

eMON: Gerçek Zamanlı Gömülü Sistemlerin Çalışma Zamanı Görselleştirilmesi İçin Monitör Yazılımı

1

Berkant AKIN
Mehmet GÖKÇAY, Kaan DOĞAN
TUBİTAK-SAGE

Ulusal Yazılım Mimarisi Konferansı
Ankara, 2010

İçerik

2

- Neden eMON Yazılımı?
- eMON Yazılım Mimarisi
 - Sembol Yöneticisi
 - Hata Ayıklama Bilgi Biçimleri
 - ✦ ELF, DWARF, COFF,
 - Parametre Yöneticisi
 - Betik(script) Yöneticisi
 - Görsel Bileşen Mimarisi
 - Gömülü İşler, eMON-Ghost
- Sonuç

Neden eMON Yazılımı?

3

- Deneycilik

- Var olmak $\leftarrow \Sigma$ Gözlemek



Neden eMON Yazılımı?

4

- Gerçek zamanda hedef bilgisayarı durdurmadan gözlemleyebilmek
 - Dinamik Sembol Yönetimi
- Gömülü ortamdan dışarıya atılan verinin tanımlanması ve görselleştirilmesi
 - Statik Sembol Yönetimi
- Etkin gözlemlene ve görselleştirme sayesinde yazılım geliştirme ve test süresini kısaltmak

eMON Yazılım Mimarisi

5

- **eMON-PC**
 - Sembol Yönetimi
 - Görselleştirme
 - Parametre Yönetimi
 - Betik Yönetimi
- **eMON-Ghost**
 - Gömülü ortamda sembollerin örnekleme
 - Veri iletişimi
 - İşletim Sistemi bağımsız

eMON-PC GUI

embeddedSystemMonitor

File View Data Connection Tools Window Help

☒ SaveScript

Default/display2/display3/

Project Tree

- Projects
 - H:\mon_paper\fab_mon_desc.
 - Files
 - ornek.xml
 - Functions
 - Symbols
 - void ornek
 - int:1 a
 - int:3 b
 - float arr[4]
 - void veri

ornek.arr

Harita

GorevSeciniz Plot Mehod: Bitmap DrawPath

D3D

fps 0.097 Position <0.58469> <0.68417> <20.00000>

Object Editor

- ornek
 - int:1 a
 - int:3 b
 - float arr
 - void veri

saver

File Saver

- a
- b
- arr[0]
- arr[1]
- arr[2]
- arr[3]
- flag
- a
- b
- mem[0]
- mem[1]
- mem[2]
- mem[3]

