 2.0)))
          yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- (+ mm_y (/ mekan_x 2.0)) yatak_x))
          banmut_p (List (+ (- mm_x (/ mekan_y 2.0))
salon_y) (+ (- mm_y (/ mekan_x 2.0)) giris_x)))
        ))
      ))
    (Dikdort_Ciz (List "SALON") yazi_yuk      salon_p salon_y salon_x)
    (Dikdort_Ciz (List "GIRIS") yazi_yuk      giris_p giris_y giris_x)
    (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
    (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_y
banmut_x)
  )
)

```

```

))
(If (Equal m_yon_d 270 0.01))(Progn
  (Setq salon_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/ mekan_y 2.0))))
  (If (Or (Equal h_yon_d 315 0.01) (Equal h_yon_d 0 0.01) (Equal h_yon_d
45 0.01)))(Progn
    (Setq
      giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
      yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) giris_y))
      banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
    ))
    (If (Or (Equal h_yon_d 135 0.01) (Equal h_yon_d 180 0.01) (Equal
h_yon_d 225 0.01)))(Progn
      (Setq
        giris_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) giris_y))
        yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
        banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
      ))
      (If (Or (Equal h_yon_d 90 0.01) (Equal h_yon_d 270 0.01)))(Progn
        (If (Or (Equal k_yon_d 315 0.01) (Equal k_yon_d 0 0.01) (Equal k_yon_d
45 0.01)))(Progn
          (Setq
            giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
            yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) giris_y))
            banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
          ))
          (If (Or (Equal k_yon_d 90 0.01) (Equal k_yon_d 135 0.01) (Equal k_yon_d
180 0.01) (Equal k_yon_d 225 0.01) (Equal k_yon_d 270 0.01)))(Progn
            (Setq
              giris_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) giris_y))
              yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
              banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
            ))
          ))
        ))
      (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
      (Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_x giris_y)
      (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
      (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_x banmut_y)
    ))
  ))
(Princ))

```

- Tasarım Stratejisi M.2. (Eskiz 2) Oturumdan çıkarılan tasarım stratejisi programı

```

(Defun Yazi_yaz (s h p x y / sn sb m i sp)
  (Setq
    sn (Length s) i 0
    sb (/ (* h (1 - (* 4 sn))) 6.0)
    m (List (+ (Car p) (/ x 2.0)) (+ (Cadr p) (/ y 2.0) sb)))
  (Repeat sn
    (Setq sp (List (Car m) (- (Cadr m) (* i (/ (* 4 h) 3.0)) (/ h 2.0))))
    (Entmake
      (List
        (Cons 0 "TEXT")
        (Cons 1 (Nth i s))
        (Cons 10 sp)
        (Cons 11 sp)
        (Cons 40 h)
        (Cons 72 4)
        (Cons 73 0)))
      (Setq i (1 + i)))
  (Princ))
(Defun Dikdort_Ciz (s h p x y / p2 p3 p4)
  (Setq
    p2 (List (+ (Car p) x) (Cadr p))
    p4 (List (Car p) (+ (Cadr p) y))
    p3 (List (Car p2) (Cadr p4))
  )
  (Command "PLINE" p p2 p3 p4 "C")
  (Yazi_yaz s h p x y)
)
(Defun Dik_Ciz (s h p x y / p2 p3 p4)
  (Setq
    p2 (polar p (dtr 45) x)
    p3 (polar p2 (dtr 135) y)
    p4 (polar p3 (dtr 225) x)
  )
  (Command "PLINE" p p2 p3 p4 "C")
  (Yazi_yaz s h p x y)
)
(Defun Strjo (salon_x salon_y yatak_x yatak_y
  giris_x giris_y banmut_d
  m_yon_d h_yon_d k_yon_d yazi_yuk /
  )
  (Setq
    mekan_merkez (Getvar "VIEWCTR")
    mm_x (Car mekan_merkez) mm_y (Cadr mekan_merkez)
  )
  (If (or (Equal m_yon_d 0 0.01)
    (Equal m_yon_d 180 0.01))
    (Progn
      (Setq salon_p (list (- mm_x (/ salon_y 2.0))
        (- mm_y (/ salon_x 2.0))
      )
      sk1 salon_p

```

```

        sk2 (list (+ (car sk1) salon_y) (cadr sk1))
        sk3 (list (car sk2) (+ (cadr sk1) salon_x))
        sk4 (list (car sk1) (cadr sk3))
    )
    (Dikdort_Ciz (list "SALON") yazi_yuk salon_p salon_y salon_x)
)
(Progn
    (Setq salon_p (list (- mm_x (/ salon_x 2.0))
                        (- mm_y (/ salon_y 2.0))
    )
    sk1 salon_p
    sk2 (list (+ (car sk1) salon_x) (cadr sk1))
    sk3 (list (car sk2) (+ (cadr sk1) salon_y))
    sk4 (list (car sk1) (cadr sk3))
    )
    (Dikdort_Ciz (list "SALON") yazi_yuk salon_p salon_x salon_y)
)
)
;Hakim Ruzgar 0 ve 45 derece icin
    (If (or (Equal h_yon_d 0 0.01)
            (Equal h_yon_d 45 0.01))
        (Progn
            (Setq temp1 (polar sk3 (dtr 225) (/ giris_y 2.0))
            giris_p (polar temp1 (dtr 315) (/ giris_x 2.0))
            gkb 1
            )
            (Dik_Ciz (list "GIRIS") yazi_yuk giris_p giris_y giris_x)
        )
    )
;Hakim Ruzgar 90 ve 135 derece icin
    (If (or (Equal h_yon_d 90 0.01)
            (Equal h_yon_d 135 0.01))
        (Progn
            (Setq temp1 (polar sk4 (dtr 315) (/ giris_y 2.0))
            giris_p (polar temp1 (dtr 225) (/ giris_x 2.0))
            gkb 2
            )
            (Dik_Ciz (list "GIRIS") yazi_yuk giris_p giris_x giris_y)
        )
    )
;Hakim Ruzgar 180 ve 225 derece icin
    (If (or (Equal h_yon_d 180 0.01)
            (Equal h_yon_d 225 0.01))
        (Progn
            (Setq temp1 (polar sk1 (dtr 225) (/ giris_y 2.0))
            giris_p (polar temp1 (dtr 315) (/ giris_x 2.0))
            gkb 3
            )
            (Dik_Ciz (list "GIRIS") yazi_yuk giris_p giris_y giris_x)
        )
    )
;Hakim Ruzgar 270 ve 315 derece icin
    (If (or (Equal h_yon_d 270 0.01)
            (Equal h_yon_d 315 0.01))

```

```

(Progn
  (Setq temp1 (polar sk2 (dtr 315) (/ giris_y 2.0))
    giris_p (polar temp1 (dtr 225) (/ giris_x 2.0))
    gkb 4
  )
  (Dik_Ciz (list "GIRIS") yazi_yuk giris_p giris_x giris_y)
)

;Banyo mutfak yerlesiyor
(if (Equal m_yon_d 0 0.01) (Setq banmut_p (list (car sk1) mm_y)))
(if (Equal m_yon_d 90 0.01) (Setq banmut_p (list mm_x (cadr sk1))))
(if (Equal m_yon_d 180 0.01) (Setq banmut_p (list (car sk2) mm_y)))
(if (Equal m_yon_d 270 0.01) (Setq banmut_p (list mm_x (cadr sk4))))
(command "circle" banmut_p "d" banmut_d)
(command "text" "j" "m" banmut_p yazi_yuk "" "BANYO" "text" "" "MUTFAK")

;yatak 0 ve 180 için
(if (or (Equal m_yon_d 0 0.01)
  (Equal m_yon_d 180 0.01))
  (Progn
    (if (Equal k_yon_d 90 0.01)
      (Setq yatak_p (list (- (+ (car sk1) (/ (distance sk1 sk2) 2.0)) (/ yatay_y 2.0))
        (- (cadr sk1) (/ yatay_x 2.0))
      )
    )
    (if (Equal k_yon_d 270 0.01)
      (Setq yatak_p (list (- (+ (car sk4) (/ (distance sk4 sk3) 2.0)) (/ yatay_y 2.0))
        (- (cadr sk4) (/ yatay_x 2.0))
      )
    )
  )
  (if (and (or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01))
    (or (= gkb 1) (= gkb 2))
    (Setq yatak_p (list (- (+ (car sk1) (/ (distance sk1 sk2) 2.0)) (/ yatay_y 2.0))
      (- (cadr sk1) (/ yatay_x 2.0))
    )
    (Setq yatak_p (list (- (+ (car sk4) (/ (distance sk4 sk3) 2.0)) (/ yatay_y 2.0))
      (- (cadr sk4) (/ yatay_x 2.0))
    )
  )
)

;yatak 90 270
(if (or (Equal m_yon_d 90 0.01)
  (Equal m_yon_d 270 0.01))
  (Progn
    (if (Equal k_yon_d 0 0.01)
      (Setq yatak_p (list (- (car sk4) (/ yatay_y 2.0))

```

```

        )
    )
    )
    (if (Equal k_yon_d 180 0.01)
        (Setq yatak_p (list (- (car sk1) (/ yatak_y 2.0))
                             (- (cadr sk1) (/ (distance sk1 sk3) 2.0)) (/ yatak_x 2.0))
        )
    )
    )
    )
    )
    (Dikdort_Ciz (list "YATAK" "ODASI") yazı_yuk yatak_p yatak_x yatak_y)
    )

```

- Tasarım Stratejisi M.4. Oturumdan çıkarılan tasarım stratejisi programı

```

(Defun Cizgi_Ciz (pl / p l)
  (If pl (Progn
    (Setq i 0)
    (Command "PLINE")
    (While (Setq p (Nth i pl))
      (Command p)
      (Setq i (1+i)))
    (Command "")))
(Defun Yazı_yaz (s h p x y / sn sb m i sp)
  (Setq
    sn (Length s) i 0
    sb (/ (* h (1- (* 4 sn))) 6.0)
    m (List (+ (Car p) (/ x 2.0)) (+ (Cadr p) (/ y 2.0) sb)))
  (Repeat sn
    (Setq sp (List (Car m) (- (Cadr m) (* i (/ (* 4 h) 3.0)) (/ h 2.0))))
    (Entmake
      (List
        (Cons 0 "TEXT")
        (Cons 1 (Nth i s))
        (Cons 10 sp)
        (Cons 11 sp)
        (Cons 40 h)
        (Cons 72 4)
        (Cons 73 0)))
    (Setq i (1+i)))
(Princ))
(Defun Dikdort_Ciz (s h p x y / p2 p3 p4)
  (Setq
    p2 (List (+ (Car p) x) (Cadr p))
    p4 (List (Car p) (+ (Cadr p) y))
    p3 (List (Car p2) (Cadr p4)))
  (Command "PLINE" p p2 p3 p4 "C")
  (Yazı_yaz s h p x y))
(Defun Strj7 (salon_x salon_y yatak_x yatak_y
  banyo_x banyo_y mutfak_x mutfak_y

```

```

m_yon_d h_yon_d k_yon_d yazi_yuk /
mekan_merkez mm_x mm_y mekan_x mekan_y mekan_x2 mekan_y2
salon_pl giris_p yatak_p banyo_p mutfak_p
salon_px yatak_px salon_py yatak_py salon_tp giris_tp)
(Setvar "LIMCHECK" 0)
(Setq
mekan_merkez (Getvar "VIEWCTR")
mm_x (Car mekan_merkez) mm_y (Cadr mekan_merkez)
mekan_x salon_x mekan_y (+ (/ mutfak_y 2.0) salon_y)
mekan_x2 (/ mekan_x 2.0) mekan_y2 (/ mekan_y 2.0)
salon_k (/ yatak_y 2.0)
salon_u (- salon_x yatak_x banyo_x mutfak_x salon_k))
(If (Equal m_yon_d 0 0.01))(Progn
(Setq
salon_px (+ (- mm_x mekan_y2) (/ mutfak_y 2.0))
yatak_px (- mm_x mekan_y2)
salon_tp (List (+ salon_px (/ mutfak_y 2.0)) (- mm_y mekan_x2)))
(If (Or (Equal h_yon_d 45 0.01) (Equal h_yon_d 90 0.01) (Equal h_yon_d
135 0.01)))(Progn
(Setq
salon_pl (List
(List salon_px (+ (- mm_y mekan_x2) salon_k))
(List salon_px (- mm_y mekan_x2))
(List (+ mm_x mekan_y2) mm_y)
(List salon_px (+ mm_y mekan_x2))
(List salon_px (- (+ mm_y mekan_x2) salon_u)))
yatak_p (List yatak_px (+ (- mm_y mekan_x2) salon_k))
banyo_p (List yatak_px (+ (- mm_y mekan_x2) salon_k
yatak_x))
mutfak_p (List yatak_px (+ (- mm_y mekan_x2) salon_k
yatak_x banyo_x))
giris_tp (List salon_px (- (+ mm_y mekan_x2) salon_u))
giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 2) 3.0))
))
(If (Or (Equal h_yon_d 225 0.01) (Equal h_yon_d 270 0.01) (Equal
h_yon_d 315 0.01)))(Progn
(Setq
salon_pl (List
(List salon_px (+ (- mm_y mekan_x2) salon_u))
(List salon_px (- mm_y mekan_x2))
(List (+ mm_x mekan_y2) mm_y)
(List salon_px (+ mm_y mekan_x2))
(List salon_px (- (+ mm_y mekan_x2) salon_k)))
yatak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u))
banyo_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x))
mutfak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x banyo_x))
giris_tp (List salon_px (- mm_y mekan_x2))
giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 3) 2.0))
))
(If (Or (Equal h_yon_d 0 0.01) (Equal h_yon_d 180 0.01)))(Progn
(If (Or (Equal k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal
k_yon_d 135 0.01)))(Progn

```

```

(Setq
salon_k))
salon_pl (List
(List salon_px (+ (- mm_y mekan_x2)
(List salon_px (- mm_y mekan_x2))
(List (+ mm_x mekan_y2) mm_y)
(List salon_px (+ mm_y mekan_x2))
(List salon_px (- (+ mm_y mekan_x2)
salon_u)))
yatak_p (List yatak_px (+ (- mm_y mekan_x2)
salon_k))
banyo_p (List yatak_px (+ (- mm_y mekan_x2)
salon_k yatak_x))
mutlak_p (List yatak_px (+ (- mm_y mekan_x2)
salon_k yatak_x banyo_x))
giris_tp (List salon_px (- (+ mm_y mekan_x2)
salon_u))
giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 2)
3.0))
))
(If (Or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 225 0.01) (Equal k_yon_d 270 0.01) (Equal k_yon_d 315 0.01)) (Progn
(Setq
salon_pl (List
(List salon_px (+ (- mm_y mekan_x2) salon_u))
(List salon_px (- mm_y mekan_x2))
(List (+ mm_x mekan_y2) mm_y)
(List salon_px (+ mm_y mekan_x2))
(List salon_px (- (+ mm_y mekan_x2) salon_k)))
yatak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u))
banyo_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x))
mutlak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x banyo_x))
giris_tp (List salon_px (- mm_y mekan_x2))
giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 3) 2.0))
))
))
(Prompt "\nsalon_x :")(Princ salon_x) (Prompt "\nsalon_y :")(Princ
salon_y)
(Prompt "\nyatak_x :")(Princ yatak_x) (Prompt "\nyatak_y :")(Princ yatak_y)
(Prompt "\nbanyo_x :")(Princ banyo_x) (Prompt "\tbanyo_y :")(Princ
banyo_y)
(Prompt "\nmutfak_x :")(Princ mutlak_x) (Prompt "\tmutfak_y :")(Princ
mutlak_y)
(Prompt "\nmekan_x :")(Princ mekan_x) (Prompt "\tmekan_y :")(Princ
mekan_y)
(Prompt "\nsalon_px :")(Princ salon_px) (Prompt "\tyatak_px :")(Princ
yatak_px)
(Prompt "\nsalon_pl :")(Princ salon_pl)
(Cizgi_Ciz salon_pl)
(Yazi_Yaz (List "SALON") yazi_yuk salon_tp (/ (* salon_y 2) 3.0) salon_x)
(Yazi_Yaz (List "GIRIS") yazi_yuk giris_tp giris_tx giris_ty)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)

```

