
ROS – Remote Operations Service

Adamis Gusztáv
(adamis@tmit.bme.hu)
Réthy György
(Gyorgy.Rethy@ericsson.com)
Ziegler Gábor
(gabor.ziegler@ericsson.com)

Példa: szendvicsautomata (1)



Szendvicset kérek



Példa: szendvicsautomata (2)



Sajtósat vagy sonkásat?



2



Sonkásat



3

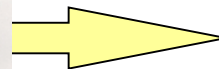
Példa: szendvicsautomata (3)



200 Ft-ot kérek

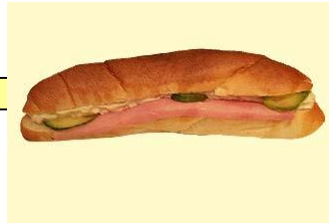
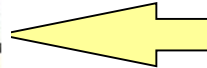


4



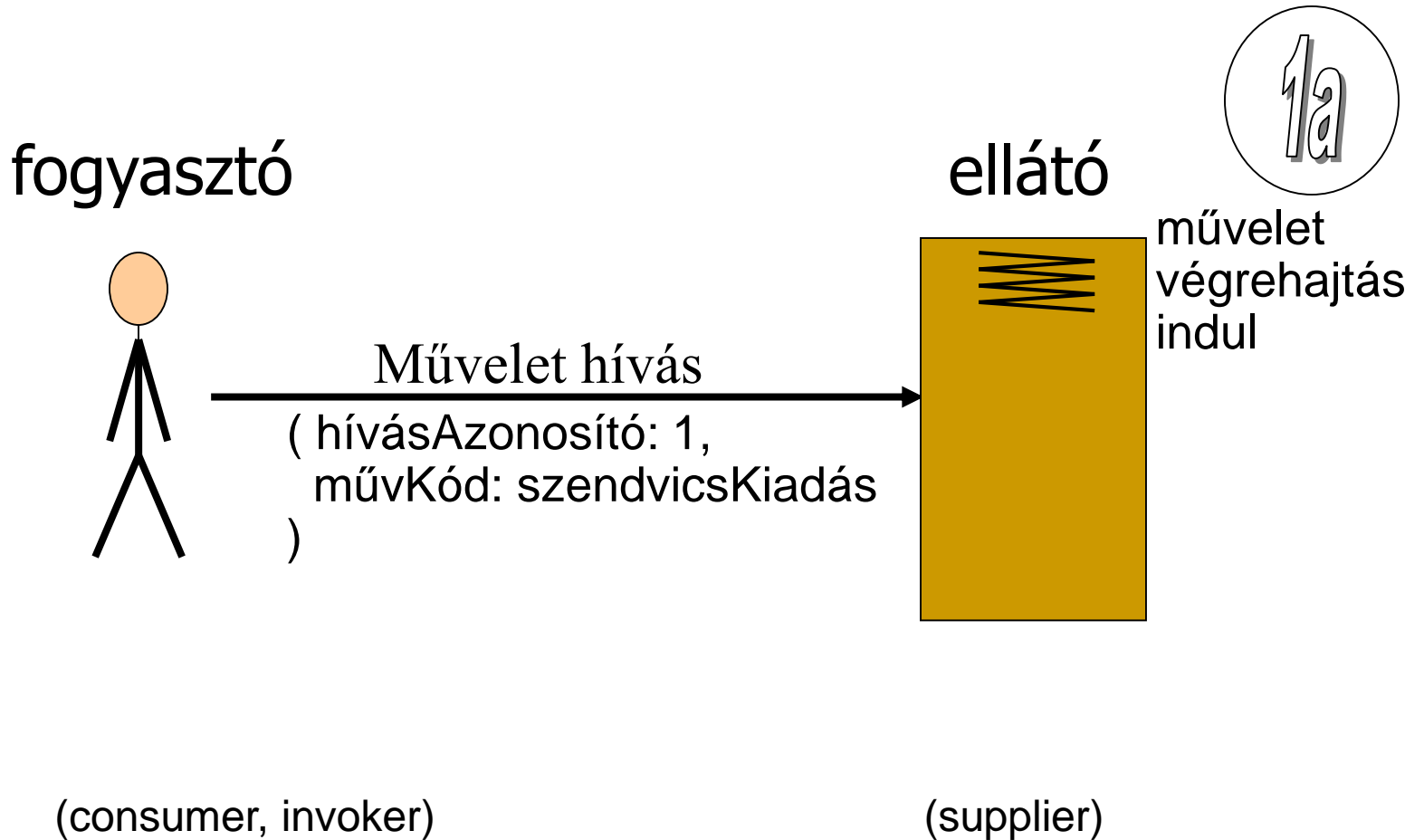
5

Példa: szendvicsautomata (4)



6

Formalizáljuk a kommunikációt: sikeres végrehajtás (1)



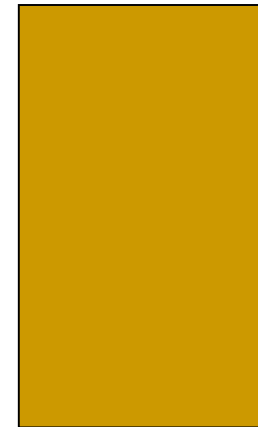
Formalizáljuk a kommunikációt: sikeres végrehajtás (2)