char flag

flag	
arr[0]	
arr[1]	
arr[2]	
arr[3]	
a	
b	

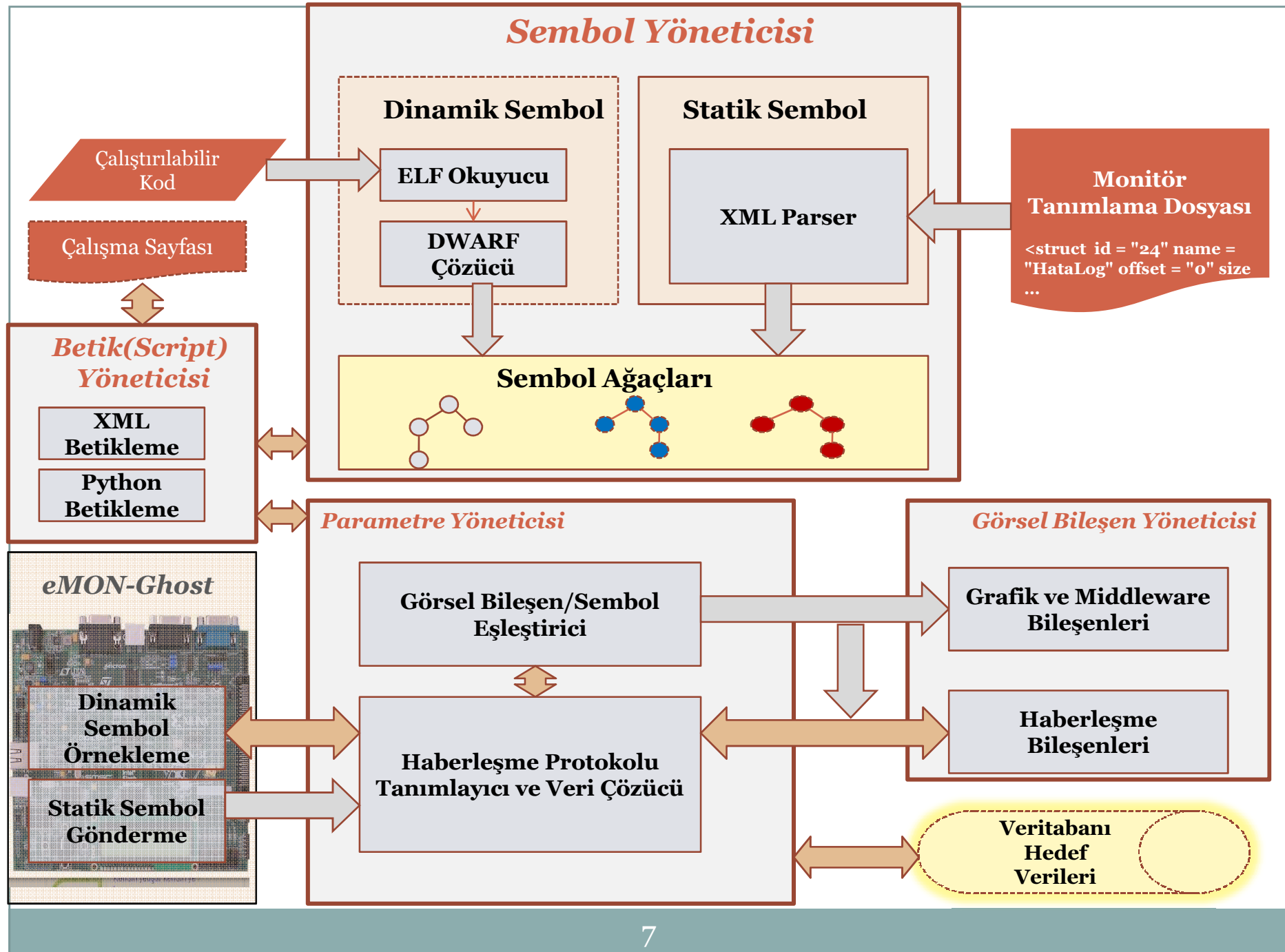
lla: 36.01156 39.39336 -1000.000000

lla: 36.01156 39.39336 -1000.000000

fps 0.097 Position <lat: 0.000> <lon: 180.00>

```
width="260" height="176" window_state="0"/><SCREEN_SHOOT_TAB5_DATA tab_name="Default" tab_index="0" title="D3D" pos_x="579" pos_y="174" width="276" height="228" window_state="0"/><SCREEN_SHOOT_TAB5_DATA tab_name="Default" tab_index="0" title="saver" pos_x="226" pos_y="302" width="200" height="200" window_state="0"/><SCREEN_SHOOT_TAB5_DATA tab_name="Default" tab_index="0" title="Harita" pos_x="892" pos_y="100" width="719" height="596" window_state="0"/></SCREEN_SHOOT_TAB5>
Mouse Down
<_HUD><TITLE title="HUD" /></_HUD>
```

Probe Count 0 Monitor Count 0 Packet Waiting 0 TermPort Time



Dinamik Sembol Yöneticisi

Hata Ayıklama Bilgi Biçimleri

8

- IDE'ler içerisinde yer alan “Hata Ayıklayıcılar(debugger)” hata ayıklama bilgilerini kullanarak işlevlerini yerine getirirler
- Populer Hata Ayıklama Biçimleri
 - ELF /DWARF
 - COFF/STABS
 - .PDB

eMON Yazılımı hata ayıklama bilgisi içerisinde yer alan verileri kullanarak sembol bilgilerine erişir.