```

(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_y banyo_x)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_y mutfak_x)
))
(If (Equal m_yon_d 90 0.01))(Progn
  (Setq
    salon_py (- mm_x mekan_x2) (+ (- mm_y mekan_y2) (/ mutfak_y
2.0))
    yatak_px (- mm_x mekan_y2)
    salon_tp (List (- mm_x mekan_x2) (+ salon_py (/ mutfak_y 2.0)) (-
mm_y mekan_x2)))
  (If (Or (Equal h_yon_d 315 0.01) (Equal h_yon_d 0 0.01) (Equal h_yon_d
45 0.01)))(Progn
    (Setq
      giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- mm_y (/
mekan_y 2.0)))
      yatak_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
      banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (-
mm_y (/ mekan_y 2.0))))
    ))
  (If (Or (Equal h_yon_d 135 0.01) (Equal h_yon_d 180 0.01) (Equal
h_yon_d 225 0.01)))(Progn
    (Setq
      giris_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
      yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- mm_y
(/ mekan_y 2.0)))
      banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (-
mm_y (/ mekan_y 2.0))))
    ))
  (If (Or (Equal h_yon_d 90 0.01) (Equal h_yon_d 270 0.01)))(Progn
    (If (Or (Equal k_yon_d 315 0.01) (Equal k_yon_d 0 0.01) (Equal
k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal k_yon_d 270 0.01)))(Progn
      (Setq
        giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (-
mm_y (/ mekan_y 2.0)))
        yatak_p (List (- mm_x (/ mekan_x 2.0)) (-
mm_y (/ mekan_y 2.0)))
        banmut_p (List (+ (- mm_x (/ mekan_x 2.0))
yatak_x) (- mm_y (/ mekan_y 2.0))))
      ))
    (If (Or (Equal k_yon_d 135 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 225 0.01)))(Progn
      (Setq
        giris_p (List (- mm_x (/ mekan_x 2.0)) (-
mm_y (/ mekan_y 2.0)))
        yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x)
(- mm_y (/ mekan_y 2.0)))
        banmut_p (List (+ (- mm_x (/ mekan_x 2.0))
giris_x) (- mm_y (/ mekan_y 2.0))))
      ))
    ))
  (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
  (Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_x giris_y)
)

```

```

(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_x
banmut_y)
))
(If (Equal m_yon_d 180 0.01) (Progn
  (Setq salon_p (List (+ (- mm_x (/ mekan_y 2.0)) (- mm_y (/ mekan_x 2.0))))
  (If (Or (Equal h_yon_d 45 0.01) (Equal h_yon_d 90 0.01) (Equal h_yon_d
135 0.01))) (Progn
    (Setq
      giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) giris_x))
      yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (-
mm_y (/ mekan_x 2.0)))
      banmut_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) yatak_x)))
    ))
    (If (Or (Equal h_yon_d 225 0.01) (Equal h_yon_d 270 0.01) (Equal
h_yon_d 315 0.01))) (Progn
      (Setq
        giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- mm_y
(/ mekan_x 2.0)))
        yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) yatak_x))
        banmut_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) giris_x)))
      ))
      (If (Or (Equal h_yon_d 0 0.01) (Equal h_yon_d 180 0.01))) (Progn
        (If (Or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal k_yon_d 135 0.01))) (Progn
          (Setq
            giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- (+ mm_y (/ mekan_x 2.0)) giris_x))
            yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- mm_y (/ mekan_x 2.0)))
            banmut_p (List (+ (- mm_x (/ mekan_y 2.0))
salon_y) (+ (- mm_y (/ mekan_x 2.0)) yatak_x)))
          ))
          (If (Or (Equal k_yon_d 225 0.01) (Equal k_yon_d 270 0.01) (Equal
k_yon_d 315 0.01))) (Progn
            (Setq
              giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- mm_y (/ mekan_x 2.0)))
              yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- (+ mm_y (/ mekan_x 2.0)) yatak_x))
              banmut_p (List (+ (- mm_x (/ mekan_y 2.0))
salon_y) (+ (- mm_y (/ mekan_x 2.0)) giris_x)))
            ))
          ))
          (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_y salon_x)
          (Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_y giris_x)
          (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
          (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_y
banmut_x)
        ))
      ))
    ))
  ))

```

```

      (If (Equal m_yon_d 270 0.01))(Progn
        (Setq salon_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/ mekan_y 2.0))))
        (If (Or (Equal h_yon_d 315 0.01) (Equal h_yon_d 0 0.01) (Equal h_yon_d
45 0.01)))(Progn
          (Setq
            giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
            yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) giris_y))
            banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
          ))
        (If (Or (Equal h_yon_d 135 0.01) (Equal h_yon_d 180 0.01) (Equal
h_yon_d 225 0.01)))(Progn
          (Setq
            giris_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) giris_y))
            yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
            banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
          ))
        (If (Or (Equal h_yon_d 90 0.01) (Equal h_yon_d 270 0.01)))(Progn
          (If (Or (Equal k_yon_d 315 0.01) (Equal k_yon_d 0 0.01) (Equal k_yon_d
45 0.01)))(Progn
            (Setq
              giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
              yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) giris_y))
              banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
            ))
          (If (Or (Equal k_yon_d 90 0.01) (Equal k_yon_d 135 0.01) (Equal k_yon_d
180 0.01) (Equal k_yon_d 225 0.01) (Equal k_yon_d 270 0.01)))(Progn
            (Setq
              giris_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) giris_y))
              yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
              banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
            ))
          ))
        (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
        (Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_x giris_y)
        (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
        (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_x banmut_y)
      ))
    (Princ))

```

- Tasarım Stratejisi M.7. Oturumdan çıkarılan tasarım stratejisi programı

```

(Defun Cizgi_Ciz (pl / p l)
  (If pl (Progn
    (Setq i 0)
    (Command "PLINE")
    (While (Setq p (Nth i pl))
      (Command p)
      (Setq i (1 + i)))
    (Command ""))))
(Defun Yazı_yaz (sl h p x y / sn sb m i sp)
  (Setq
    sn (Length sl) i 0
    sb (/ (* h (1 - (* 4 sn))) 6.0)
    m (List (+ (Car p) (/ x 2.0)) (+ (Cadr p) (/ y 2.0) sb)))
  (Repeat sn
    (Setq sp (List (Car m) (- (Cadr m) (* i (/ (* 4 h) 3.0)) (/ h 2.0))))
    (Entmake
      (List
        (Cons 0 "TEXT")
        (Cons 1 (Nth i sl))
        (Cons 10 sp)
        (Cons 11 sp)
        (Cons 40 h)
        (Cons 72 4)
        (Cons 73 0)))
      (Setq i (1 + i)))
  (Princ))
(Defun Dikdort_Ciz (s h p x y / p2 p3 p4)
  (Setq
    p2 (List (+ (Car p) x) (Cadr p))
    p4 (List (Car p) (+ (Cadr p) y))
    p3 (List (Car p2) (Cadr p4)))
  (Command "PLINE" p p2 p3 p4 "C")
  (Yazı_yaz s h p x y))
(Defun Strj7 (salon_x salon_y yatak_x yatak_y
  banyo_x banyo_y mutfak_x mutfak_y
  m_yon_d h_yon_d k_yon_d yazı_yuk /
  mekan_merkez mm_x mm_y mekan_x mekan_y mekan_x2 mekan_y2
  salon_pl giris_p yatak_p banyo_p mutfak_p
  salon_px yatak_px salon_py yatak_py salon_tp giris_tp)
  (Setvar "LIMCHECK" 0)
  (Setq
    mekan_merkez (Getvar "VIEWCTR")
    mm_x (Car mekan_merkez) mm_y (Cadr mekan_merkez)
    mekan_x salon_x mekan_y (+ (/ mutfak_y 2.0) salon_y)
    mekan_x2 (/ mekan_x 2.0) mekan_y2 (/ mekan_y 2.0)
    salon_k (/ yatak_y 2.0)
    salon_u (- salon_x yatak_x banyo_x mutfak_x salon_k))
  (If (Equal m_yon_d 0 0.01) (Progn
    (Setq
      salon_px (+ (- mm_x mekan_y2) (/ mutfak_y 2.0))
      yatak_px (- mm_x mekan_y2)
      salon_tp (List (+ salon_px (/ mutfak_y 2.0)) (- mm_y mekan_x2)))
    (If (Or (Equal h_yon_d 45 0.01) (Equal h_yon_d 90 0.01) (Equal h_yon_d
135 0.01))) (Progn

```

```

(Setq
  salon_pl (List
    (List salon_px (+ (- mm_y mekan_x2) salon_k))
    (List salon_px (- mm_y mekan_x2))
    (List (+ mm_x mekan_y2) mm_y)
    (List salon_px (+ mm_y mekan_x2))
    (List salon_px (- (+ mm_y mekan_x2) salon_u)))
  yatak_p (List yatak_px (+ (- mm_y mekan_x2) salon_k))
  banyo_p (List yatak_px (+ (- mm_y mekan_x2) salon_k
yatak_x))
  mutfak_p (List yatak_px (+ (- mm_y mekan_x2) salon_k
yatak_x banyo_x))
  giris_tp (List salon_px (- (+ mm_y mekan_x2) salon_u))
  giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 2) 3.0))
))
(If (Or (Equal h_yon_d 225 0.01) (Equal h_yon_d 270 0.01) (Equal
h_yon_d 315 0.01)))(Progn
  (Setq
    salon_pl (List
      (List salon_px (+ (- mm_y mekan_x2) salon_u))
      (List salon_px (- mm_y mekan_x2))
      (List (+ mm_x mekan_y2) mm_y)
      (List salon_px (+ mm_y mekan_x2))
      (List salon_px (- (+ mm_y mekan_x2) salon_k)))
    yatak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u))
    banyo_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x))
    mutfak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x banyo_x))
    giris_tp (List salon_px (- mm_y mekan_x2))
    giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 3) 2.0))
  ))
(If (Or (Equal h_yon_d 0 0.01) (Equal h_yon_d 180 0.01)))(Progn
  (If (Or (Equal k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal
k_yon_d 135 0.01)))(Progn
    (Setq
      salon_pl (List
        (List salon_px (+ (- mm_y mekan_x2)
salon_k))
        (List salon_px (- mm_y mekan_x2))
        (List (+ mm_x mekan_y2) mm_y)
        (List salon_px (+ mm_y mekan_x2))
        (List salon_px (- (+ mm_y mekan_x2)
salon_u)))
        yatak_p (List yatak_px (+ (- mm_y mekan_x2)
salon_k))
        banyo_p (List yatak_px (+ (- mm_y mekan_x2)
salon_k yatak_x))
        mutfak_p (List yatak_px (+ (- mm_y mekan_x2)
salon_k yatak_x banyo_x))
        giris_tp (List salon_px (- (+ mm_y mekan_x2)
salon_u))
        giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 2)
3.0))
    ))
  ))

```

```

))
(If (Or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 225 0.01) (Equal k_yon_d 270 0.01) (Equal k_yon_d 315 0.01)) (Progn
  (Setq
    salon_pl (List
      (List salon_px (+ (- mm_y mekan_x2) salon_u))
      (List salon_px (- mm_y mekan_x2))
      (List (+ mm_x mekan_y2) mm_y)
      (List salon_px (+ mm_y mekan_x2))
      (List salon_px (- (+ mm_y mekan_x2) salon_k)))
    yatak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u))
    banyo_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x))
    mutfak_p (List yatak_px (+ (- mm_y mekan_x2) salon_u
yatak_x banyo_x))
    giris_tp (List salon_px (- mm_y mekan_x2))
    giris_tx (/ (* salon_u 2) 3.0) giris_ty (/ (* salon_u 3) 2.0))
  ))
))
(Prompt "\nsalon_x :")(Princ salon_x) (Prompt "\nsalon_y :")(Princ
salon_y)
(Prompt "\nyatak_x :")(Princ yatak_x) (Prompt "\nyatak_y :")(Princ yatak_y)
(Prompt "\nbanyo_x :")(Princ banyo_x) (Prompt "\nbanyo_y :")(Princ
banyo_y)
(Prompt "\nmutfak_x :")(Princ mutfak_x) (Prompt "\nmutfak_y :")(Princ
mutfak_y)
(Prompt "\nmekan_x :")(Princ mekan_x) (Prompt "\nmekan_y :")(Princ
mekan_y)
(Prompt "\nsalon_px :")(Princ salon_px) (Prompt "\nyatak_px :")(Princ
yatak_px)
(Prompt "\nsalon_pl :")(Princ salon_pl)
(Cizgi_Ciz salon_pl)
(Yazi_Yaz (List "SALON") yazi_yuk salon_tp (/ (* salon_y 2) 3.0) salon_x)
(Yazi_Yaz (List "GIRIS") yazi_yuk giris_tp giris_tx giris_ty)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
(Dikdort_Ciz (List "BANYO") yazi_yuk banyo_p banyo_y banyo_x)
(Dikdort_Ciz (List "MUTFAK") yazi_yuk mutfak_p mutfak_y mutfak_x)
))
(If (Equal m_yon_d 90 0.01) (Progn
  (Setq
    salon_py (- mm_x mekan_x2) (+ (- mm_y mekan_y2) (/ mutfak_y
2.0))
    yatak_px (- mm_x mekan_y2)
    salon_tp (List (- mm_x mekan_x2) (+ salon_py (/ mutfak_y 2.0)) (-
mm_y mekan_x2)))
  (If (Or (Equal h_yon_d 315 0.01) (Equal h_yon_d 0 0.01) (Equal h_yon_d
45 0.01)) (Progn
    (Setq
      giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- mm_y (/
mekan_y 2.0)))
      yatak_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
      banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (-
mm_y (/ mekan_y 2.0))))
    ))
  ))

```

```

    ))
    (If (Or (Equal h_yon_d 135 0.01) (Equal h_yon_d 180 0.01) (Equal
h_yon_d 225 0.01)))(Progn
      (Setq
        giris_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/
mekan_y 2.0)))
        yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- mm_y
(/ mekan_y 2.0)))
        banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (-
mm_y (/ mekan_y 2.0))))
    ))
    (If (Or (Equal h_yon_d 90 0.01) (Equal h_yon_d 270 0.01)))(Progn
      (If (Or (Equal k_yon_d 315 0.01) (Equal k_yon_d 0 0.01) (Equal
k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal k_yon_d 270 0.01)))(Progn
        (Setq
          giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (-
mm_y (/ mekan_y 2.0)))
          yatak_p (List (- mm_x (/ mekan_x 2.0)) (-
mm_y (/ mekan_y 2.0)))
          banmut_p (List (+ (- mm_x (/ mekan_x 2.0))
yatak_x) (- mm_y (/ mekan_y 2.0))))
        ))
      (If (Or (Equal k_yon_d 135 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 225 0.01)))(Progn
        (Setq
          giris_p (List (- mm_x (/ mekan_x 2.0)) (-
mm_y (/ mekan_y 2.0)))
          yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x)
(- mm_y (/ mekan_y 2.0)))
          banmut_p (List (+ (- mm_x (/ mekan_x 2.0))
giris_x) (- mm_y (/ mekan_y 2.0))))
        ))
      ))
    (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
    (Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_x giris_y)
    (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
    (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_x
banmut_y)
  ))
  (If (Equal m_yon_d 180 0.01)))(Progn
    (Setq salon_p (List (- mm_x (/ mekan_y 2.0)) (- mm_y (/ mekan_x 2.0))))
    (If (Or (Equal h_yon_d 45 0.01) (Equal h_yon_d 90 0.01) (Equal h_yon_d
135 0.01)))(Progn
      (Setq
        giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) giris_x))
        yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (-
mm_y (/ mekan_x 2.0)))
        banmut_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) yatak_x)))
      ))
    (If (Or (Equal h_yon_d 225 0.01) (Equal h_yon_d 270 0.01) (Equal
h_yon_d 315 0.01)))(Progn
      (Setq

```