2a



Művelet hívás

(hívásAzonosító : 2,
művKód: szendvicsVálasztás,
paraméter:{sajtos,sonkás},
kapcsolódóHívásAzonosító: 1
)



fogyasztó

ellátó

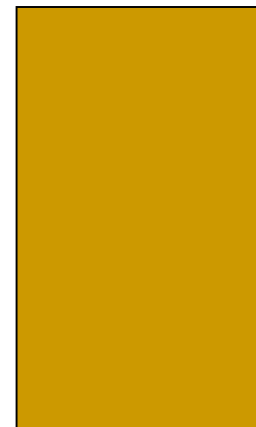
3a

művelet végrehajtás

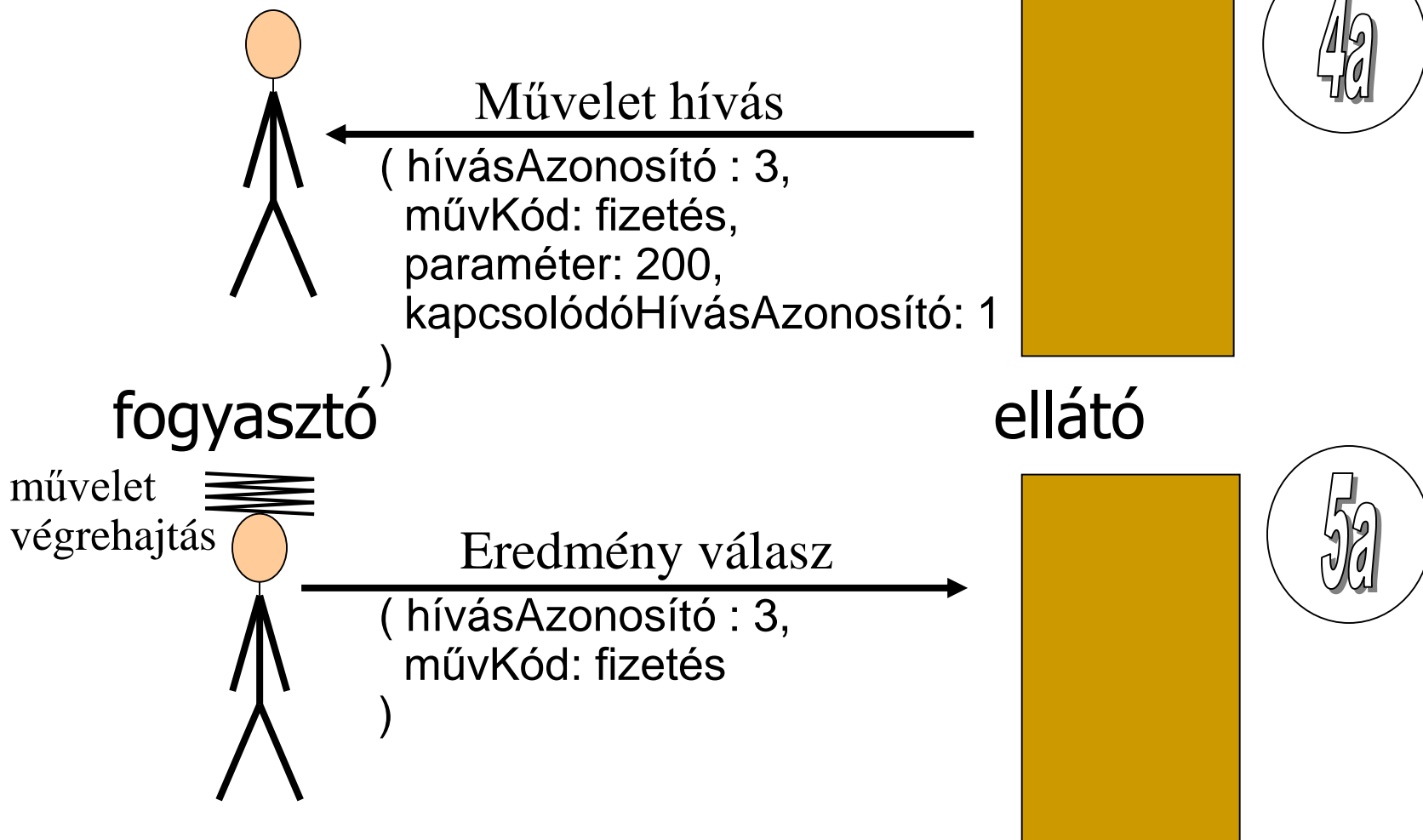


Eredmény válasz

(hívásAzonosító : 2,
művKód: szendvicsVálasztás,
paraméter: sonkás
)



Formalizáljuk a kommunikációt: sikeres végrehajtás (3)



Formalizáljuk a kommunikációt: sikeres végrehajtás (4)



Formalizáljuk a kommunikációt: az eddigiek összefoglalása

MŰVELET	műveleti kód	Hívás paraméter	Kapcsolódó művelet	Van válasz?	Eredmény paraméter
	[kód érték]	[típus]	[művelet]	[logikai érték]	[típus]
Szendvics Kiadás	szendvics Kiadás	-	{ Szendvics Választás, Fizetés }	igaz	-
Szendvics Választás	szendvics Választás	Szendvics Lista	-	igaz	szendvics ListaElem
Fizetés	fizetés	Egész szám	-	igaz	-

ROS alapelvek I.