Dinamik Sembol Yöneticisi

Hata Ayıklama, DWARF

9

- **DWARF: Debug With Arbitrary Record Format**
 - Popüler hata ayıklama bilgi formatıdır.
 - Bilgi tanımları(değişken, tip, fonksiyon) DIE(debug information entry) diye adlandırılan veri yapıları ile tanımlanır.
 - ✦ DIE'ler ağaç yapısındadır
 - ✦ DIE'ler Derleme birimleri (*.c/*.cpp, ...) içerisinde yer alırlar
- **DWARF bilgisi ELF(executable & linkable file) dosyası içerisindeki aşağıdaki bölümler içine yerleştirilir**
 - .debug_info, .debug_abbrev, .strtab, .debug_str

Hata Ayıklama, DWARF

10

```
hello.c:
1: int main()
2: {
3:   printf("Hello World!\n");
4:   return 0;
5: }
```

DIE - Compilation Unit

Dir = /home/dwarf/examples
Name = hello.c
LowPC = 0x0
HighPC = 0x2b
Producer = GCC



DIE - Subprogram

Name = main
File = hello.c
Line = 2
Type = int
LowPC = 0x0
HighPC = 0x2b
External = yes

DIE - Base Type

Name = int
ByteSize = 4
Encoding = signed
integer

Graphical representation of DWARF data

```
DW_TAG_base_type
    DW_AT_name = int
    DW_AT_byte_size = 4
    DW_AT_encoding = signed
```

int base type on 32-bit processor.

```
<1>: DW_TAG_variable
    DW_AT_name = px
    DW_AT_type = <2>
```

```
<2>: DW_TAG_pointer_type
    DW_AT_byte_size = 4
    DW_AT_type = <3>
```

```
<3>: DW_TAG_base_type
    DW_AT_name = word
    DW_AT_byte_size = 4
    DW_AT_encoding = signed
```

DWARF description of "int *px".

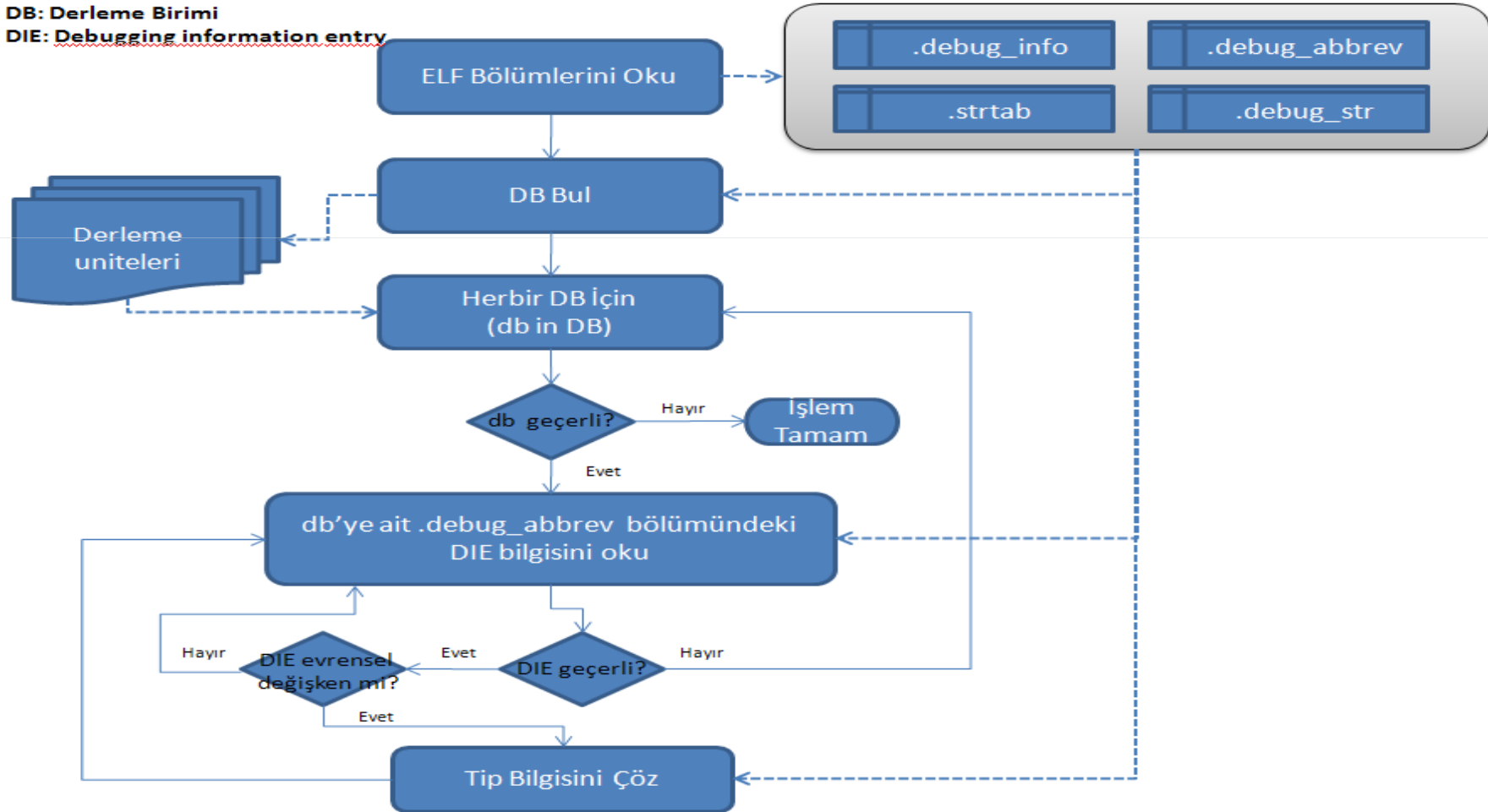
Sembol Yöneticisi, Dinamik Sembol Yönetimi