```

        giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- mm_y
(/ mekan_x 2.0)))
        yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (- (+
mm_y (/ mekan_x 2.0)) yatak_x))
        banmut_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y) (+ (-
mm_y (/ mekan_x 2.0)) giris_x)))
    ))
    (If (Or (Equal h_yon_d 0 0.01) (Equal h_yon_d 180 0.01)) (Progn
        (If (Or (Equal k_yon_d 0 0.01) (Equal k_yon_d 180 0.01) (Equal
k_yon_d 45 0.01) (Equal k_yon_d 90 0.01) (Equal k_yon_d 135 0.01)) (Progn
            (Setq
                giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- (+ mm_y (/ mekan_x 2.0)) giris_x))
                yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- mm_y (/ mekan_x 2.0)))
                banmut_p (List (+ (- mm_x (/ mekan_y 2.0))
salon_y) (+ (- mm_y (/ mekan_x 2.0)) yatak_x)))
            ))
        (If (Or (Equal k_yon_d 225 0.01) (Equal k_yon_d 270 0.01) (Equal
k_yon_d 315 0.01)) (Progn
            (Setq
                giris_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- mm_y (/ mekan_x 2.0)))
                yatak_p (List (+ (- mm_x (/ mekan_y 2.0)) salon_y)
(- (+ mm_y (/ mekan_x 2.0)) yatak_x))
                banmut_p (List (+ (- mm_x (/ mekan_y 2.0))
salon_y) (+ (- mm_y (/ mekan_x 2.0)) giris_x)))
            ))
        ))
    (Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_y salon_x)
    (Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_y giris_x)
    (Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_y yatak_x)
    (Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_y
banmut_x)
    ))
    (If (Equal m_yon_d 270 0.01) (Progn
        (Setq salon_p (List (- mm_x (/ mekan_x 2.0)) (- mm_y (/ mekan_y 2.0))))
        (If (Or (Equal h_yon_d 315 0.01) (Equal h_yon_d 0 0.01) (Equal h_yon_d
45 0.01)) (Progn
            (Setq
                giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
                yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) giris_y))
                banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
            ))
        (If (Or (Equal h_yon_d 135 0.01) (Equal h_yon_d 180 0.01) (Equal
h_yon_d 225 0.01)) (Progn
            (Setq
                giris_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) giris_y))
                yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))

```