- Operation — művelet
 - A „művelet” valamilyen funkció, amit a
 - Fogasztó (consumer, invoker) kezdeményez, és az
 - Ellátó (supplier) hajt végre
- Lásd ROS — OPERATION *információs objektum*

Formalizáljuk a kommunikációt:

Információs objektumok (IO)

osztály definíció

MŰVELET	műveleti kód	Hívás paraméter	Kapcsolódó művelet	Van válasz?	Eredmény paraméter
	[kód érték]	[típus]	[művelet]	[logikai érték]	[típus]
Szendvics Kiadás	1	-	{ Szendvics Választás, Fizetés }	igaz	-
Szendvics Választás	2	Szendvics Lista	-	igaz	szendvics ListaElem
Fizetés	3	Egész szám	-	igaz	-

Inf.Obj. Set (készlet)

példány definíció

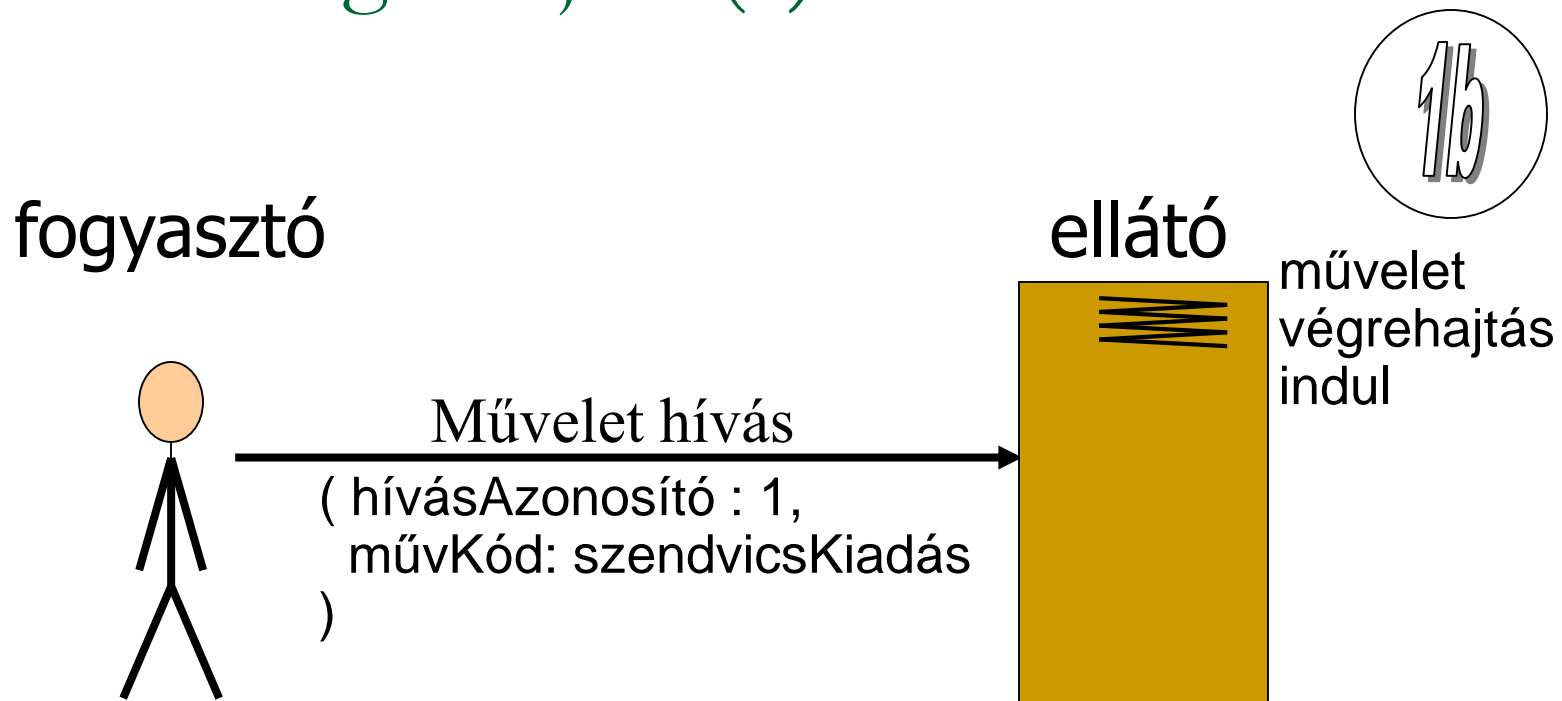
ASN.1 ROS szendvicsautomata I.

```
SzendvicsKiadas ::= OPERATION {  
    LINKED {SzendvicsValasztas, Fizetes}  
    CODE 1}
```

```
SzendvicsValasztas ::= OPERATION {  
    ARGUMENT SEQUENCE OF Szendvics,  
    RESULT Szendvics  
    CODE 2}
```