11

- Evrensel Değişkenlerin Tip/Boyut/Adres bilgileri DWARF içerisinde elde edilir

DB: Derleme Birimi

DIE: Debugging information entry



Sembol Yöneticisi, Statik Sembol Yönetimi

12

- Monitör tanımlama dosyaları ile yazılım içerisindeki semboller tanımlanır.
- iki ana bölümden oluşur
 - Mimarisel Bilgiler
 - ✦ Byte Boyutu
 - ✦ Temel Veri tipleri
 - ✦ Kodlama bilgisi(float, int, ..)
 - Sembol Bilgileri
 - ✦ DWARF uyumlu offset, boyut ve tip bilgileri
 - Offset bilgisi derleyiciden derleyiciye göre farklılık gösterebilir
 - ✦ Benzersiz ID

Statik Sembol Yönetimi, Örnek Monitör Tanımla

```
struct ornek {  
    int a:1;  
    int b:3;  
    float arr[4];  
    struct{  
        char flag;  
        double mem[16];  
    }veri;  
}
```

```
<file name = "ornek_mon_tanimlama">  
  <architecture>  
    <byte_size>8</byte_size>  
    <types>  
      <type name = "char" bytes = "1" encoding = "DW_ATE_signed_char"/>  
      <type name = "int" bytes = "4" encoding = "DW_ATE_signed"/>  
      <type name = "float" bytes = "4" encoding = "DW_ATE_float"/>  
      <type name = "double" bytes = "8" encoding = "DW_ATE_double"/>  
    </types>  
  </architecture>  
  <struct id = "0" name = "ornek " offset = "0" size = "66">  
    <field name = "a" type="int" bit_offset="0" bit_size = "1" offset = "0"/>  
    <field name = "b" type="int" bit_offset = "1" bit_size = "3" offset = "0"/>  
    <field name = "arr" type = "float" array = "[4]" offset = "1"/>  
    <struct name = "veri " offset = "17" size = "49">  
      <field name = "flag" type = "char" offset = "0"/>  
      <field name = "mem" type = "double" array = "[16]" offset = "1"/>  
    </struct>  
  </struct>  
</file>
```