```

banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
))
(If (Or (Equal h_yon_d 90 0.01) (Equal h_yon_d 270 0.01)) (Progn
(If (Or (Equal k_yon_d 315 0.01) (Equal k_yon_d 0 0.01) (Equal k_yon_d
45 0.01)) (Progn
(Setq
giris_p (List (- (+ mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
yatak_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y
(/ mekan_y 2.0)) giris_y))
banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
))
(If (Or (Equal k_yon_d 90 0.01) (Equal k_yon_d 135 0.01) (Equal k_yon_d
180 0.01) (Equal k_yon_d 225 0.01) (Equal k_yon_d 270 0.01)) (Progn
(Setq
giris_p (List (- mm_x (/ mekan_x 2.0)) (- (+ mm_y (/
mekan_y 2.0)) giris_y))
yatak_p (List (- (+ mm_x (/ mekan_x 2.0)) yatak_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y))
banmut_p (List (+ (- mm_x (/ mekan_x 2.0)) giris_x) (- (+
mm_y (/ mekan_y 2.0)) giris_y)))
))
))
(Dikdort_Ciz (List "SALON") yazi_yuk salon_p salon_x salon_y)
(Dikdort_Ciz (List "GIRIS") yazi_yuk giris_p giris_x giris_y)
(Dikdort_Ciz (List "YATAK" "ODASI") yazi_yuk yatak_p yatak_x yatak_y)
(Dikdort_Ciz (List "BANYO" "MUTFAK") yazi_yuk banmut_p banmut_x banmut_y)
))
(Princ))

```

- Tasarım Stratejisi M.8. Oturumdan çıkarılan tasarım stratejisi programı

BÖLÜM 5. SONUÇ

Amaçlanan tezin hedefine ulaşması için aracın uzman sistem olmasına karar verilmiş ve uzman sistemler hakkında ön inceleme yapılmıştır. İncelenen uzman sistemlerden kural tabanlı sistemler tezin işleyişine uygun bulunmuştur.

Yazlık konutun nitelikleri ve tasarımına yönelik yapılan literatür incelemelerinde konutla ilgili kabuller yapılmıştır. Tasarımı ile ilgili iki adet tasarım stratejisi çıkarılmıştır.

Literatürden çıkarılan kurallar ve kısıtlamalar, ilk aşamada uzman mimardan bilgi alımı için yapılacak oturum örneğinin tasarımında kullanılmıştır. Elde edilen bilgilere göre oluşturulan örnek arsa ve koşullar altında oturum yapılmıştır. Oturum sonucunda iki adet belirlenen örneğe tam uyan tasarım stratejisi çıkartılmıştır. Çıkarılan diğer 6 örnek uzman mimarın oturum sırasında evin biçimsel kuralları ve mekan sayısının sabit tutulmasını kaldırarak farklı oluşumların incelenmesinin uygun bulmasıyla oluşturulmuştur. Oluşturulan bu örneklerle farklı tasarım stratejileri de elde edilmiştir.

Oturumun ardından, girişinde bir mekan olarak kabul edilebileceğine, konut formunda üçgen ve mekan formunda daireninde kullanılabileceğine karar verilmiştir. Bu kısıtlamaların kaldırılmasıyla tasarımda farklılıklara da yer verme imkanı oluşmuştur.

Literatür ve oturumdan elde edilen örnekler daha sonra video'yla saptanan oturum ve oturumda kullanılan eskizlerden tasarım stratejilerinin tek tek kuralları çıkarılmıştır. Çıkarılan bu kurallar

stratejilere ayrılarak oluşturulan örneklerin tasarım kuralları çıkarılmıştır.

Çıkarılan tasarım stratejilerinin kuralları sayesinde oluşturulan uzman sistemin tasarım kuralları elde edilmiştir.

Oluşturulan uzman sistem programında kullanıcı mimarlık öğrencisinin tasarım stratejilerinde kullanılan yönler (Manzara, kuzey ve hakim rüzgar), alan ve boyutları (arsa, konut) nasıl girmesine de karar verildikten sonra uzman sistemin programlanmasına başlanmıştır.

Sonuçta yaratılan paket programda toplam 5 adet tasarım stratejisi kullanılmıştır. Oluşturulan tasarım stratejilerinin tümünün de bilgisayara aktarılabilir olmasına rağmen aktarılmaması bu çalışmanın deneysel olması ve metot yerleştikten sonra örneklerin çoğaltılabilir olmasıdır.

Bu çalışmanın hedef kullanıcısı mimarlık öğrencisidir. Mimarlık öğrencileri bu çalışmayı 'Yazlık Konut' tasarımı ile ilgili proje çalışmalarına başlamadan önce bir uzman mimarın tasarıma nasıl başladığını ve mekan yönlendirmesini nasıl yaptığını anlamak için kullanabilirler.

Bu araştırma deneysel boyutlar ve bazı kabuller altında oluşturulduğu için mimarlık mesleğinde şu anda gerçek bir kullanıma sahip olması çok zordur. Fakat bu çalışma mimarlık öğrenimi gören öğrencilere bir ve/veya ikinci proje derslerinde tasarıma nasıl başlanabileceği, mekanlara hangi oranlarda alan, boyut ve yönlerin uygun olabileceğini görmeleri açısından yararlıdır. Ayrıca programın değişen yönler ve metrekaireler karşısında tasarımı yenilemesi mimarlık öğrencilerine şartlar değiştikçe olasılıkların ne olabileceğini göstermesi açısından da yararlı olabileceği düşünülmektedir.

ZEUS ile řu anda yazlık konutta beř adet tasarım stratejisi programlanmıřtır. Bu sayı ileri arttırılabilir. Fakat bu tezin geliřtirilme yöntemi bu olmamalıdır. Tez bařlangıcında belirlenen kısıtlamaların ve kabullerin kaldırılmasıyla uzmanla daha detaylı ve olası bütün řartların incelenmesiyle elde edilecek tasarım stratejileriyle geliřtirilmelidir. Tezin konusu yine 'Yazlık Konut' olabilir. Ancak arsa, mekan sayısı, konut alanı, mekan ve konut formu ve kat sayısı gibi kabuller ve kısıtlamalar kaldırılarak mimarın tasarımını etkileyen etkenler daha detaylı olarak uzman sistem içine alınmalıdır. Yukarıda belirtilen geliřmeler uzman sistem üzerinde yapıldığında mimarlarında kullanabileceęi tasarıma destek veren bir paket oluřturabileceęi düşünölmektedir.

Eęer tez bu yönde geliřtirilirse dięer disiplinlerin de katılımıyla arsadan projeyi alıp uygulama projesine kadar kullanıcı desteęiyle ilerleyen bir sistem yaratmak mümkün olabilir. Proje sadece plan olarak ele alınmaz aynı zamanda cephe etütleri ve mekan tefriř alternatifleri sunabilir. Buna örnek olarak Manolya Kavaklı'nın İ.T.Ü. Mim. Faköltesi'nde verdięi Yüksek Lisans tezinde oluřturduęu 'Mutfak Tasarımı için Uzman Sistem' programı verilebilir. Geliřtirilen uzman sistem, mutfak için uzman sistem programıyla birleřtirilerek mutfakların tefriřleri de yapılabilir. Bu metotla oluřturulacak banyo, salon, birinci yatak odası ve ikinci yatak odası gibi alt uzman sistemlerle desteklenirse oluřturulacak tasarım sistemi gerek piyasada gerekse öęrenimde daha yaygın olarak kullanıcılara hitap edebilir.

Tabii bu sistemde kullanılacak araçlar ve metotlar farklılařmak zorundadır. Bu sistemlerde mevcut kullanılan programlama dilleri yetersiz veya sistemin geliřmesini engelleyecek kısıtlamalar çıkaracaktır. Bu yüzden farklı disiplinlerden örneęin bilgisayar programcılığı, bilgi mühendislięi, inřaat ve tesisat mühendislięi vb. konularda uzmanlařmıř yüksek düzeyde denegime ve bilgi sahip personelle çalışılmalıdır.

Bu çalışmanın ileride yapılabilecek çalışmalara yol gösterici ve mimarlık öğrencilerine destek verebilecek olmasından dolayı yararlı bir çalışma olduğuna düşüncesi benimsenmiştir.



KAYNAKLAR

[1] ROSENMAN, M. A., GERO, J. S., OXMAN, R. E., (July 1991) What's in a Case, The use of case bases and databases in design. CAAD Futures, Zürich.

[2] GERO, J. S. (1987) Prototypes; a New Schema for Knowledge-based Design *Working Paper Architectural Computing Unit*, Department of Architectural Science, University of Sydney.

[3] REISBECK, C. K., SCHANK, R. C. (1989), Inside Case-based Reasoning, Lawrence Erlbaum, Hillsdale, New Jersey.

[4] HUTCHINSON, P. J. ROSEMAN, M. J., GERO, J. S. (1987), Retwall: An Expert System for Selection of Preliminary Design of Earth Retaining Structures, *Knowledge Based Systems*.

[5] RIESBECK, C. K., SCHENK, R. C. (1988), *Inside Case-based Reasoning*, Lawrence Erlbaum, Hillsdale, New Jersey

[6] WIIG, K., (1990) Expert Systems, A Manager's Guide, International Labour Office, Geneva

[7] WATERMAN, Donald A., (1986) A Guide to Expert Systems, Addison Wesley Publishing Comp. Inc. California

[8] BAYAZIT, N., KAVAKLI, M., (1992 Mart-Nisan) 'Uzman Sistemler', 'Bilgisayar Magazin'

[9] ANDERSON, J. R., (1981) Cognitive Skills and their Acquisition, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[10] LENAT, D. B., (1983) Theory formation by heuristic search. The nature of heuristics II: background and examples, AI 21 31-60

[11] MITCHEL, T. M., KELLER, R., KEDAR-CABELLI, S. (1986).
Explanations Based Generalisations: A unifying view, Machine Learning,

[12] AutoLISP Release 10, (1988) Programmer's Reference,
Autodesk AB, Sweden.

[13] AutoCAD Release 12, (1992) Programmer's Reference,
Autodesk BV, Switzerland.

[14] AutoCAD Release 12, (1992) Customization Manual,
Autodesk BV, Switzerland.



EKLER

Ek A. Tasarım Stratejisi A ve B için, Mekan Alanları ve Kenar Uzunlukları Oranları

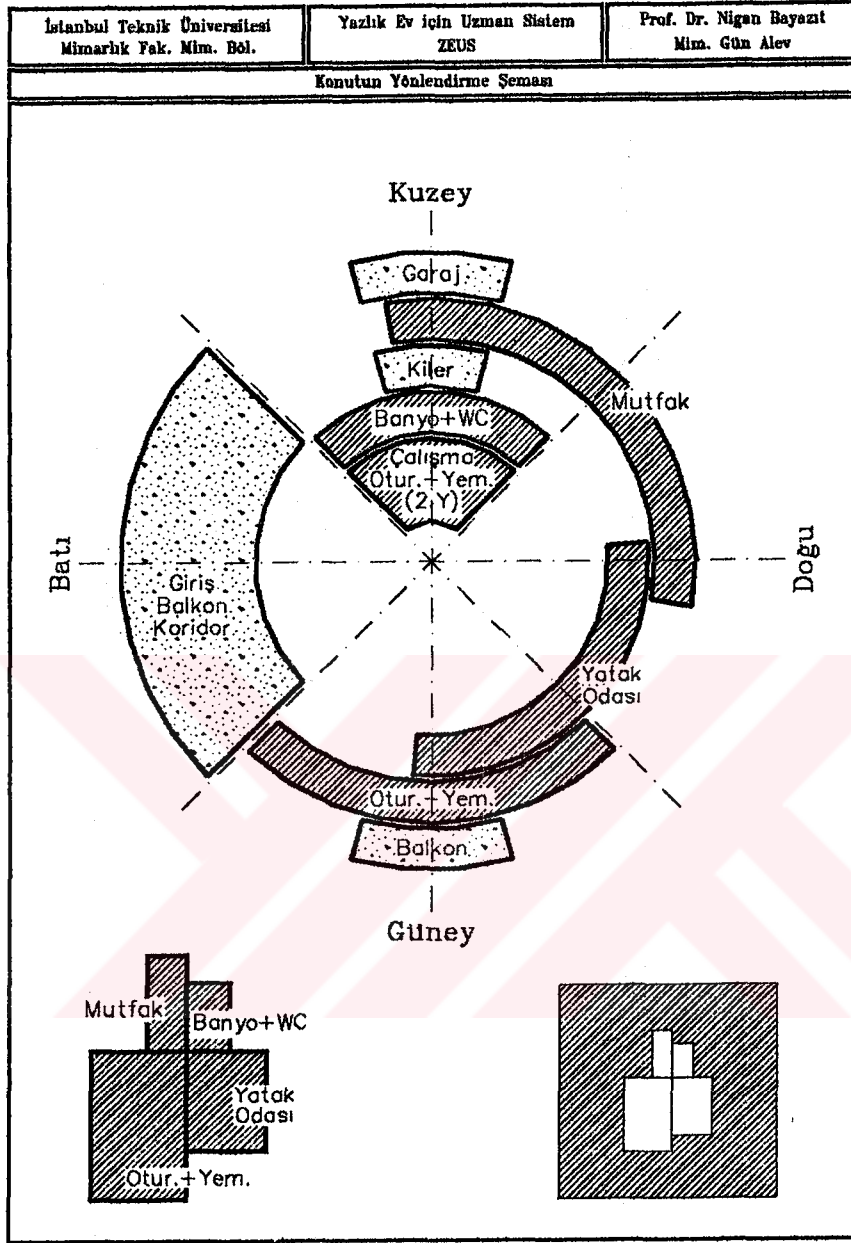
Yazlık Konut'un sahip olduğu mekanların alanların; toplam ev alanına oranının bulunması için; Ülkemizde gerçekleştirilmiş örnekler incelenmiş ve yazılı literatürden ölçümler yapılarak hazırlanmıştır. Evlerin tek tek alanlarını ve bu alanlardan çıkarılan oranları aşağıdaki tablodan bulabilirsiniz. Mekan oranları ve uzunluk oranlarını aşağıdaki 'Evin Toplam Mekan ve Kenar Alanları Oranları' Tablosunda bulabilirsiniz. Bu tablo hesaplamalar sonucu Literatür örneklerinden çıkarılan ortalama oranlardır.

Tablo A.1. Konutun Strateji A ve B için, Toplam Mekan ve Kenar Alanları Oranları.

	Salon	Mutfak	Banyo	Yatak Odası
Mekanin oranı	0.4918	0.12915	0.10211	0.27694
1. Kenar uzunluğu	1.59	2.46	1.60	1.29

Evin net alanı bir metrekare olarak kabul edilmiştir. Birinci Kenar uzunluğu Mekan alanının bu orana bölünerek bulunur. İkinci Kenar uzunluğu; Mekan alanının birinci Kenar uzunluğuna bölünmesiyle çıkarılır.

Ek B. Konut Yönlendirme Şeması



Ek C. Programlama Dilleri Klavuzu

AutoLisp

Komutlar ve Kullanımları

* [] parantez içine gerekirse yazılan argüman veya değişkenler yazılır.

* DEĞİŞKEN : O programda kullanılan Local değişkenler. Programın işi bittiğinde bellekten bu değerler silinebiliyor.

Sabitler ve Değişkenler

Türü	Açıklama	Örnekler
1. Tamsayılar	Inti	
2. Gerçek Sayılar	Real, 14 Hane	2.33 , 0.7354
3. Katarlar	Strings, 100 Karakter	
4. Listeler	Lists	(x,y)

AutoLisp Fonksiyonları

Matematik Fonksiyonlar

Komut	Açıklama
(abs x)	mutlak değeri X
(sqrt x)	karekök X
(expt x p)	X üzeri P
(log x)	doğal log X
(sin a)	
(cos a)	
(atan a)	tanjant A
(fix x)	gerçek X'i tamsayıya çeviriyor. Ondalık kısmı kesiliyor.
(rem a b)	A / B nin kalanı
(gcd (SA1) (SA2))	İki tam sayının en büyük ortak bölenini bulur.
(float (numara))	Tam sayıyı Gerçeğe çevirir
(max (a b c ...))	a, b, c, ... nin en büyüğü
(min (a b c ...))	a, b, c, ... nin en küçüğü
(zerop (öge))	Eğer öge tamsayı ve gerçek ise sonuç T değilse nil olur.
(minusp (öge))	Eğer değer gerçek veya tamsayı ve negatif ise sonuç T değilse nil olur
(numberp (öge))	Eğer değer gerçek veya tamsayı ise sonuç T değilse nil olur
(not (öge))	Eğer öge nil ise sonuç T değilse nil olur
if ve mat. fonkları	=, eq, equal, <, >, <=, >=, /= (eşit değil)
(and (fonk1) (fonk2) ...)	Bütün koşullar gerçekleşirse
(or (fonk1) (fonk2) ...)	Koşullardan sadece biri gerçekleşirse

Katar Fonksiyonları

Fonksiyon	Açıklama	Örnek
(strlen S)	Katarın karakter sayısını verir.	(strlen "AHMET") - 5
(strcat S1 S2 ...)	Katar birleştirme fonk.	
(substr S i b)	Alt katar fonk. I=başlangıç karakteri, B=kaç karakter okunacağı	(substr "SALIM" 2 3) - "LI"

(strcase S M)	Verilen katar büyük veya küçük harflere çevirir	(strcase "Ali") - "ALI" (strcase "Ali" T) - ali
(subst (Yeni) (Eski) (liste))	Listedeki Elemanların eskilerinin yerine yeni elemanları yerleşmede kullanılır.	Sf. 60
(itoa i)	Tam sayıyı katarla dönüştürür.	
(atoi S)	Katarı tam sayıya dönüştürür.	
(atof S)	Katarı gerçel sayıya dönüştürür.	
(rtos X mod K)	Gerçel Sayıyı katarla dönüştürür. mod tipi: 1 Decimal, 2 Mühendislik	(rtos 17.5 1 4) - "17.5000"
(chr i)	ASCII kodu i olan karakter	(chr 65) - "A"
(ascii c)	"c" karakterinin ASCII kodu	(ascii "A") - 65

Giriş Fonksiyonları (getxxx * (ek argümanlar) [ileti])

Fonksiyon	Açıklama
getint	Tam sayı girişi için
getreal	gerçel sayı girişi için
getsring [l]	Katar girişi için kullanılır l= içinde boşluklar olan katarlarda alabilmek için
(getkword (mesaj))	Initget ile kullanılır. katar olarak şifre anahtarı okur Sf.41
getpoint [P1]	Nokta koordinatlarını almak için
getcorner [P1]	Nokta koordinatlarını almak için
getvar	Sayısal sistem değişkenlerini okumak için
(getenv (sys var nm))	Katar sistem değişkenlerini okumak için
setvar	Sistem değişkenlerine değer atamak için
(initget (şart) (katar))	GETxxx fonk.larında sınırlamalar getirir (getstring ve getvar dışında) 1= Boş yanıt iptal 2= Sıfır yanıtı iptal 4= Negatif yanıt iptal 16= 2B noktaları 3B noktalara çevir 32= getpoint ve getcorner'daki çizgileri kesik çiz

Geometrik Fonksiyonlar

Fonksiyon	Açıklama
distance P1 P2	İki nokta arası mesafe
(angtos Aç Mode Birim)	radıyan açığı belirtilen formata çevirir Mode: 0 - Derece, 1 - Derece, Dakika, Saniye, 2- Grad, 3- Radyan, 4 - Surveyor's unit.
angle P1 P2	İki noktadan geçen doğrunun X eksenine ile yaptığı açı
(polar P1 Aç Uzunluk)	@ mesafe, açı ile nokta bulma
(getangle (nokta) (mesaj))	Radyan ile iki noktanın oluşturduğu açığı sorar Sf.39
(osnap P1 (mode))	Bilinen P1 noktasını Snap araçları ile yeniden yakalamak için kullanılır (setq P2 (osnap P1 "midp"))
inters P1 P2 P3 P4 [nil]	P1, P2 ve P3, P4 noktaları ile tanımlanmış iki doğrunun kesişim noktasının bulunması

Liste ve Sıra Fonksiyonları

Fonksiyon	Açıklama
(list A1 A2 A3 ...)	(A1 A2 A3 ...) listesini oluşturuyor.
(cons A liste)	Listenin başına A elemanını ekliyor
(append L1 L2 ...)	L1 ve L2 ... listelerini birleştiriyor
Mevcut Listedeki yararlanma fonksiyonları	
(car liste)	Listenin ilk elemanını alıyor
(cdr liste)	İlk elemandan sonra geriye kalanları alıyor
(cadr liste)	İkinci elemanı alıyor
(cddr liste)	Üçüncü elemanı alıyor
(nth N liste)	N.ci elemanı alıyor
(last liste)	Listenin son elemanını alıyor
(length (liste))	Listedeki eleman sayısı
(listp (değişken))	Eğer değişken liste ise sonuç T değilse nil olur
(mapcar (fonk) (list) (list))	Liste'lerdeki elemanları tek tek fonk'na tabi tutar Sf.48
(reverse (liste))	Listedeki elemanların dizilişlerini ters çevirir.

Bilgi Bankası ve Varlık Fonksiyonları

Fonksiyon	Açıklama	Örnek
(entsel)	Select Object	
(ssget)	Select Objects	
(ssget "X" (filtre listesi))	Bütün çiziminizde varlıkları tiplerine, katmanlarına veya renklerine göre seçebilirsiniz. 0 = Varlık tipi 2= Blok adı için INSERT 6 = Çizgi tipi adı 7 = Metin ve Attribute için STYLE adı 8 = Katman adı 38 = Kod (Gerçek) 39 = Yükseklik 62 = Renk numarası 66 = Attributes follow flag 210 = 3B uzatma doğrultu vektörü (Üç gerçek listesi)	
(sslenght (SS))	Seçim Setindeki varlık sayısı	
(ssname (SS) numara)	Seçim setindeki varlıklara index numarası ile ulaşmak için kullanılır. SS'in ilk varlığın numarası 0 dir, ikincisi 1 numaralıdır.	(setq ENT1 (ssname SSET 0))
(ssadd (varlık adı) (SS))	Seçim setine varlık eklemek için kullanılır	(ssadd N1 SS)
(ssdel (varlık adı) (SS))	Seçim setinden varlık silmek için kullanılır	(ssdel N1 SS)
(ssmemb (varlık adı) (SS))	Bir varlığın Seçim Setinin elemanı olup olmadığını kontrol eder. Eğer Elemanı ise sonuç elemandır, değilse sonuç nil olur.	(ssmemb N1 SS)

(entnext (varlık adı))	Seçim Seti içindeki elemanlara adlarıyla ulaşmamıza yardımcı olur. Sf 70	(setq N1 (entnext)) = çizimdeki ilk varlığı N1'e eşitler (setq N2 (entnext N1)) = N1'den sonra ki varlığı N2'ye atar
(entlast)	Çizimdeki en son varlığı seçer. Sf 71	(setq N1 (entlast)) = Çizimdeki son varlığı N1'e eşitler
(entget (varlık adı))	Varlığı oluşturan bilgi bankasına ulaşır. ASSOC fonk ile ACAD DXF group kodlarıyla varlığın bilgilerine ulaşabilirsiniz.	(setq A (entget (entlast)))
(entdel (varlık adı))	Seçim Setinden bir varlık siler	Sf. 72
(entmod (varlık listesi))	Seçim Setindeki değişikliğin çizim bilgi bankasına yansımaları sağlar.	Sf 75 Bak: Sf 60 Subst
(entupd (varlık adı))	Entmod ile yenilenmiş varlığın ekran görüntüsünde yenilenmesini sağlar	Sf 75

Diğer Fonksiyonlar

Fonksiyon	Açıklama
(menucmd (katar))	Menü içindeki alt menülere gitmeye yarar. S= Ekran menüsü B= Tuşlar menüsü I= Icon menüsü P1 - P10= 1 den 10a kadar PullDown menüleri T1 - T3= 1 den 4e kadar Tablet menüleri A1= AUX1 menüsü (menucmd "P1=Numerik") (menucmd "P1=*")
(progn (fonk1) (fonk2) ...)	If ve cond fonksiyonlarında birden çok fonksiyon tanımlamak için kullanılır

Kütük Fonksiyonları

Fonksiyon	Açıklama
(open F m)	Kütük açma fonksiyonu d=dosyanın adı m= dosya açma modu "r" okuma için açma "w" yeni dosya yaratarak çıktı dosyası açma "a" mevcut dosyayı çıktı için açma (setq KUT (open "dosya.uzt" "r"))
(read-line F)	Kütük okuma (setq S (read-line KUT))
(write-line S F)	Kütüğe yazma S=katar (read-line S KUT)
(close F)	Kütüğü kapatma (close KUT)

Varlık Bilgi Bankası Codeları

Bilgi Kodu	Açıklama
(-2)	"SEQUEND" için ana varlık (Main entity) adı
(-1 . <Entity name: 60000053C>)	Varlık Adı
(0 . "TEXT")	Varlık Cinsi
(1 . "Metin")	Metin
(5 . "7E")	Handle
(6 . "DASHED")	Çizgi tipi
(7 . "STANDARD")	Metin Stili
(8 . "YAZI")	Katman
(10 . 16)	Koordinatlar
(10 . 2.0 2.0 0.0)	Başlangıç Noktası
(11 . 0.0 0.0 3.5)	"TEXT" için Alignment point, "LINE" için son nokta
(38 . 10.0)	Kot (Elevation)
(39 . 2.5)	Yükseklik (Height)
(40 . 0.3)	"TEXT" için Yüksekliği (Height), "ARC" ve "CIRCLE" için yarıçap, "PLINE" için son başlangıç genişliği (width)
(41 . 1.0)	"TEXT" için Genişlik Faktörü, "PLINE" için son uç genişliği, "INSERT" için X ölçeği
(42 . 1.0)	"INSERT" için Y ölçeği
(43 . 1.0)	"INSERT" için Z ölçeği
(50 . 0.523598)	"TEXT" için Eğim açısı (Radyan Olarak) veya Başlangıç açısı
(51 . 0.0)	"TEXT" için Yatıklık Açısı, "PLINE" için Son uç açısı
(62 . 5)	Renk
(66 . 1)	İzleyen kayıtlarda alt varlıklar varsa 1 "PLINE" için "VERTEX" kayıtları "INSERT" için "ATTRIBUTE" kayıtları
(71 . 0)	Yazı üretim tipi (Text Generation factor)
(72 . 0)	Hizalama Katsayısı (Justification)
(210 . 0.0 -0.642788 0.766044)	Uzatı Yönü Vektörü (Extrusion direction vektor)

DCL Programlanabilir Sorgu Kutuları

Sorgu Kutuları Bileşenleri

Tuşlar ve Tip Tuşlar

Button: (in DCL: button)

Label: Tuşun üzerinde gelen metin

is_default: true veya false eşittir. (default false) Eğer doğruysa, entera basıldığında bu tuşa basılmış olarak hissedilir. Bir sorgu kutusunda en fazla bir tane is_default olabilir.

is_cancel: true veya false eşittir. (default false) Eğer ESC veya Control + C ye basılırsa bu tuş harekete geçer. Sorgu kutsunda en fazla bir tane bulunabilir ve bu tuş sorgu kutusu tamamıyla çağrıldığında harekete geçer.

Uyarı: Çıkış için iki tuş yanımlamayın. Exit tuşunun metnini değiştirmek için sf. 167'ye bakınız. Sorgu kutusunun devamlı cancelled kalmasını engellemek için tuşlardan en azından birinin done_dialog komutuyla bağlantısı olmalı onun için sf 187'ye bakınız.

Dialog:

label: sorgu kutusunun sol üst köşesinde bulunan metin.

value:

initial_focus: sf 148

Edit Box: (in DCL: edit_box)

label: tuşun üzerinde yazılan metin

edit_width: reel veya tam sayı olabilir. yazılacak metnin genişliğini sabitler.

edit_limit: (default 132) yazılan metnin genişliğini tanımlar.

value: (default null) edit tuşunun giriş kısmında görülen default metindir.

allow_accept: true veya false'e eşittir. Eğer doğruysa ve kullanıcı Enter'a basarsa "**is_default**"la tanımlanmış tuş harekete geçer.

Image Button: (in DCL: image_button)

color: (default 7) renk atamaları için sf 149'a bakınız.

allow_accept: true veya false'e eşittir. Eğer doğruysa ve kullanıcı Enter'a basarsa "**is_default**"la tanımlanmış tuş harekete geçer.

aspect_ratio: gösterilecek image'in en ve boy oranını gösterir. (genişlik bölü yüksekliktir.) {0.0} verildiğinde image oranlarında gösterilir ve default'da budur.

List Box: (in DCL: list_box)

label: List Box'in üstünde yazılan açıklayıcı yazıdır.

multiple_select: true veya false olabilir. Eğer true ise listeden birden çok eleman seçilebilir.

list: list_box'daki seçeneklerin bulunduğu metindir. Seçenekler birbirlerinden yeni satır işareti ile ayrılırlar. Tab işareti her satırda görülebilir.

tabs: tam sayıları içeren bir metindir ve boşluklarla ayrılmıştır. list'deki tablaların ne kadar mesafede gösterileceğini belirtir. Örneğini acad.dcl'deki acad_mlayer sorgu kutusunda görebilirsiniz.

value: sıfır veya daha fazla tam sayıdan oluşmuş ve sayıların boşlukla ayrıldığı bir metindir. multiple_select'te seçimlerin başlangıçta aldığı index değerleridir. Eğer multiple_select false'sa birden fazla tamsayı içeremez. Eğer value metni boşsa başlangıçta bir liste elemanı seçilmiyor demektir.

allow_accept: true veya false'e eşittir. Eğer doğruysa ve kullanıcı Enter'a basarsa "**is_default**"la tanımlanmış tuş harekete geçer.

Popup List: (in DCL: popup_list)

label: popup list'in solunda yazılan açıklayıcı yazıdır.

edit_width: Listenin genişliğini sabitler eğer değer tanımlanmamış veya sıfırsa listeye göre genişlik değişir.

value: Popup listenin mevcut seçiminin değerini gösteren, sıfırdan başlayan bir tamsayı içeren metindir. liste Popup yapılmadan default durumdaki metini anlatır.

list: popup list'deki seçeneklerin bulunduğu metindir. Seçenekler birbirlerinden yeni satır işareti ile ayrılırlar. Tab işareti her satırda görülebilir.

tabs: tam sayıları içeren bir metindir ve boşluklarla ayrılmıştır. list'deki tablaların ne kadar mesafede gösterileceğini belirtir. Örneğini acad.dcl'deki acad_mlayer sorgu kutusunda görebilirsiniz.

Radio Button: (in DCL: radio_button)

label: Tuşun üzerinde görünen metnindir.

value: Eğer value "1"e eşitse tuş on açılır. Eğer radio setinden birden çok tuş "1" değerine sahipse en son tuş on konumunda açılır.

Radio Column:

value: mevcut seçilmiş radio button'ın değeri ancak "1" olabilir.

Slider: (in DCL: slider)

min_value: (default 0) Kayarın hareket edeceği değerleri tanımlamada kullanılır. Gelebileceği minimum değeri tanımlar.

max_value: (default 10000) Kayarın hareket edeceği değerleri tanımlamada kullanılır. Gelebileceği maksimum değeri tanımlar.

small_increment: Kayarın arttırma değerlerini kontrol eder. Default olarak small_increment hareket değerinin yüzde biridir. Değer min_value ve max_value değerleri arasında olmalıdır.

big_increment: Kayarın arttırma değerlerini kontrol eder. Default olarak big_increment hareket değerinin yüzde biridir. Değer min_value ve max_value değerleri arasında olmalıdır.

layout: Kayar yatay veya düşey olabilir. Default yataydır. Yatay kayarlarda değer soldan sağa artar. Düşey kayarlarda değer alttan üste artar.

value: (default: min_value) kayarın mevcut (tamsayı) değerini tanımlayan metindir.

Toggle: (in DCL: toggle)

label: toggle kutusunun sağında görülen metindir.

value: (default 0)tamsayı içeren bir metindir. Eğer değer sıfırsa kutu boş çıkar eğer değer bir ise kutuda bir çek işareti veya bir X olur.

Karolaj

column, boxed_column, row, boxed_row, radio_column, boxed_radio_column,

radio_row, boxed_radio_row

Dekoratif veya Açıklayıcı Tuşlar

Image: (in DCL: image)

color: (default 7) renk atamaları için sf 149'a bakınız.

aspect_ratio: gösterilecek image'in en ve boy oranını gösterir. (genişlik bölü yüksekliktir.) (0.0) verildiğinde image oranlarında gösterilir ve default'da budur.

Text: (in DCL: text)

label: Tuşun üzerinde görülen metindir.

value: Tuşun yanında görülen metindir. Eğer metin kullanım sırasında değişmiyorsa value değerini tuşun üstüne yazabilirsiniz. Eğer kullanım sırasında metin değişecekse değişim durumlarına uygun width değerini atayıp value tanımlamanızda fayda vardır.

is_bold: true veya false olabilir (default false'dır) Eğer true ise metin kalın karakterlerle gösterilir. Fakat bu bütün bilgisayar ortamları tarafından desteklenmez.

Spacer: (in DCL: spacer) boş satır bırakmak için kullanılır.

DCL dosyaları için Şematik Uyarı seçenekleri

0: Kontrol yapma

1: Hatalar: AutoCAD'de hata oluşturabilecek hataları göster.

2: Uyarılar

3: Hints: hataları bulur ve yardımcı açıklamalar getirir. Yeni başlayanlar için idealdir.

Kullanılışı: dcl_settings : default_dcl_settings { audit_level = 3; }

DCL hatlarını yaklama

DCL dosyasını yüklemek için ;

Command: (load_dialog "hellofile")

3

DCL'i çağırmak için ;

Command: (new_dialog "hello" 3)

Eğer new_dialog'ın sonucu nil ise (start_dialog)'u kullanmayınız. Bu durumda sf 184'de bakmanızda fayda vardır.

(if (new_dialog "hello" 3) (setq retcode (start_dialog)))

Genel Uyarılar

- Tuş genişliklerini fixed_width = true yaparak aynı hiza tutmaya özen gösterin.

- Eğer sorgu kutunuz geçici olarak saklanacaksa, saklama tuşunun sonuna "<" işaretini koymanızda fayda vardır.

Sorgu kutularının AutoLISP ile düzenlenmesi

Sorgu kutusunun düzenlenmesine örnek.

"merhaba.dcl" dosyası