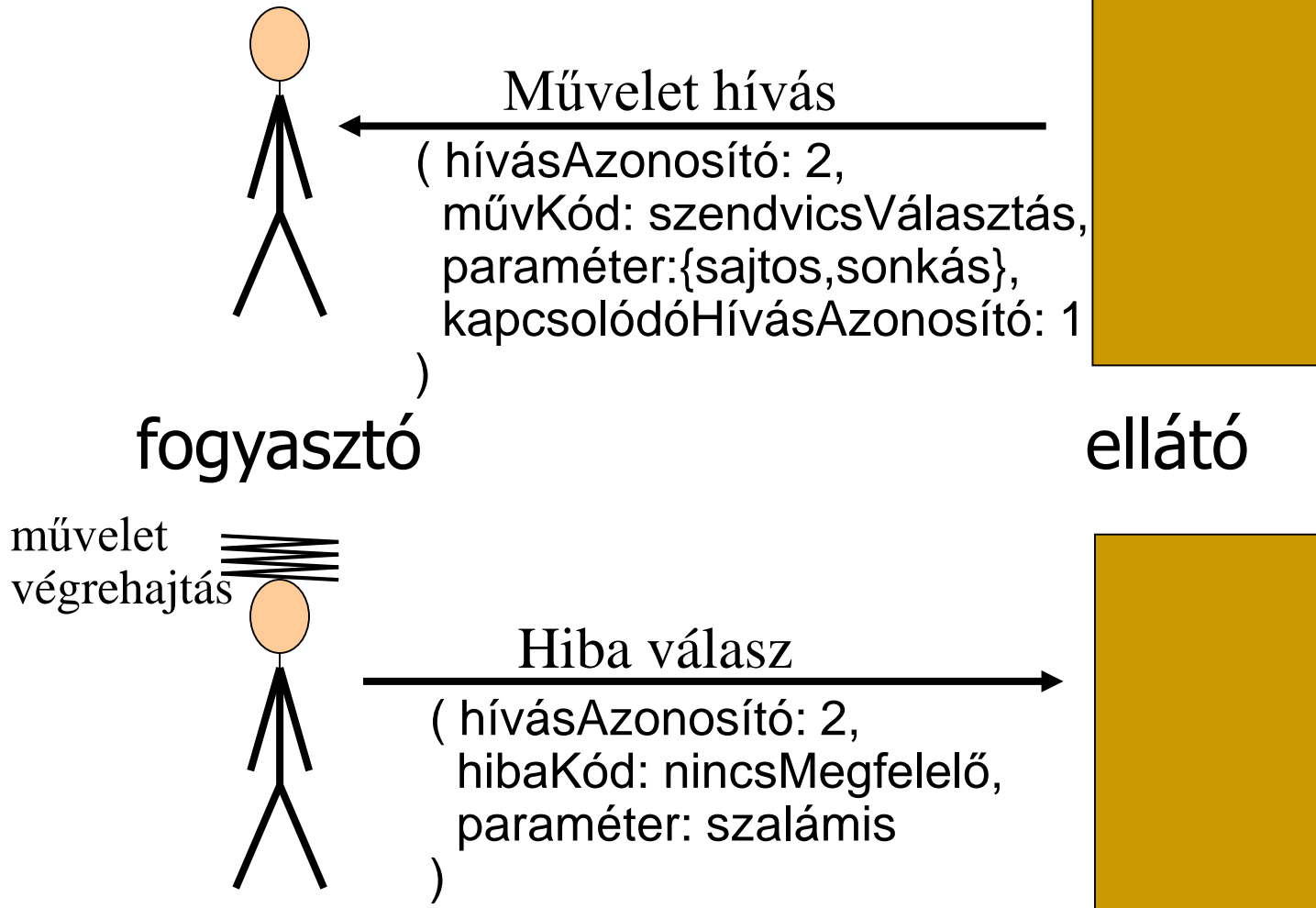
```
Fizetes ::= OPERATION {  
    ARGUMENT INTEGER  
    CODE 3}
```

Formalizáljuk a kommunikációt: sikertelen végrehajtás (1)



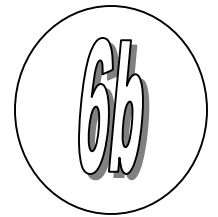
Formalizáljuk a kommunikációt: sikertelen végrehajtás (2)

2b



3b

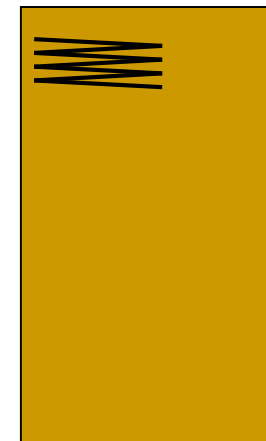
Formalizáljuk a kommunikációt: sikertelen végrehajtás (3)



fogyasztó



ellátó



Hiba válasz

(hívásAzonosító: 1,
hibaKód: kapcsolódóMűvHiba
paraméter: 2
)