Betik Yöneticisi

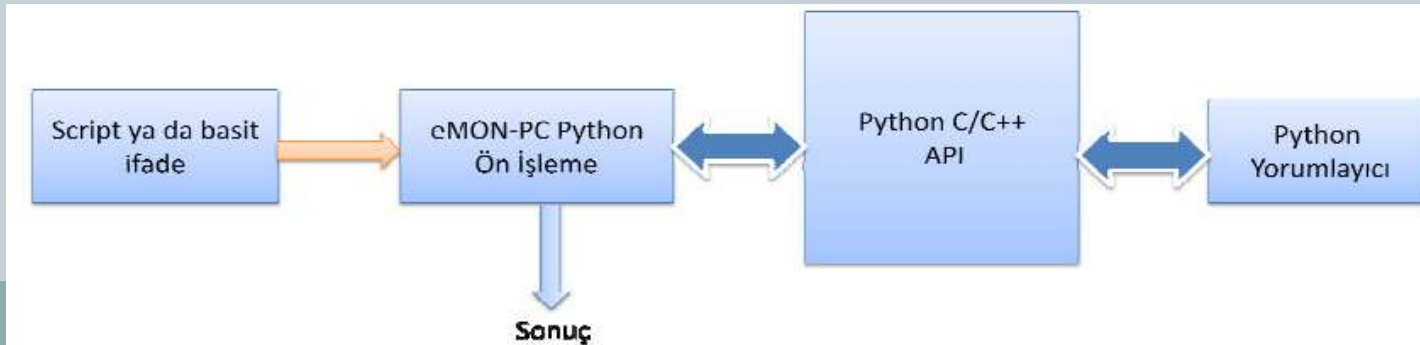
14

- Tüm kullanıcı arayüzü işlevleri betiklerle yapılır
 - XML tabanlı görsel bileşen / sembol ilişkilendirilmesi

```
1. <LOAD_MON_PROJECT project = "C:\ornek.xml" />
2. <ADD_OBJECT project = "C:\ornek.xml" file = "ornek_mon_tanimlama" path = "ornek → veri → flag"/>
3. <GRID>
4.   <TITLE title="FlagIzgara" grid_style="0"/>
5.     <GROUP_GRIDER project="C:\ornek.xml" file = "ornek_mon_tanimlama" path = "flag" grid_style= "0"/>
6. </GRID>
```

- Python Betikleme

- ✦ Görsel Bileşen tetiklemesi
- ✦ Görsel bileşen verileri üzerinde işlem yapabilme yeteneği
- ✦ Python C/C++ API