```
hello : dialog {
  label = "Ornek Sorgu Kutusu";
  : text {
    label = "Merhaba, dünya";
  }
  ok_only;
}
```

"merhaba.dcl" dosyasını çalıştırmak için kullanılacak AutoLISP dosyası.

```
(defun showalert (/ dcl_id)
  (setq dcl_id (load_dialog "merhaba.dcl"))
  (if (not (new_dialog "merhaba.dcl" dcl_id))
    (exit))

  (action_tile
    "accept"
    "(done_dialog)")

  (start_dialog)
  (unload_dialog dcl_id)
)
```

DCL sisteminin sırasal prosedürü

1. DCL dosyasını load_dialog ile yükleyin.
2. Sorgu kutusunun AutoCAD'ın grafik ekranına getirmek için new_dialog'u kullanın.
3. Sorgu kutusunun tuş değerlerini, listeleri ve görüntüleri başlatın. Tuş değerleri için set_tile, mode_tile, listeler için start_list, add_list ve end_list, Görüntüler (image) için vector_image, fill_image, slide_image ve end_image komutlarını kullanabilirsiniz.
4. Kullanıcının kontroluna bırakmak için start_dialog'u kullanın.

5. Tanımladığınız hareketlere göre kullanıcı işlemlerini yerine getirir. Bu sırada genellikle `get_tile`, `get_attr`, `set_tile` ve `mode_tile` komutları DCL için geçerlidir.

6. Kullanıcı `done_dialog` komutunu harekete geçmesini sağlayan bir seçim yaptığında, `start_dialog` komutuna geri dönlür. Sonra `unload_dialog` yapılmalıdır.

Standart prosedür aşağıdaki gibidir (iki sorgu kutulu akış).

`load_dialog`

```
new_dialog
  action_tile
start_dialog
  get_tile
  set_tile
done_dialog
```

```
new_dialog
  action_tile
start_dialog
  get_tile
  set_tile
done_dialog
```

`unload_dialog`

AutoLISP Hareket İfadeleri

Değişkenler	Açıklamaları
\$key	Seçilen tuşun key attribute'ı
\$value	Seçilen tuşun mevcut değeri. Bu metin metin kutularında yazı, toggle'da "1" veya "0".
\$data	<code>new_dialog</code> ve <code>client_data_tile</code> komutlarında sonra belirlenen program tarafından belirlenen data
\$reason	Bu değer hareketin neden ortaya çıktığı belirler. fakat bu değeri <code>edit_box</code> , <code>list_box</code> , <code>image_button</code> veya <code>slider</code> 'larla ilgili durumlarda kullanabilirsiniz.
\$x	<code>image_button</code> seçiminde X koordinati.
\$y	<code>image_button</code> seçiminde Y koordinati

Sorgu Kutularını Saklama

Hareket sırasında ki saklama cümlecği

(`action_tile "pick_pt" "(done_dialog 4)"`)

geniş açıklama için sf 195'e bak.

Tuşları organize etme

Modları ve Değerleri Tuşlara Atama

(`set_tile "lastname" name_str`) ; değeri tuşa atama

(`mode_tile "lastname" 2`) ; 2 focusu tuşa sabitler

veya

(`mode_tile "lastname" 3`) ; 3 tuşun değerini seçer.

List Box'ları veya Popup List'leri Ayarlama

- Bir listeyi yaratmak için (AutoLISP için 3)

`start_list` çağrıldıktan sonra, `add_list`'i bir çok kez kullanabilirsiniz. Her `add_list` listeye yeni bir eleman ekler. Listeye ekleme sürecine `end_list` komutu ile bitirebilirsiniz.

- Listedeki bir elemanı değiştirmek için (1)

`start_list`'den sonra, `add_list`'i bir kez çağırın ve `start_list`'e index'i belirlenmiş elemanı değiştirin. Bu süreci `end_list` komutu ile bitirin.

- Listeye bir eleman eklemek için (AutoLISP için 2)

`start_list`'den sonra, `add_list`'i kullanın ve `end_list`'le eklem işini bitirin.

sf. 202

Image'lerin yaratılması

- vector_image

Mevcut image'de bir doğru çizer (tek doğru çizgi).

- fill_image

Mevcut image'de dolu bir dikdörtgen çizer.

- slide_image

Mevcut image'de bir AutoCAD slide'i çizer.

Bu üç komuttada image'in boyutlarını bilmelisiniz. Bunun için (dimx_tile) ve (dimy_tile) komutlarını kullanabilirsiniz. Image'in orijini (0,0) her zaman sol-üst noktasıdır.

Slide gösterimi için örnek:

```
(setq x (dimx_tile "view"))
(setq y (dimy_tile "view"))
(start_image "view")
(slide_image 0 0 x y "topview")
(end_image)
```

Renkler ve daha geniş tanımlama örnekleri için sf. 205-207'ye bakınız.

Image Button girişi

```
(action_tile "image_sel" "(pick_shade $key $value $y)")
```

-

-

-

```
(defun pick_shade (key val y)
  (setq threshold (/ (dimy_tile key) 2)) ; image
  yatay olarak ikiye bölündü.
  (if (> y threshold)
    (setq result "Light")
    (setq result "Dark") )
)
```

Radio Cluster'ların Organizasyonu

```
(action_tile "view_sel" "(pick_view $value)")
```

-

-

-

```
(defun pick_view (which)
  (cond ((= which "front") (setq show_which 0))
        ((= which "top") (setq show_which 1))
        ((= which "left") (setq show_which 2))
        ((= which "right") (setq show_which 3))
  )
)
```

Slider'ların Organizasyonu

```
(action_tile "my_slider" "(slider_action $value $reason)")
```

```
(action_tile "slider_info" "(ebox_action $value $reason)")
```

-

-

-

```
(defun slider_action (val why)
  (if (or (= why 2) (= why 1))
    ( set_tile "slider_info" val)
  )
)
```

)

```
(defun ebox_action (val why)
```

```

        (if (or (= why 2) (= why 1))
            ( set_tile "myslider" val)
        )
    )
Edit Box'ların Organizasyonu
(action_tile "my_slider" "(edit_action $value $reason)")
-
-
-
(defun edit_action (val why)
    (if (or (= why 2) (= why 1))
        ; uzunluk ve deęer kontrolarını burada yapınız.
    )
)

```

Fonksiyonlar

DCL dosyalarını açma ve kapama

```

- (load_dialog dclfile)
- (unload_dialog dcl_id)

```

Sorgu Kutularının açılması ve kapama

```

(new_dialog dlgname dcl_id (action-expression (screen-pt)))
(start_dialog)
(done_dialog (status))
(term_dialog)

```

Action Expression or Callback Fonksiyonlarının hazırlanması

```

(action_tile key action-expression)

```

Tuş ve Attribute'ların düzeni

```

(mode_tile key mode)
(get_attr key attribute)
(get_tile key)

```

Gets the run-time value of the specified tile.

```

(set_tile key value)

```

Sets the run-time value of the specified tile.

List box ve Popup List'lerin tanımlanması

```

(start_list key (operation (index)) )
(add_list item)
(end_list)

```

Image'ların yaratılması

```

(dimx_tile key)
(dimy_tile key)
(start_image key)
(vector_image x1 y1 x2 y2 color)
(fill_image x1 y1 x2 y2 color)
(slide_image x1 y1 x2 y2 slnam)
(end_image)

```

Application-Specific Data