A hibás művelet kódja

Formalizáljuk a kommunikációt: a szendvicsautomata műveletei: HIBA

MŰVELET	műveleti kód	Hívás paraméter	Kapcsolódó művelet	Van válasz?	Eredmény paraméter	HIBA
	[kód érték]	[típus]	[művelet]	[logikai érték]	[típus]	[művelet]
Szendvics Kiadás	1	-	{ Szendvics Választás, Fizetés }	igaz	-	{ Kapcsolódó MűvHiba, NincsVálasz, PénzKevés }
Szendvics Választás	2	Szendvics Lista	-	igaz	szendvics ListaElem	{NincsMegfelelő}
Fizetés	3	Egész szám	-	igaz	-	-

HIBA	hibakód	Hiba paraméter
	[kód érték]	[típus]
Kapcsolódó MűvHiba	1	művAzonosító
NincsVálasz	2	művAzonosító
NincsMegfelelő	3	kívántSzendvics (opcionális)
PénzKevés	4	{művAzonosító, egész szám }

(az eredménytelen kapcsolódó művelet azonosítója)

(az eredménytelen kapcsolódó művelet azonosítója)

(az eredménytelen kapcsolódó művelet azonosítója és a hiányzó összeg)

ASN.1 ROS szendvicsautomata II.

```
SzendvicsKiadas ::= OPERATION {  
    ERRORS {KapcsolodoMuvHiba,  
    NincsValasz, NincsMegfelelo,  
    PenzKeves}  
    LINKED {SzendvicsValasztas,  
    Fizetes}  
    CODE 1}
```

```
SzendvicsValasztas ::=  
    OPERATION {  
        ARGUMENT SEQUENCE OF  
        Szendvics,  
        RESULT Szendvics  
        ERRORS {NincsMegfelelo}  
        CODE 2 }
```

```
Fizetes ::= OPERATION {  
    ARGUMENT INTEGER  
    CODE 3}
```

```
KapcsolodoMuvHiba ::= ERROR {  
    PARAMETER opcode  
    CODE 1}  
NincsValasz ::= ERROR {  
    PARAMETER opcode  
    CODE 2}  
NincsMegfelelo ::= ERROR {  
    PARAMETER Szendvics OPTIONAL  
    CODE 3}  
PenzKeves ::= ERROR {  
    PARAMETER  
        SEQUENCE {  
            code opcode,  
            missingAmount INTEGER  
        }  
    CODE 4}
```

Formalizáljuk a kommunikációt: elutasítás

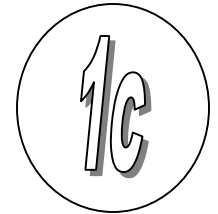
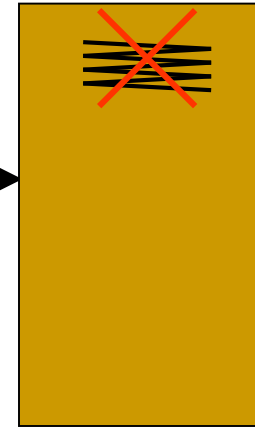
fogyasztó



Művelet hívás

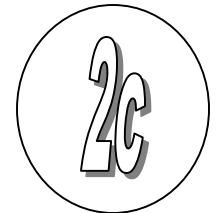
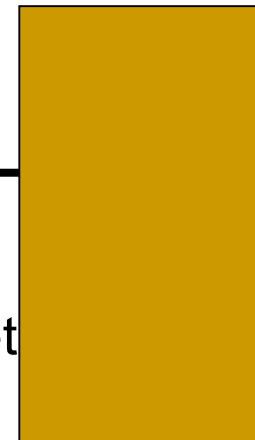
(hívásAzonosító : 1,
műv.kód: kávéKiadás
)

ellátó



Elutasítás

(hívásAzonosító: 1,
probléma: hívás
IsmeretlenMűvelet
)



ASN.1 ROS szendvicsautomata III.

```
SzendvicsKiadas ::= OPERATION {  
    ERRORS {KapcsolodoMuvHiba,  
    NincsValasz, NincsMegfelelo,  
    PenzKeves}  
    LINKED {SzendvicsValasztas,  
    Fizetes}  
    CODE 1}
```

```
SzendvicsValasztas ::=  
    OPERATION {  
    ARGUMENT SEQUENCE OF  
    Szendvics,  
    RESULT Szendvics  
    ERRORS {NincsMegfelelo}  
    CODE 2 }
```

```
Fizetes ::= OPERATION {  
    ARGUMENT INTEGER  
    CODE 3}
```

```
Elutasitas ::= OPERATION
```

```
KapcsolodoMuvHiba ::= ERROR {  
    PARAMETER opcode  
    CODE 1}  
NincsValasz ::= ERROR {  
    PARAMETER opcode  
    CODE 2}  
NincsMegfelelo ::= ERROR {  
    PARAMETER Szendvics OPTIONAL  
    CODE 3}  
PenzKeves ::= ERROR {  
    PARAMETER  
        SEQUENCE {  
            code opcode,  
            missingAmount INTEGER  
        }  
    CODE 4}
```

Művelet-csomomagok

	Művelet	
mindkettő hívhatja	nincs ilyen	nincs ilyen
a műveletet a fogyasztó hívja	Szendvicskiadás	Italkiadás
a műveletet az ellátó hívja	Szendvicsválasztás, Fizetés	Italválasztás, Fizetés
azonosító	szendvicsAutomata	italAutomata

SzendvicsAutomata
művelet-csomag

ItalAutomata
művelet-csomag

ROS alapelvek II.