Parametre Yöneticisi

15

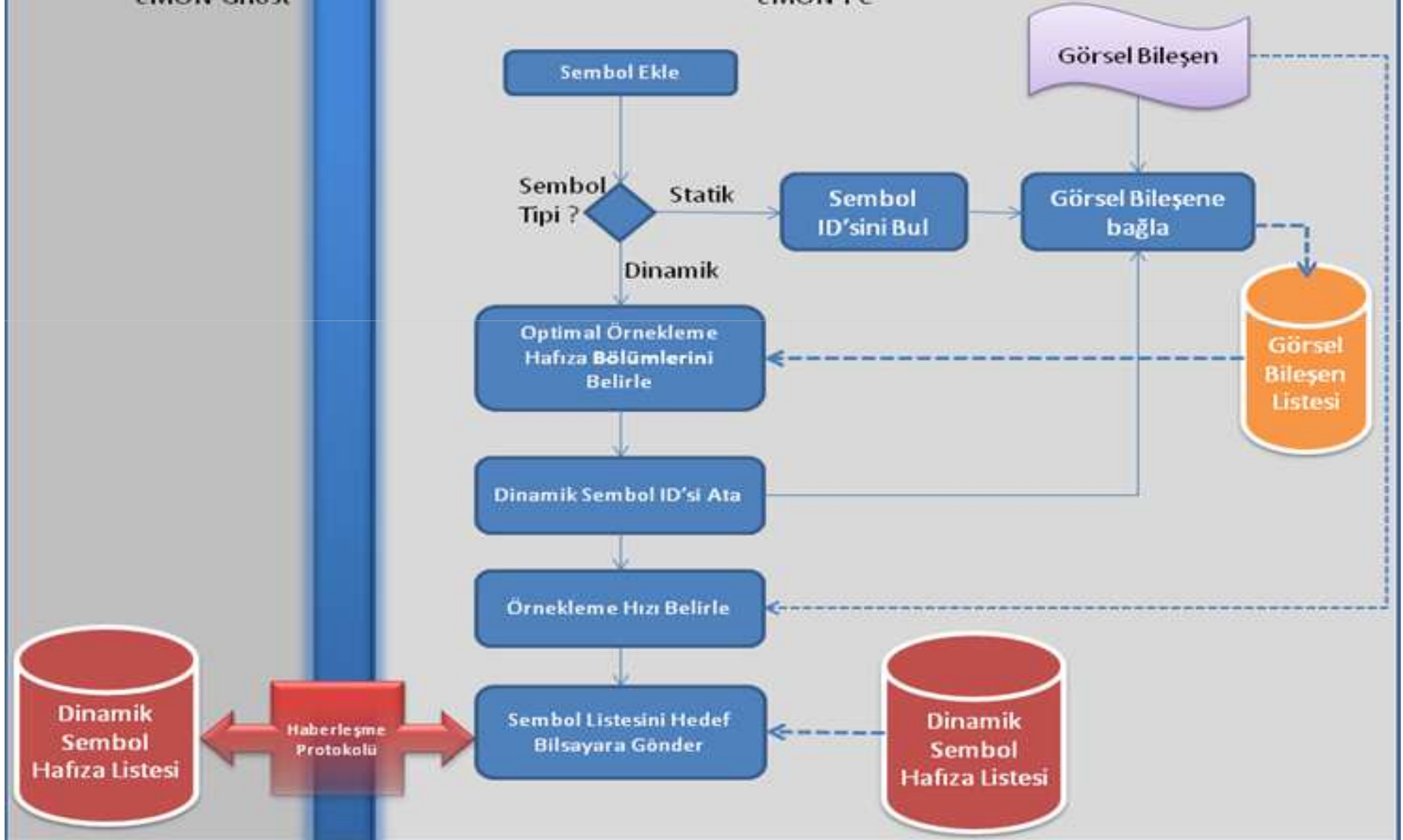
- **Sembol Ekleme**
 - Sembol-Görsel Bileşen eşleştirilmesinin yapılması
 - Dinamik sembollerin gruplanması
 - ✦ Optimal gruplama
- **Sembol Güncelleme**
 - Hedef bilgisayardan alınan verilerin DWARF kodlama bilgilerine göre çözülmesi