```

(client_data_tile key clientdata)

```

Ek D. UM'a verilen açıklayıcı föyler

FÖY 1 Mimari Tasarlama Eğitiminde Yazlık Konut Tasarımı Amacıyla Kullanılabilecek Bir Uzman Sistem

ZEUS'un Tanımlanması

İ.T.Ü. Mimarlık Fakültesinde Uzman Sistemler (US) üzerinde yapılan çalışmalardan biridir. Amacı Mimarlık Öğrencilerine tasarım aşamasında seçenek sunmak ve deneyimli mimarların konuya nasıl yaklaştığı hakkında bilgi vermektir.

ZEUS'un nasıl olacağı

ZEUS; Literatürden ve konusu üzerinde deneyimli mimarlardan alınacak süreç, aşama ve tarz hakkındaki bilgilerin bilgisayara aktarılmasıyla oluşturulacaktır. Sizden alınan bilgiler sınıflandırılıp, sıralanacaktır. Daha sonra bu bilgiler içerisinde tezde kullanılacak bilgiler ayrılacak ve daha sonra programlamaya uygun formlara dönüştürülecektir. Bu sebepten dolayı, sizden bu tasarım süreci içerisinde yaptığınız, düşündüğünüz her türlü bilgiyi söylemenizi veya çizdiğiniz çizgiyi açıklamanızı rica ediyoruz.

US bilgisayara AutoCAD R12 for Windows paket programı ve AutoLisp, DCL ve C gibi programlama dilleri kullanılarak aktarılacaktır.

FÖY 2, Yazlık Konut Arazi'si ile ilgili Bilgiler:

- Arazinin yeri: Arazi İstanbul yakınlarında deniz kıyısına yakın fakat deniz kenarında değildir.

- Arazinin Alanı: 442 m2. dir. Eni 17m. ve Boyu 26 m. dir.

- Yoldan çekilme 5m., Önden çekilme 10 m. ve Yanlardan çekilme mesafesi 3 m. olarak kabul edilmiştir.

- Evin net alanı minimum 35 m2. olabilir.

- Arazinin KAKS'ı 1, TAKS'ı ise 0.20'dir.

- Arazinin Kuzey Yönü 45, Hakim Rüzgar Yönü 135 ve Manzara Aksı 270 derecedir. (Manzara Aksı; Manzara Yayının açı ortayıdır.)

- Arazi düz ve üzerinde yapılaşmayı etkileyebilecek düzeyde bitki örtüsü yoktur.

- Su, gaz ve kanalizasyon bağlantıları yol cephesinden sağlanmaktadır.

Kullanıcılarla ilgili bilgiler:

- Kullanıcı Aile ortalama 25 yaşında çocuksuz bir çifttir.

- Yazlık evlerinde; oturma, yemek ve dinlenme eylemlerine izin verebilecek bir salon, fazla güneş almayacak bir yatak odası, hem tuvalet hemde yıkanma eylemlerine hizmet edebilecek bir banyo ve açık olmayan bir mutfak istemektedirler.

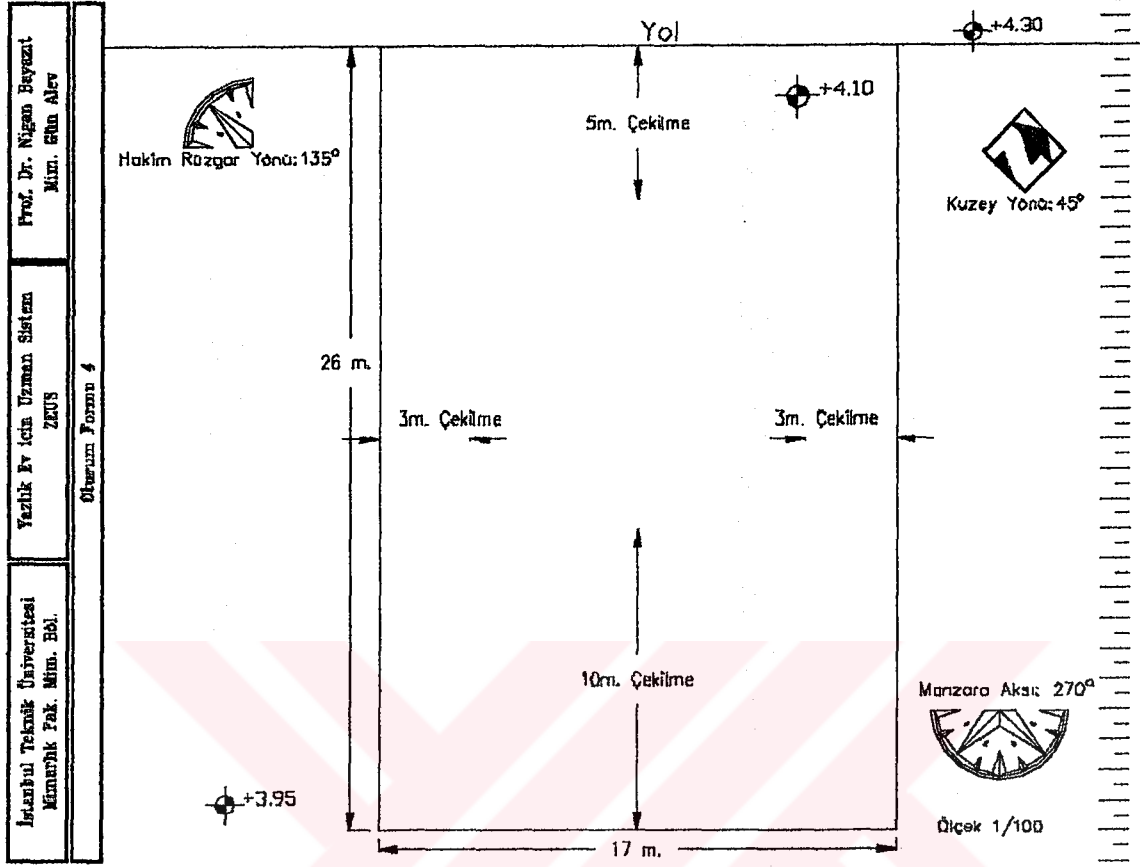
- Ayrıca evin tek katlı olmasını ve mümkün olduğu kadar yer kaybının olmamasını istemektedir.

FÖY 3

İstanbul Teknik Üniversitesi Nispetiye Fak. Mimarlık Bld.	Yazlık Ev İçin Uzman Sistem ZEUS	Prof. Dr. Nigaz Bayazit Mim. Gün Alev
Ölçüm Formu 3		
Adınız: _____ Soyadınız: _____ Ünvanınız: _____ Akademik Eğitim düzeyiniz: _____		
Arazinin Alanı: 442m ² Arazinin Boyu: 26m. Arazinin Eni: 17m. Evin Net Alanı: 45m ² . KAKS: 1 TAKS: 0.20	Kuzey Yönü: 45 der. Hakim Rüzgar Yönü: 135 der. Manzara Akışı: 270 der. Yoldan çekilme mesafesi 5 m ² . Önden çekilme mesafesi 10 m ² . Yonlardan çekilme mesafesi 3 m ² .	
Evin Programı: Salon, Mutfak, Banyo ve 1 adet Yatak Odası. Tek Katlı.		

Şekil D.1. Arsa Krokisi.

FÖY D

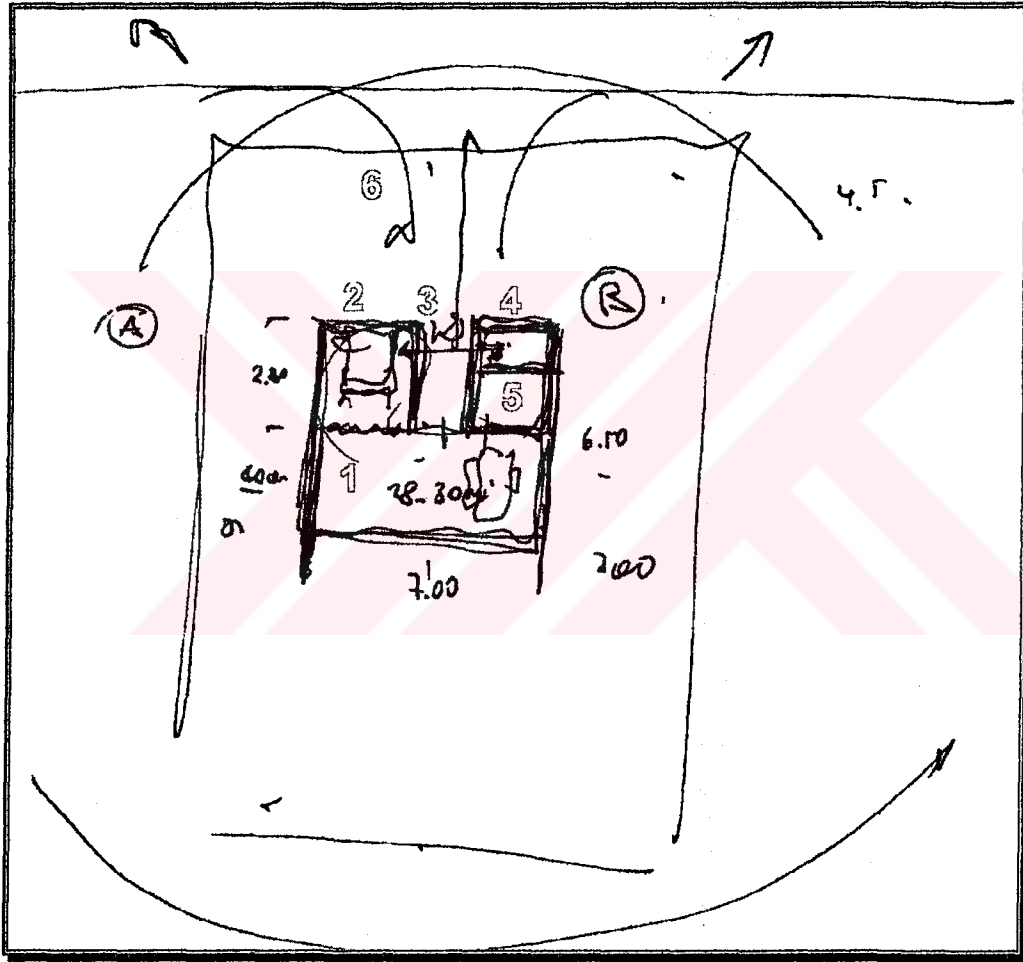


Şekil D.2. Vaziyet Planı.

Ek E. Oturumun Metni ve Yapılan Örnek Tasarımlar

Uzmanla yapılan oturum 3 Kasım 1993'de UM'in ve Tez Yürütücüsü ve tezi hazırlayanın katılımıyla yapılmıştır. Oturumda kullanılan açıklama föyler Ek D'de bulunmaktadır.

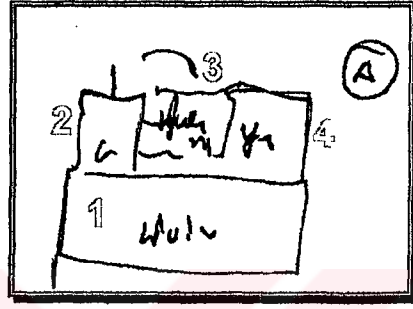
Aşağıdaki metinde UM; UM'ın söylediklerini, TD; Tez danışmanın ve TH tezi hazırlayanın söylediklerini belirtmektedir.



Şekil E.1. Eskiz 1_ 1'inci dikdörtgen salon olarak çizildi. İkinci dikdörtgen yatak odası olarak, 3'üncü dikdörtgen giriş koridoru olarak, 4'üncü dikdörtgen banyo ve 5'inci dikdörtgen mutfak olarak çizildi. TD'n "Konuşmanızı istiyoruz yalnız." demesinden sonra binanın çevresini çizdi. Hakimî RûzîHrîlar, Kuzey Yönü, kanallar ve çekilmeler işlendi. Evin alanından mekanların boyut dağılımı çıkarıldı.

TD - Konuşmanızı istiyoruz yalnız.

UM - Şimdi şöyle Gün Bey anlatırken vaziyet planına baktım. Üçer metre yanlardan çekilme var. Burdan zannediyorum, 10 m. minimum 35 m² olması isteniyor. Hakim rüzTHrlar şöyle geliyor. Burda manzara var. 15 - 20 cm.lik meğil var ,Güney-doğu güney-batı istikametinde. Tesisat kanaları yoldan geliyor ve gelişleri böyle oluyor. Ev basit bir ev planı olarak merkezi bir form teşekkül ediyor. Yatak odasının fazla güneş almaması vs. gibi bir takım istekler var. Manzaraya açık bir salon, kapalı bir mutfak isteniyor ve tek katlı olması şart koşuluyor.



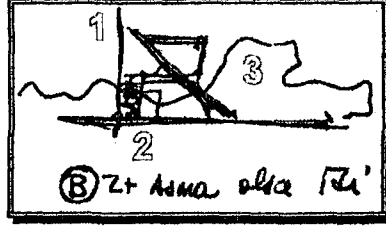
Şekil E.2. Eskiz 2_ 1'inci dikdörtgen salon, 2'inci Dikdörtgen giriş, 3'üncü dikdörtgen dışta banyo, salon tarafında mutfak, 4'üncü dikdörtgen yatak odası.

UM - Bu durumda iki ihtimal doğuyor; şu verilere göre önde yer alan salona bir yerden girdiğinizde bu tarafa göre inkişaf etme imkanınız arkada ıslak hacimler kapalı mutfak veya giriş ortaya gelirse bu ihtimal ortaya çıkıyor. Bunların ikisi arasındaki mukayasede çok ufak farklar var. Bu durumda 45 m²yi tamamlamak üzere(düşünüyor, hesap yapıyor) 8x5 gibi 7x6 gibi olsa 7x6.5 olduğu taktirde bunun ıslakhacimler ve yatak odası için 2.75-2.80 (m) bir aks alırsak, geriye 4 duvarlar hariç kalır. 28 - 30 m². bir oturma odası olur. Buraya mutfak açılır önünde ayrı yemek yeme imkanı var. Yatak odası gerektiğinde açılabilmesi kullanma yüzeyini genişletebilir.

Burada banyo ve tuvaletin karşıya geçmesi evin küçük olmasıyla büyük bir problem doğurmuyor. Bunlara ek bunların ekonomisi ve bağlantısı rahatlıyor.

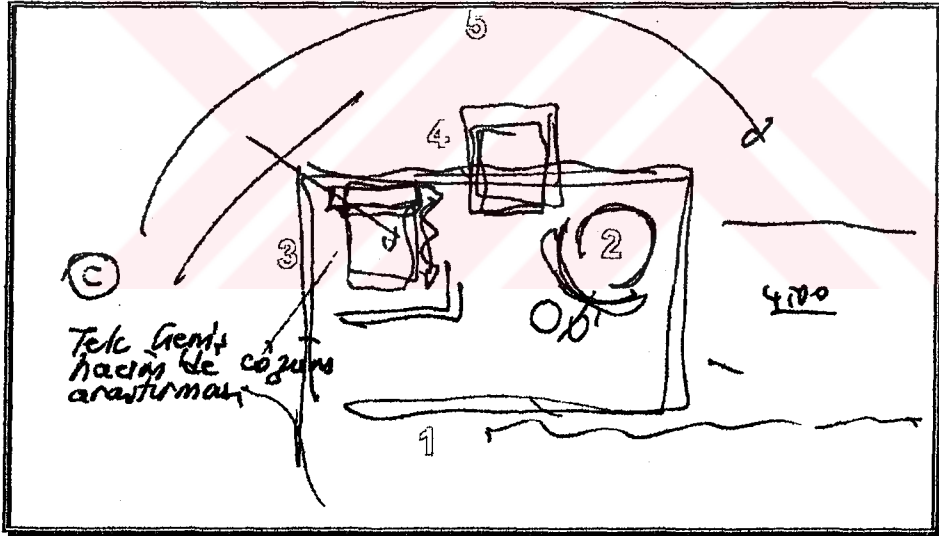
Eğer sıradan arsalar yanyana ise bu yatak odasının burada veya diğer yönde olmasının bir mahsuru yok. Belki burada baş aşağı

gelebilir. buraya gelebilir. Bu bence kullanıcıların isteğine bağlı. En basit şekliyle bu.



Şekil E.3. Eskiz 3_ İlk olarak yukarıdaki yatak odasının taşıyıcısı ve yatak odası çizildi. 2'inci olarak zemin kat çizildi. 3.'üncü olarak doğanın bina içine girişi çizildi.

UM - Eğer katlı olsaydı başka imkanlar çıkabilirdi. Ama öyle bir şey istenmediğine göre, belki o zaman yatak odası yukarı alınırdı. Enteresan bir mimari çıkabilirdi. Buraya kadar cam olan, bunun içinden çıkan aşağıda ıslak hacimlerin olduğu, önünde tümüyle yeşile açılan başka bir form aranabilirdi. Ama burda hadise son derece basit.



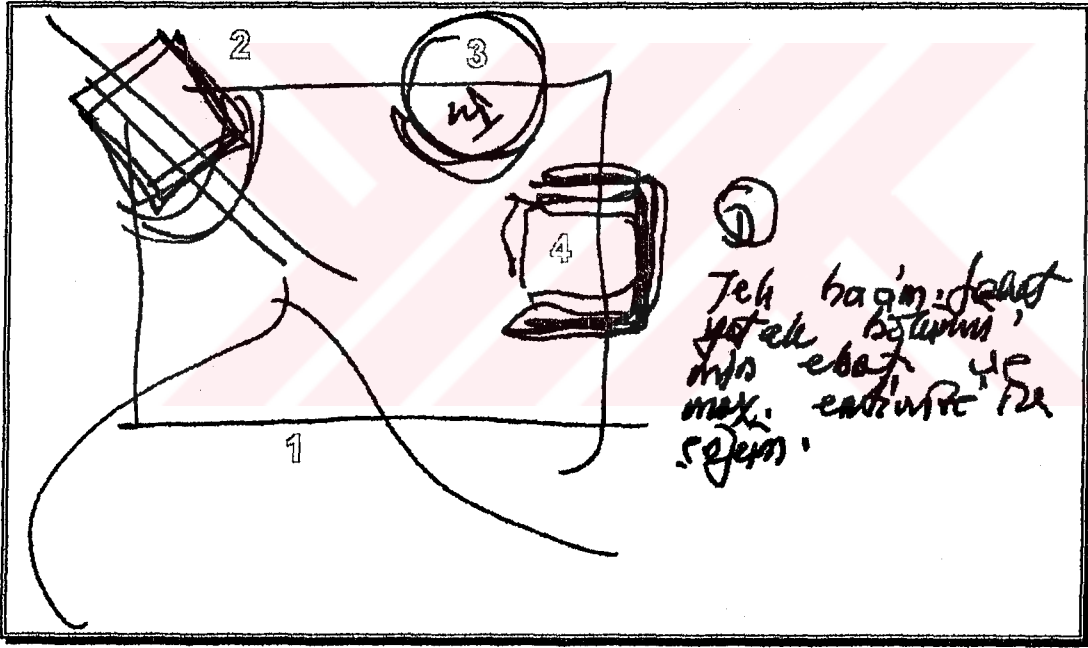
Şekil E.4. Eskiz 4_ 1'inci Dikdörtgen evin sınırlarını belirleyecek şekilde çizildi. 2'inci olarak yarı açık mutfak nişi çizildi. 3'üncü olarak yine yarı açık yatak odası çizildi. 4'üncü olarak giriş rüzTHrlığı çizildi. 5'inci olarak çizilen; tek hacmin dışı daha açık olduğunu belirten bir açılım eğrisi oldu.

UM - Bunun dışında program bu kadar kapalı olamasa; Bunu başka şekillerde halletme imkanı olabilir.

TD - Söyler misiniz yaptıklarınızı.

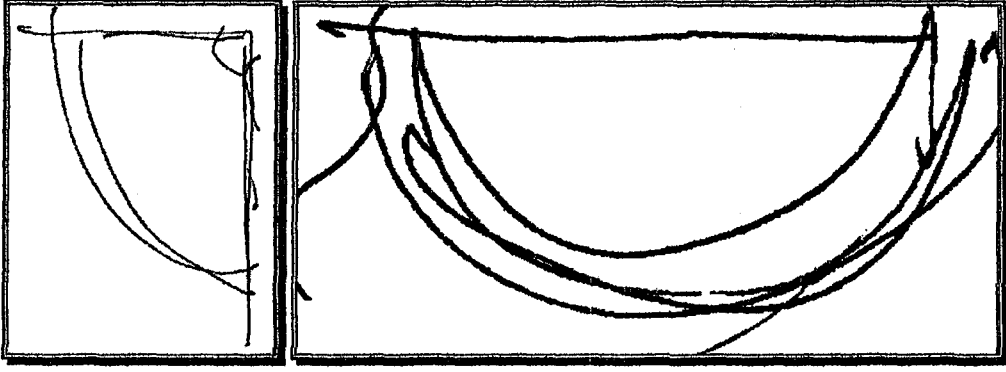
UM - Şimdi bütün bu klasik formlardan kurtularak, başka araştırmalara gelindiği takdirde, mutfakın kapalı olması isteniyor eğer yarı açık olabilseydi.

UM- O zaman tek hacim içerisinde, 45 m² yine bunları kapsar ve tavana değmeyen yüzer elemanlarla bu üç boyutu tutardı ve o zaman ışıktaki yardımcı olurdu 4. boyuttada büyüdü ve yayılırdı etrafa. O zaman bir çerçeve kısmen transparan, kısmen dış iç entegrasyonuna imkan veren. Yine giriş belki köşesel bir giriş her şey olabilir. Tek hacim içerisinde ve mutfak servis imkanı bankoda veya tekrar masada, yatak icabında açılabilir, kaldırılabilir bir yatak madem ki iki kişiler çocuksuz, bunların hepsi mümkün. Mümkün merteye yaşama sahasını günlük ihtiyaçları rahatlatma yönüne gidilebilirdi.



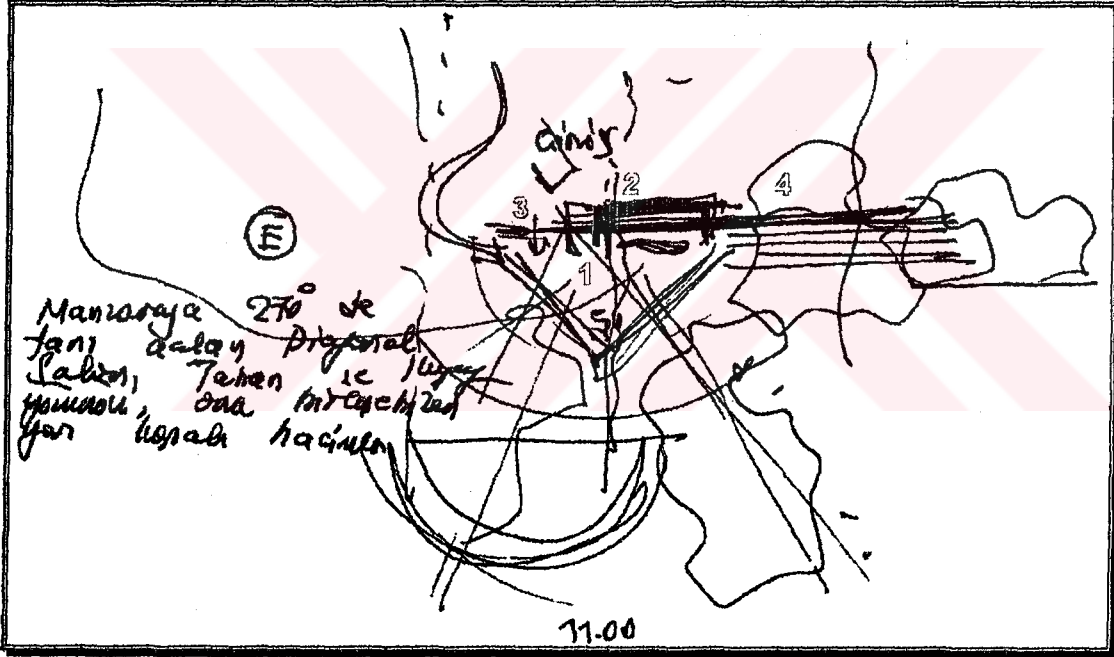
Şekil E.5. Eskiz 5 _ 1 evin dış sınırları, 2 giriş holü, 3 banyo ve mutfak ve 4 yatak odası.

Formu hep böyle çiziyorum ama böyle olması şart değil. Yine ıslak hacimler ve mutfak burada. Belki ona yakın bir yerde yatak hacmi, mesela şöyle bir şeyde olabilir yatak kadar yatak hacmi olabilir. Belki yine köşesel bir giriş, ana hacmi ısırarak sert bir eleman, sonra elegans bir eleman, Bu form yine dikdörtgen çünkü bu çok küçük bir bina olduğu için ama yine başka formlarda olabilir.



Şekil E.6. ve 7. Eskiz 6 ve 7_ Eskiz 6'da ilk manzara yönü, eskiz 7 tam manzara açılımı.

UM - Tam manzaralı fakat yerinde görmekle mümkün olabilecek şeylerde var. Burada manzara simetrik gibi bir yarım daire çiziyor, ama bu bir üçgenede dönüşebilir.



Şekil E.8. Eskiz 8_ 1. olarak salonun üçgeni yerleştirildi. 2. olarak banyo, mutfak ve yatak odası nişleri yerleştirildi. 3. olarak giriş yerleştirildi. 4. olarak ev çevresi düzenlendi.

UM - Salon mutfak veya yatak yine aynı mekan içerisinde giriş ve verdiğiniz 180 derecelik açılıma kuçak açan bir salon, tefrişi kolay yine dik açılı ve bahçeyide geyet iyi bir şekilde düzenlemek ve bezemek imkanı var. Böyle bir formda olabilirdi.

TD - Onuda düşünebiliriz.

UM - Anladım efendim.

UM - O zaman A, B olarak sınıflandıralım. Bunu A burada B zemin artı asma olsaydı. Bundan sonra tek geniş hacimde çözüm imkanı buna C diyorum sonra burada D dediğimiz alternatifde tek hacim fakat yatak bölümü minimum ebat ve max. entelitte ile çözüm burda E alternatifinde manzaraya tam hakim olan bir salon taban ve kuzey yönünde birleşe bilen.

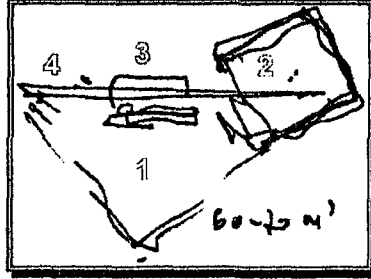
TD - Orada arsa nerede ? Arsanın sınırları...

UM - Şöyle efendim. Yine üçer metre minimum fakat 3 metre çekince 11 metre kalıyor o fazlasıyla yeterli bu tür bir yerleşime, taban çarpı yüksekliğin yarısı 9 metre alsak yine 11 metre içinde kalıyor. Hatta bir tarafa çekip bir tarafı bol bırakmak imkanı var eğer komşular onu gerektiriyorsa. Birde mümkün merteye geriye aldığımız için önde çok güzel bir açılım kalıyor. Bahçeleri formal yarı formal düzenlemek imkanı veriyor bu. Proporsiyon olarak 5'te 2 ,5'te 3 gibi kalıyor bir yana çekersek, hatta bir güzel imkanda bu servisin ve yatak odasının ayrı bahçeleri olabiliyor ki; bunlar farklı yöne bakarak nisbeten entim kalıyor. Duvar vesaireyi uzatarak bölmek imkanı doğuyor. Burda da yine girişi köşeden arıyoruz. Girdiğiniz zaman zengin bir perspektifle karşılaşıyorsunuz. Tabii yerindeki şartlarla bu diğer köşeye kayabilir. Ama böyle bir planda herhalde ortada girmemeli. Bu uzaya iyi açılan ve karışan perspektifler veriyor. Ötekilerde camda olsa bir rigite var. Burada boyutlar çok daha zengin kalacağı için, tabanlar yüksek olduğu için cam satıhıda şey ettiği için buna özellikle mevsimine göre bütün bahçeyi kavrar hale geliyorsunuz bunda.