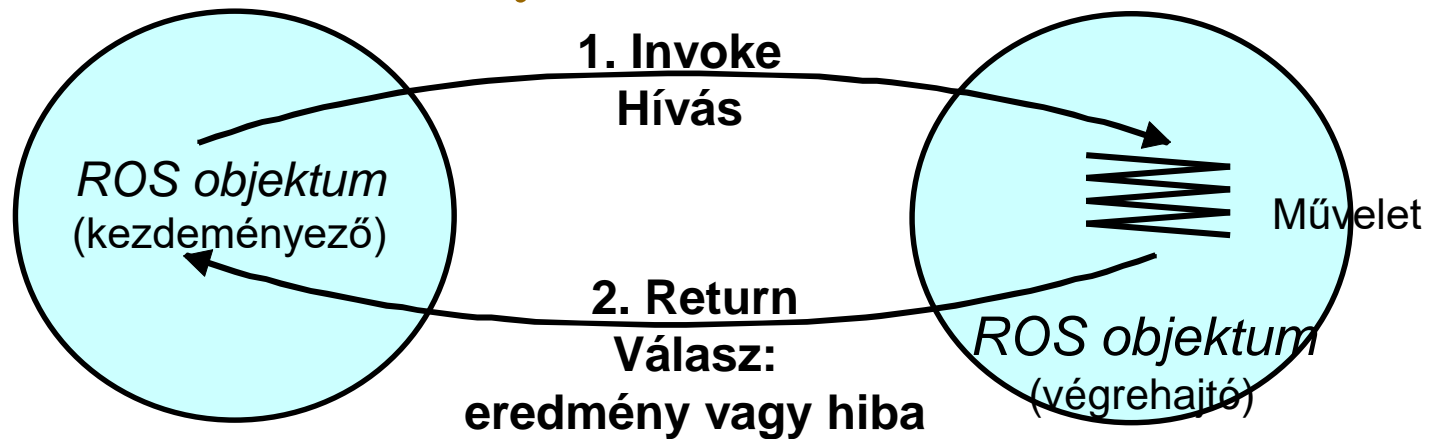
Művelet-csomagok

- A művelet csomag szerepeket definiál
 - Kommunikáló objektumpárok között
 - Műveletek segítségével
 - Amelyeket a páros tagjai egymástól igénybevehetnek
 - Ha a szerepek aszimmetrikusak
 - Akkor az egyiket (mindegy, melyik egyiket!) „fogyasztó” (*CONSUMER*),
 - A másikat (mindegy, melyik másikat!) az „ellátó” (*SUPPLIER*) nevekkel illetjük őket

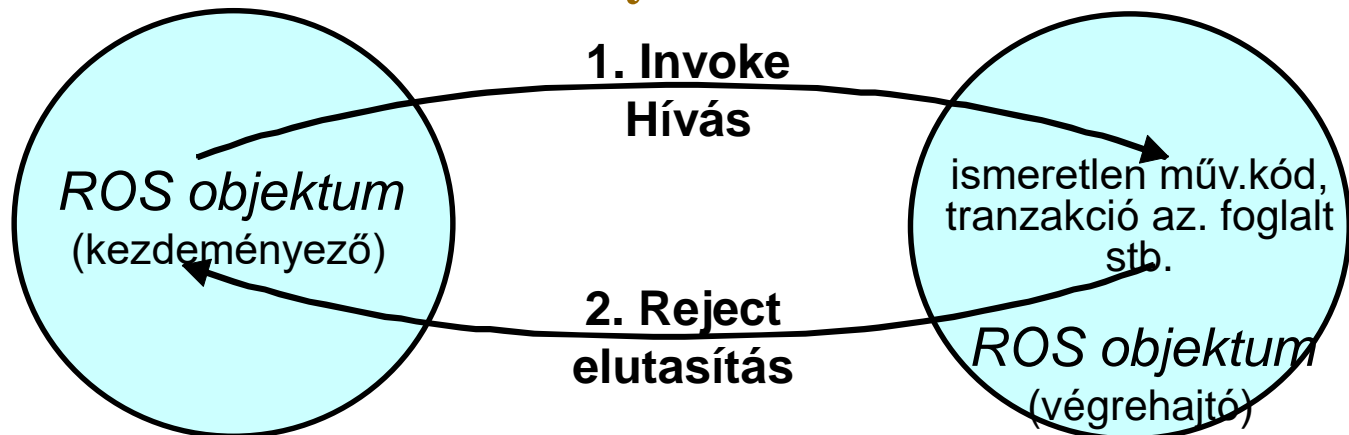
A ROS koncepció (ITU-T X.880)