Parametre Yöneticisi, Sembol Ekleme

16

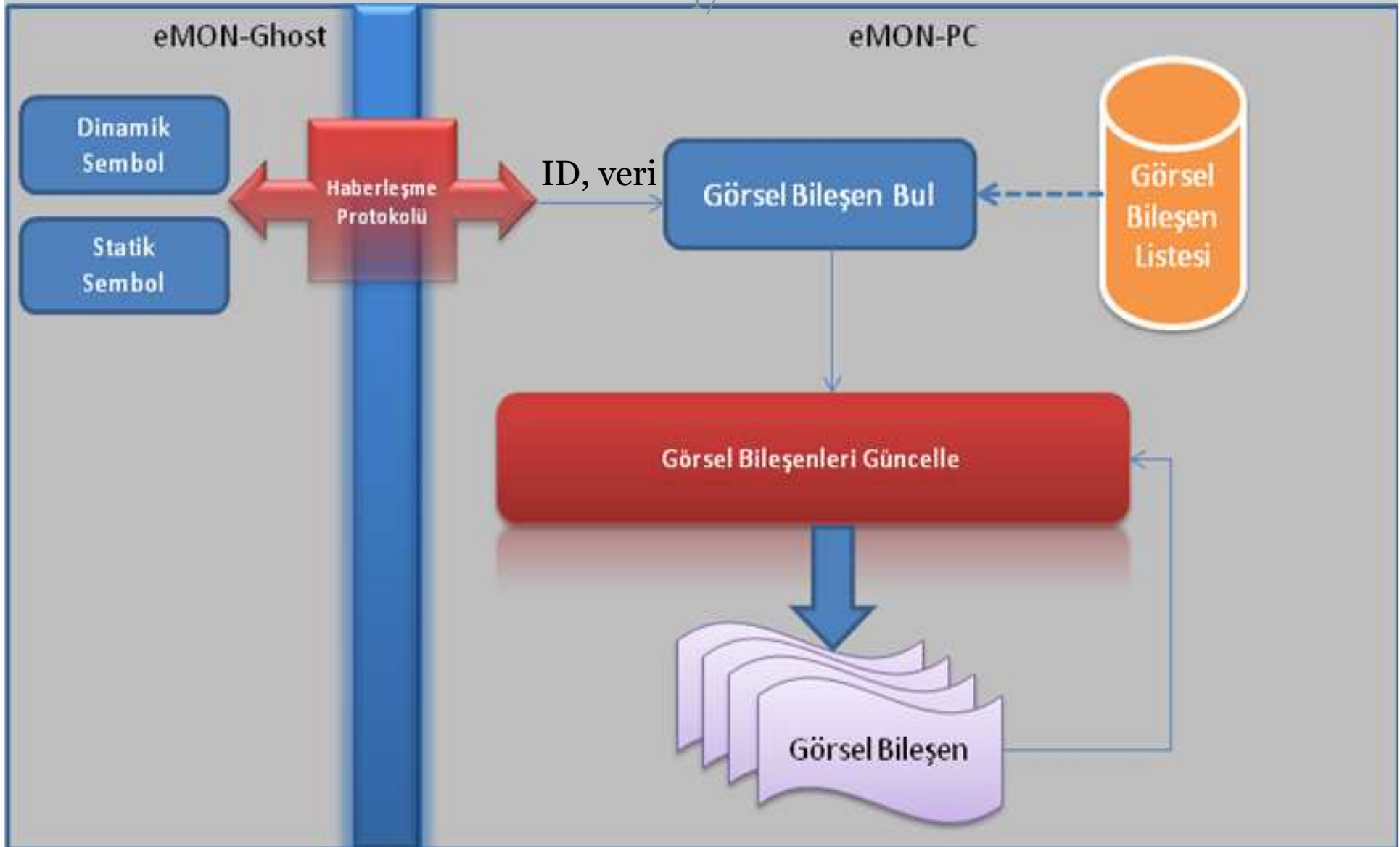
eMON-Ghost

eMON-PC



Parametre Yöneticisi, Sembol Güncelleme

17

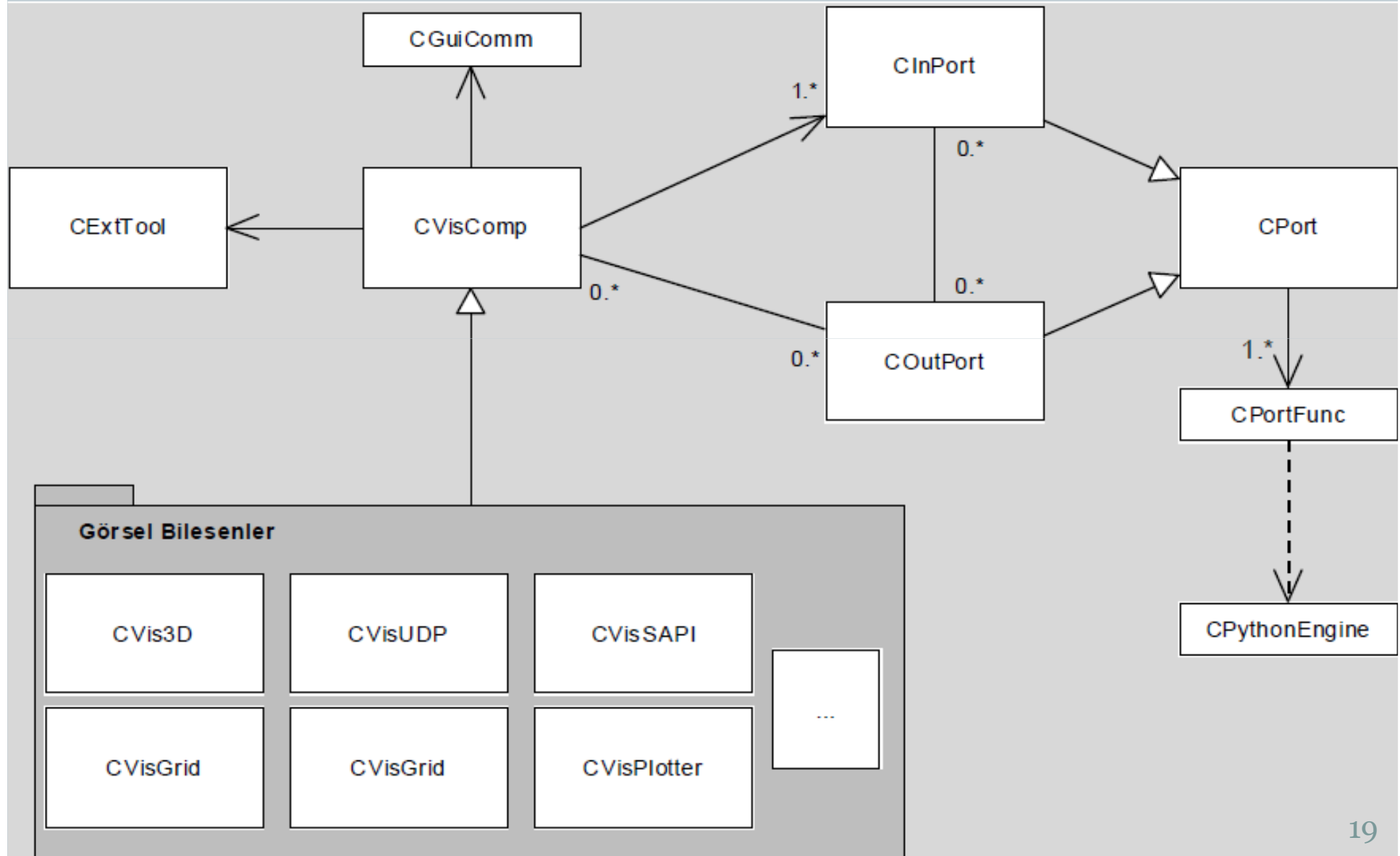


Parametre Yöneticisi, Haberleşme

18

- Geçerli haberleşme arayüzü Ethernet / UDP
 - RS232 / SpaceWire gibi iletişim arayuzleri de mevcuttur
- XML tabanlı haberleşme mesaj tanımlama arayüzü
 - Esneklik
 - Evrensellik

Görsel Bileşen Yöneticisi



eMON-Ghost

20

- Hedef bilgisayarda çalışan C/C++ temelli gömülü yazılımdır
 - Dinamik sembol listesini yönetmek
 - Dinamik sembol örneklemesini gerçekleştirmek
 - Statik sembol gönderimini yapmak
 - Sembol örnekleme ve gönderim istatistiklerini tutmak
- eMON-Ghost ana fonksiyonun çağırılması
 - İşletim sistemi içerisinde bir ödevde
 - Zamanlayıcı fonksiyonu içerisinde
- Sembol listesi büyüdükçe gerçek zamanlı operasyon bozulabilir
 - (Observer Effect)

Sonuç

21

- Gerçek zamanda etkin gözlemlene yeteneği
 - ELF/DWARF bilgisi kullanılarak dinamik gözlem
 - XML tabanlı sembol tanımlama
- Betikleme servisleri
- Hata ayıklama ve test süresinin kısaltılması
 - (De Facto)