Tabii başka formlarda olabilir. yalnız çok küçük olduğu için zorlamalar, konseptteki zaafı belli edecek nisbet bozuklukları çıkabilir tahmin ediyorum. Belki 90 ,100 metrekare olsaydı, başka şeylerde araştırılabilirdi. Bu nede olsa çok kompakt bir ev tek hacimle çok iyi çözülmesi gerekiyor. Asma kat olsa çok başka şeyler çıkabilir. O olmadığına göre bu tip formları önerebiliyorum.

TH - Aslında verilen föylerde 45 m2. sınırı yok. Yani Evin alanı daha büyütülebilir. Yalnızca mekan sayısında bir sınırlama var.

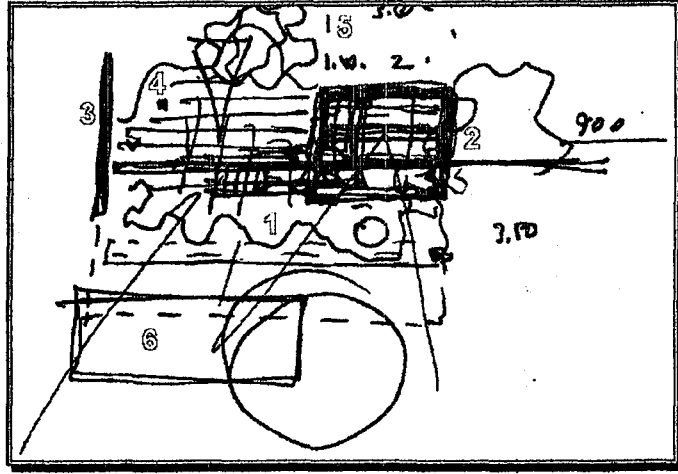
UM - Yani şunu söylemek istiyorum Gün; tek katlı olduğunda bu ebatlar 15 m2. büyük olsa pek fazla bir şey getirmez sanıyorum.



Şekil E.9. Eskiz 9 _ 1. olarak salon yerleştirildi. 2. olarak ayrı yatak hacmi, 3. olarak banyo ve mutfak nişleri ve 4. olarak giriş çizildi.

UM - Olsa olsa yavaş yavaş yatak odasını koparmaya gidebilirsin. Kendi içerisinde ayrı bir bölüm haline getirmeye başlayabilirsin. Ama her halükarda her eleman çok büyük bir eleman olamayacağı için bunu yatak odası yapıp, burada mutfağı yapıp yine girişi buradan yapınca yine 60 70 m2.de çok büyük bir faydası olmayacağı kanaatindeyim. Onun için ne kadar pür olursa o kadar iyi. Geometrik form içerisinde halledebilirsek. Ne kadar temiz olursa o kadar üçgen, dikdörtgen vs. yani hatta yamuk bile olmamalı, çünkü metrekare enteresan, kritik. Ama bu asma katlı olursa çok başka şeyler olabilir.

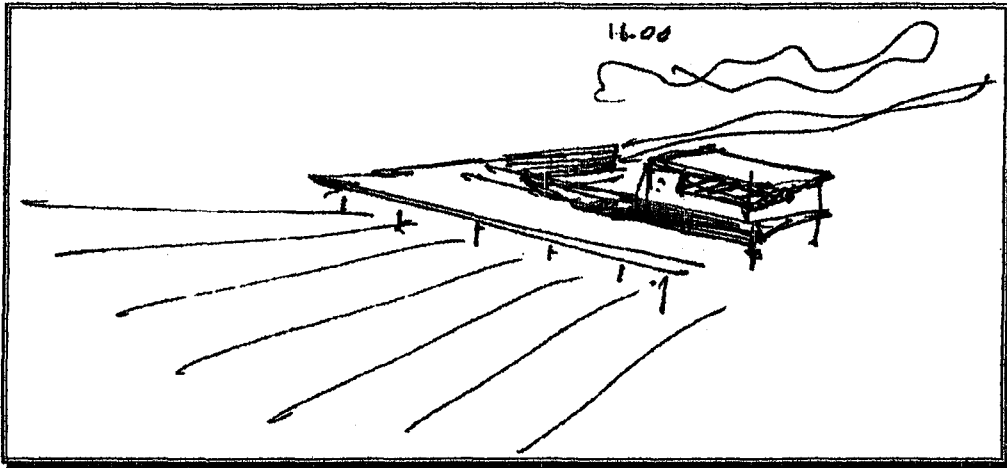
TD - Programlaması zor olduğu için. Buna girmedik. Aslında ne olabilir. Siz önerin, biz onları da sonra inceleriz.



Şekil E.10. Eskiz 10_1. olarak salon çizildi. 2. olarak ikinci kattaki yatak odası ve altında yer alacak banyo ve mutfak işlendi. 3. olarak dikey duvar. 4. olarak giriş işlendi. 5. olarak arka bahçe düzenlemesi yapıldı. 6. olarak mekan boyutları ve önde olası havuz işlendi.

UM - Diyelim ki max. lineer olarak kullanacağız. Farz edelim. Bahçenin güneydoğu güneybatı doğrultusunda çok güzel bir görünüm var. Max. geri çektik o zaman bu planın 11 metreye kadar gelmesinde fayda var, arkayı böldükten sonra. O zaman ne oluyor salon ve mutfak aşağıda ondan sonra buradan yukarıya çıkıyoruz burada da şöyle yatak olsa.

TD - Bu kesit mi? plan mı?



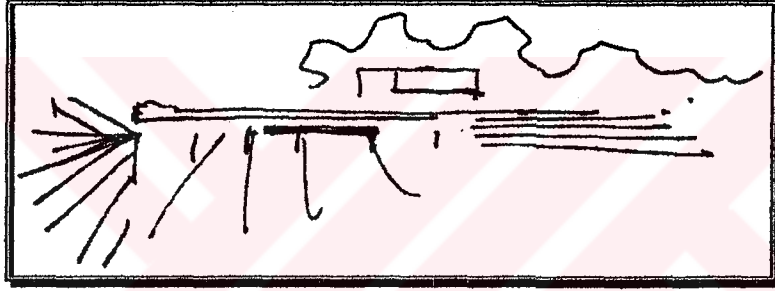
Şekil E.11. Eskiz 11_ Eskiz 10'daki evin perspektifi.

UM - Plan efendim. Buradan girdik üstü kapalı ... bunu buna bindirdik. ... bu living ... buradan da girişi, burada bütün bahçeyi

içeriye soksak tümüyle salon yukarı çıkan merdiven bunu içinde merdiven salonu ısıyor, salon büyük bir açıyla açılıyor, yukarıda yatak odası.

TD - Salonun sınırları nerede ?

UM - Salonun sınırları şöyle. Üstü yatak bunun, altı mutfak. Dolayısıyla burada ilginç bir pasyo çıkıyor. Giriş şurada, Arka bahçe gerçi şimale açık ama servis avlusu olarak çok iyi. Yani arabasını park eder. Yukardaki yatak odası bütün ayak şeylerinden kurtuluyor. Yine çok basit ama böyle bir manzaraya karşı uyanmak çok güzel tabii. Salonun tabii örtü kaplanmasını düşünürsek son derece güzel azizim.

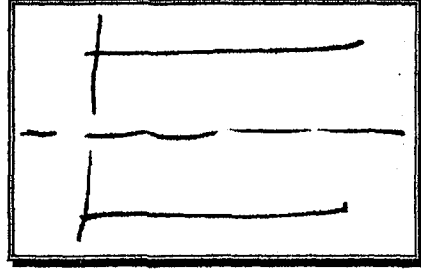


Şekil E.12. Eskiz 12_ Eskiz 10'daki evin ön görünüşü

UM - Buradan çekilen bir hat bu böyle çıkıyor. Baktığımız zaman son derece basit. Bu duvar olarak uzuyor bu bize geliyor. Yukarıda bu tarafa yine duvar gidiyor. Tabii içini göre biliyorsunuz şeffaf. Son derece eleTHn hem de yapımı basit malzeme itibari ile ister beton olsun ister ahşap olsun son derece açık. Bu sempati ile giriyor bu içeri kadar nakil oluyor. Merdiven zenginlik veriyor baş kurtardıktan sonra. Yukarıda bir duşu belki aşağıda bir tuvaleti ve mutfağı. Mutfağı 1.10-1.20m, mutfağı 2m. yukarıda 3-3.10m 9 m2. yatak odası. Aşağıda da 3.25-3.50m min. tutmalı ama 10m'ye kadar giden bir salon ama salonun darlığı hiç hissedilmeyecek tümüyle açılıyor. Tabii bütün saçağı portaforda çıkarmak mümkün. Tabii bu bahçe çok güzel olur havuzuyla daire veya dikdörtgen şekliyle. Böylece yaz mevsiminde tümüyle kullanılır hale geliyor. Salon olarakta bölmek yani burda yemek imkanı belki içli dışlı şöminesi, burada giriş hemen girişin yanında yukarı çıkma imkanı olabilir. Buna da F diyoruz. Asma katlı olması hali.

Başka neler yapalım. Alternatifler çok, ama sonu yok.

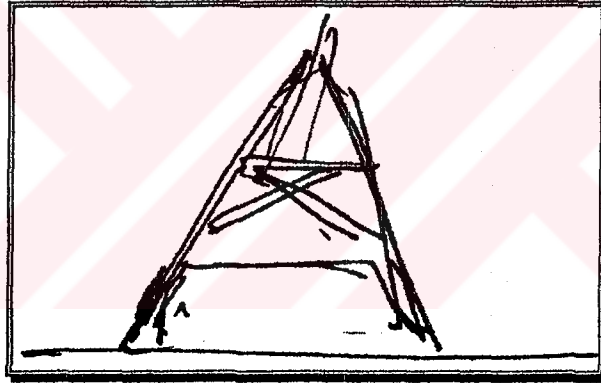
TD - Çatısını ne yapalım.



Şekil E.13. Eskiz 13_ Çatı planı.

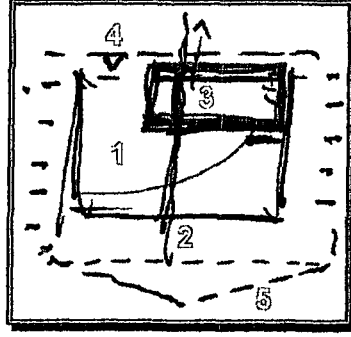
UM - Aynı şeyi söyleyecektim. Mesela planı dikdörtgen yapsak.

TD - Yine tek katlı alternatif üzerinden gidelim.



Şekil E.14. Eskiz 14_ Çatı Görünüşü

UM - Tabii. Tek katlı yaptığımız takdirde hacimleri yüzer yaptığımızda, Çatı taşıyıcılarıyla birlikte yükselse, uzayı kapatsa çok iyi tabii. Buradan tepe ışığı falan alsa, küçük hacimde yüksek olması çok iyi olabilir yerine göre. Taşıyıcılarıyla yere kadar inen bir çatı. Buna mükabil önü açılıyor, cam. Yine bir dikdörtgen mahya böyle gidiyor.



Şekil E.15. Eskiz 15_ 1. olarak evin sınırları. 2. olarak çatı mahyası. 3. olarak yatak odası ve belki ikinci kattaki ütü veya çamaşır odası. 4. olarak giriş çizildi. 5. olarak çatının hizası ve çatı konstrüksiyonun zeminde kesite giren taşıyıcıları.

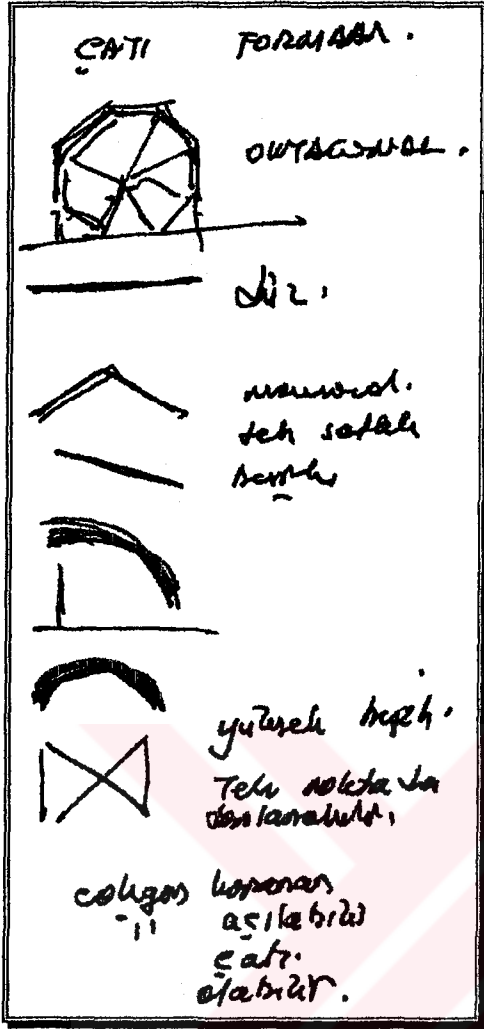
UM - Demin çizdiğimiz gibi yatak ve ıslak hacim burada ayrılıyor. Bunu üstü ütü, çamaşır odası gibi yaşanmayan bir mekan olabilir. Ona belki çekme bir merdivenle çıkabilirsin, fazla yer kaybı olmasın. Yanlız öne açık konstrüksiyon meğilden dolayı dışarı geliyor. Çatı söyle nokta nokta öndeki saçak daha çok çıkıyor çünkü daha fazla güneş alan bir yer. Belki meğilli.



Şekil E.16. Eskiz 16_ Eskiz 15 ve 14'deki evin görünüşü.

UM - Cepheye baktığımızda şurasıyla tümüyle açık yukarısı belki açık belki kapalı. Sempatik bir malzeme ağırlıklı cam böyle durmasın diye. Ve devamında böyle. İstenirse yan ışıklarla çatıda oynayabilirsin. Bu üstte çamaşır veya ütü odası yapabilirsin. Bunun bir iyi yönünde kışın ısıtma olmadan bile kurander olmasada yani irregüler bir ısıtma olsa bile yatak odasının sıcak kalması ve mutfakın koku yapması halinde kendi içinde halletmesi ve dışarı atması. O faydası var.

Aslında prefabrike olabilir. Ordan doğma planlar gelebilir. Üretici küçük elemanlar olabilir. Bir zamanlar İsrail'de yapmışlardı. Bunların hafif metal olması gibi sıcak memleket olduğu taktirde ...



Şekil E.17. Eskiz 17_ Olası çatı tipleri.

TD - O nedir ? plan mı ? kesit mi ?

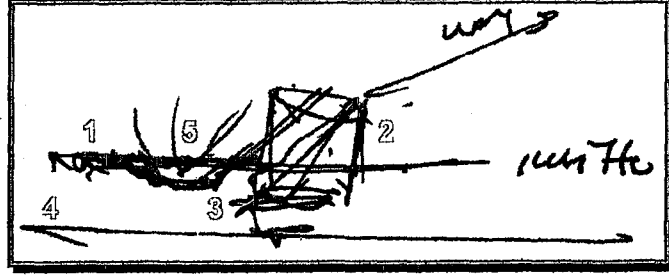
UM - Bunlar kesit ama form olarak planda da kullanabiliyoruz görünüşte de strüktürel bağlantıları görünüyor.

TD - Bu jeodezik kübbe gibi bir şey mi oluyor.

MB - Sıcak memlekette olduğu taktirde üzerinde hasır falan kullanarak. Biraz irregüler olsada kendi formu içinde çözümler olabilir.

TD - Peki bu dikdörtgene başka nasıl bir çatı olabilir.

UM - Çatı formları. Şimdi en basitinden gidelim efendim; küçük mafsallı, bir tane tek sıklı, bir tane beşik, bir tane dörtte bir silindir, bir tane yarım silindir. Beton ve izolasyon, yine çok yüksek beşik, tek noktada toplanabilir kare veya dikdörtgende olsa olabilir, düz. Tabii koniksel şeylerde olabilir ama 45 m². için çok fazla olur. Ama iklim onu icap ettirir, ters şemsiyede olabilir. Veya hut çok geniş şemsiye olur içine kapanırsın. Çokgen kapanan veya çokgen açılabilen çatı olabilir.

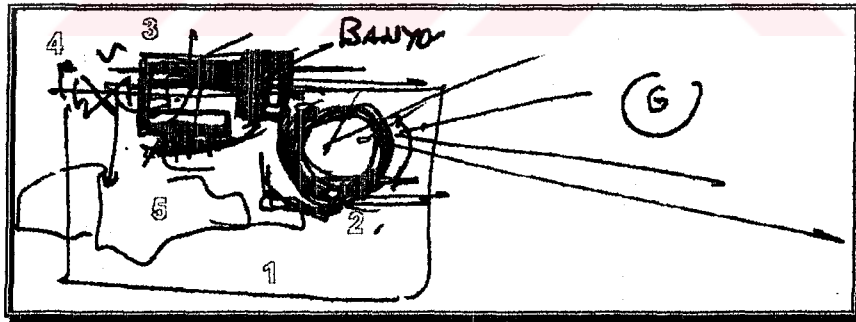


Şekil E.18. Eskiz 18_1. olarak birinci katın çatı hizası. 2. olarak yatak odasının bulunacağı silindirdir. 3 olarak yatak odasının zemin katla bağlantısı. 4. olarak zemin katın döşeme hizası. 5. olarak mutfak çatı penceresi.

UM - Tabii konuştuğça bir çok alternatif geliyor. Düz olurda, o kadar güzel bir mehtapta adam ayda bir iki gece görmek ister. Yatak veya mutfak orayı yükseltmen gerekebilir. Tabii bunları göz önünde tutabilirsin. O eğimlerle yatak odasını veya yatak nişini bir şekilde halletmek. Sonra burada bir bağlantıyla mutfak açmak çökertip yani dışı sadece sağır fakat çatıdan hava alan bir mutfak tertip ki plastik elemanları ...

TD - O nedir ? plan mı ? kesit mi ?

UM - Kesitte efendim.



Şekil E.19. Eskiz 19_1. olarak evin sınırları. 2. olarak yatak odası nişi. 3 mutfak odası veya açık mutfak ve yanında banyo belki iki girişli. 4 olarak rüzgarlıklıklı giriş. 5. olarak salon tefrişi.

UM - Planda burdan alsak yatağı bu şekilde böyle halletsek, bunun üstüne gelecek bir eleman bununla beraber yuvarlak bir yatakda olabilir. Baş bu tarafa kapalı. yandan girilen odadan bakıldığında gözükmeyen fakat ters istikamete açık. Belki bir basamakta yüksek, malzemesi farklı, salona nazaran da olabilir. Belki

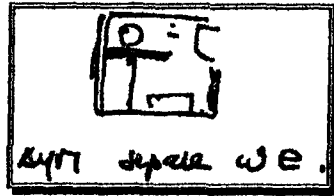
bu tarafıda kapalı koltuklarda dönebilirsin. O zaman çok az yer kaplayacak 2.80 - 3.00 m çaplı bir yer fazlasıyla kafi bütün mekan oturma hacmine kalacak. Burayada yine kapalı veya açık mutfak gelecek, kapalı olur çünkü öyle isteniyor. RürTHrlikle da bir giriş gelecek. Buradan servisi, buradan girişi. Belki yatak nişinin salon kısmı daha yerli oturma imkanı taşıyabilir. Sonra mobilyalar gelebilir.

TD - Aslında açık mutfak düşünebiliriz.

TH - Evet. Artık açık mutfak düşünmemiz gerekiyor sanırım.

UM - Evet 45 m2. de açık olsa mutfak, açık ama koku yapmayacak bir şekilde yani alaturka yemek pişirme imkanı olan ama kokuyu çabuk dışarı atabilecek imkanı olan. Mesela şurda açık fakat burada fanlarla falan kokuyu hemen atabilecek. İyi bir tesisatı olamasa bile korenle atabilecek. Yani davlumbazı cam veya burda çift taraflı açabilecek veya pişirmenin önü sürgü, o zaman pişen yemek burada servis olabilecek o zaman masayada gerek kalmıyor. Çok rahatlar kahvaltı için geçersin yemek için geçersin. Burada tertemiz bir hacim gelir.

TD - Banyo ?



Şekil E.20. Eskiz 20_ Banyo örneği

UM - Buraya koyuyorum efendim. Belki iki kapısıda olabilir. İki kişi ama farz etki misafir var. Üç dört kişi oldular. Genellikle banyo içinde tuvaleti ayrı yaparsan çok iyi olur. Kesinlikle iç içe olmasın. Tek hacim olsun tamam ama böyle giriliyorsa burda tuvalet böyle olsun. O zaman tekde olsa iki kişi kullanma imkanı doğuyor. Tesisat açısından da ekstra bir külfet getirmiyor.

TD - Çok teŖekkür ederiz. Eskizleri de alacađız. Bunları tezdede kullanıcađız. Bir mahsuru yoktur inŖallah, isminizle tabii.

TH - TeŖekkür ederim.

Yarım saat süren görüşmenin sonu.



Ek F. Kurallar Tablolarında Kullanılan Kısaltmalar.

Tablo F.1. Kurallarda Kullanılan Kısaltmalar.

Bilgi Türü	Kısalt	Bilgi Türü	Kısalt	Bilgi Türü	Kısalt
Uzman Bilgisi	UB	Uzmandan İstenen	UI	Uzman Tercih	UT
Kuzey Açısı	KA	Hakim Rüzgar Açısı	HA	Manzara Açısı	MA
Dikdörtgen	DD	Üçgen	ÜG	Salon	SAL
Yatak Odası	YOD	Mutfak	MUT	Banyo	BAN
Giriş	GİR	Alan Hesabı	AH		

Ek H. Bilgi Giriş için Program Denemesi

```
; 5:40 18/12/92 tmask1.lsp
;2345678901234567890123456789012345678901234567890123456789
01234567890
(princ "\n          1992 - 1993 Ogretim Yili")
(princ "\n          Yazlik Ev icin Uzman Sistem v 1.0")
(princ "\n      Tez Yurutucusu: Prof. Dr. Nigan Bayazit * Tez Sahibi: Mim. Gun Alev")
(princ "\n")
(princ "\n")
(princ (strcat "\nMin. Arazi alani          : " (rtos aralmin 2 2) " m2"))
(princ (strcat "\nMin. Arazinin yatay uzunlugu : " (rtos arxmin 2 2) " m"))
(princ (strcat "\nMin. Arazinin dusey uzunlugu : " (rtos arymmin 2 2) " m"))
(princ (strcat "\nMin. Evin oturma alani      : " (rtos otalmin 2 2) " m2"))
(princ "\n")
(princ (strcat "\nArazi alani          : " (rtos aral 2 2) " m2"))
(princ (strcat "\nArazinin yatay uzunlugu      : " (rtos arx 2 2) " m"))
(princ (strcat "\nArazinin dusey uzunlugu      : " (rtos ary 2 2) " m"))
(princ (strcat "\nArazinin KAKS'i          : " (rtos kaks 2 2) ))
(princ (strcat "\nArazinin TAKS'i          : " (rtos taks 2 2) ))
(princ (strcat "\nYazlik Evin oturma alani      : " (rtos otal 2 2) " m2"))
(princ "\n")
(princ "\n")
(princ (strcat "\nSalonun Net alani          : " (rtos sal 2 2) " m2"))
(princ (strcat "\nSal. Yatay uzun. : " (rtos salx 2 2) " m * Sal. Dusey uzun.: " (rtos saly 2 2) "
m"))
(princ (strcat "\nMutfagin Net alani          : " (rtos mut 2 2) " m2"))
(princ (strcat "\nMut. Yatay uzun. : " (rtos mutx 2 2) " m * Mut. Dusey uzun.: " (rtos muty
2 2) " m"))
(princ (strcat "\nBanyonun Net alani          : " (rtos ban 2 2) " m2"))
(princ (strcat "\nBan. Yatay uzun. : " (rtos banx 2 2) " m * Ban. Dusey uzun.: " (rtos bany
2 2) " m"))
(princ (strcat "\nYatak Odasinin Net alani      : " (rtos yat 2 2) " m2"))
(princ (strcat "\nYat. Yatay uzun. : " (rtos yatx 2 2) " m * Yat. Dusey uzun.: " (rtos yaty 2 2)
" m"))
(princ "\n")
(princ "\n")
```

Özgeçmiş

Halen İ.T.Ü. Fen Bilimleri Enstitüsü, Mimarlık Ana Bilim dalı Bina Bilgisi programında Yüksek Lisans öğrencisi olan Mimar Gün Alev, Ocak 24, 1970'de Ankara'da doğmuştur. İlkokulu, Ortaokulu Şişli Terakki Vakfı Özel Şişli Terakki Lise'side bitirmiştir. Lise öğrenimini de aynı okulda iyi derece ile 12. olarak tamamlamıştır. Lise eğitimi sırasında TÜBİTAK Liseler arası Fizik Yarışmasında mansiyon almıştır.

1987 yılında İ.T.Ü. Mimarlık Fakültesi, Mimarlık Bölümünü birincilikle kazanmış ve 1991'de bu eğitimini altıncı olarak bitirmiştir. Aynı yıl Yüksek Lisans programına sınavsız girmiştir. Yabancı dillerden İngilizce'yi bilmektedir. Bilgisayar kullanmaya onbir yaşında başlamıştır. Yüksek düzeyli BASIC, AutoLISP ve DCL programlama dillerini ve orta düzeyde FORTRAN ve DBASE programlama dillerini kullanmaktadır. Ayrıca IBM RISC 6000 serisi WorkStation'lar üzerinde çalışan CATIA CAD/CAM paket programını kullanabilmektedir. Yüksek Lisans programının bitirme ödevini AutoLISP dilinde AutoCAD programı üzerinden bir Uzman Sistem yapmıştır.

Stajlarının birincisini Mimar Cengiz Bektaş'ın yanında 'Mersin Gökdeleni'nde, ikincisini 'Enka-Libya Brega New Town Şantiyesi İnce İşler Bölümü'nde ve üçüncü stajını Finlandiya'nın önde gelen 'CJN Mimarlık Bürosu'nda bilgisayar destekli tasarım bölümünde Helsinki yenileme projesi ve T Tower projeleri üzerinde çalışarak yapmıştır.

Lisans eğitiminden sonra serbest mimar olarak çalışmaya başlamıştır. Halen mimari planlama, projelendirme ve mimarlara çizim ve geliştirilmiş mimari sunum konusunda destek veren bir mimari büronun ortağı olarak çalışmalarına devam etmektedir.