ROS(E) - Remote Operations Service (Element) - távoli művelet szolgálat(elem)

1. Sikeres művelet kezdeményezés

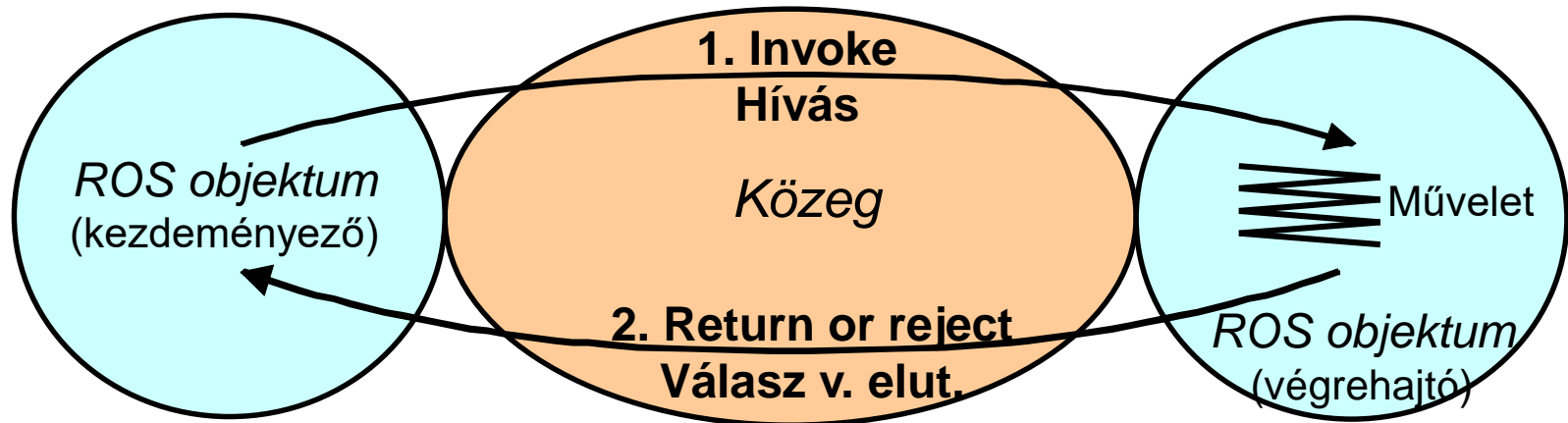


2. Sikertelen művelet kezdeményezés

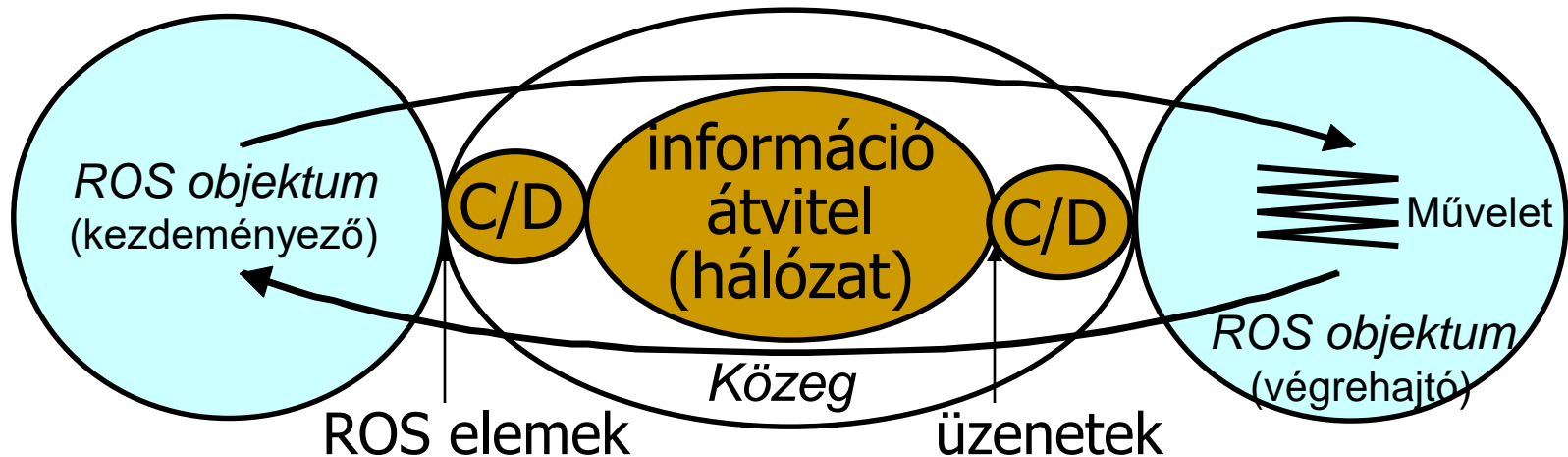


ROS megvalósulás

1. Megvalósulása gépen belül vagy helyi hálózaton



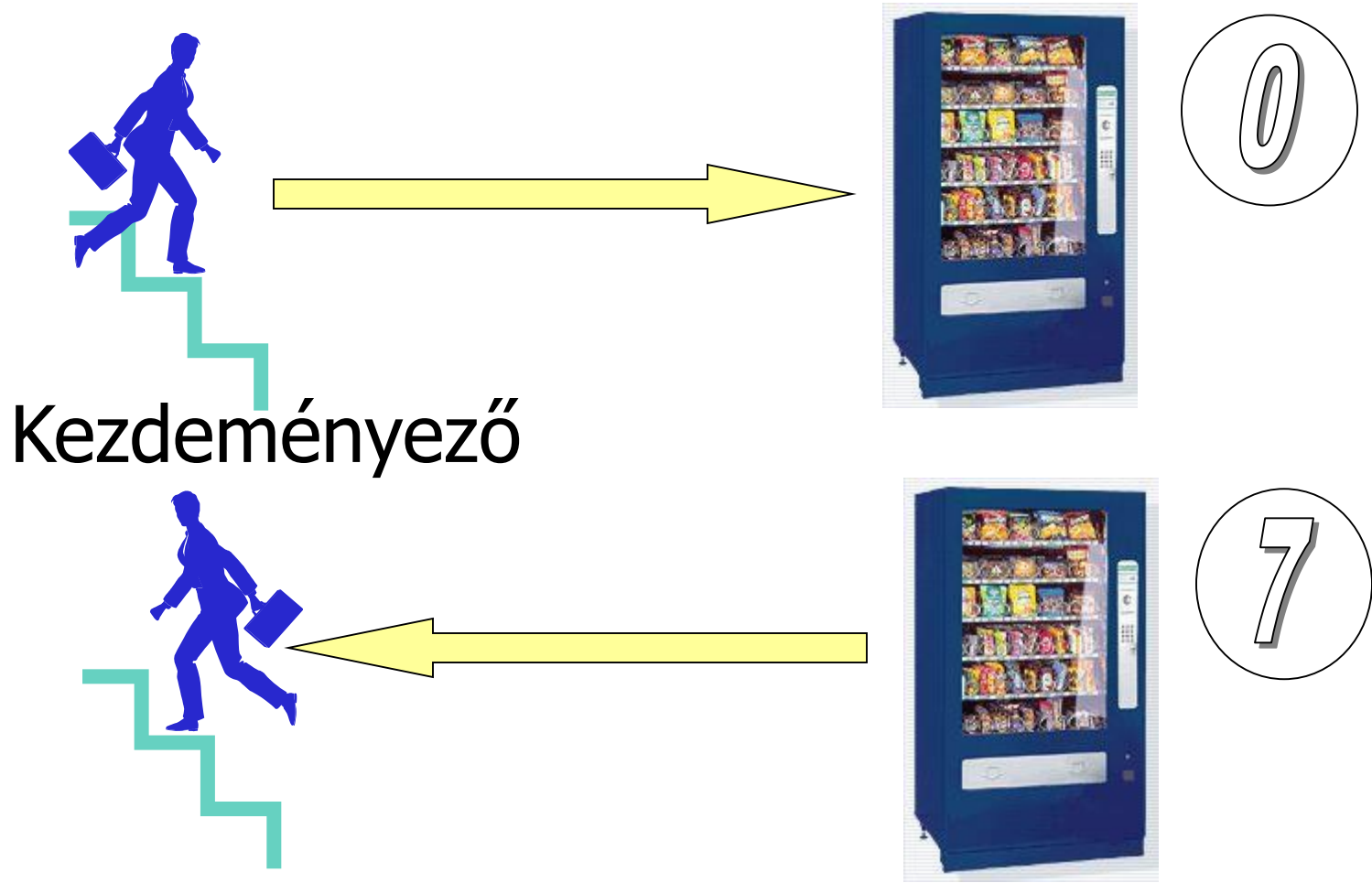
2. Megvalósulása kommunikációs hálózaton



ROS alapelvek III — asszociáció

- Az *asszociáció (társítás)* kapcsolat két objektum között,
 - ami kontextust ad műveletek igénybevételéhez és végrehajtásához

Példa: Asszociáció (társítás) létrejötte és oldása



ROS alapelvek IV — kapcsolati csomag

- A kapcsolati csomag (CONNECTION PACKAGE)
 - Megadja az asszociáció...
 - Dinamikus létrehozásának és lebontásának műveleteit és
 - A hozzá kapcsolódó QoS-t

Az asszociáció létrejötte - a kapcsolat-csomag

	Művelet
kötés (bind)	Érkezés
oldás (unbind)	Távozás
Válaszoló oldhat-e?	igaz
oldás lehet sikertelen?	hamis
azonosító	automatázás

az asszociáció
kezdeményezője

pl. működési zavar
("se kép se hang")

ROS alapelvek IV — asszociációs szerződés

- Az asszociációs szerződés szabályozza, hogy
 - Melyik ROS objektum
 - Kivel „társulhat” ...

Asszociációs szerződés

	Csomagok	
A kezdeményező az egyes műveletekben lehet fogyasztó vagy ellátó	nem	nem
A kezdeményező mindig fogyasztó	SzendvicsAutomata	ItalAutomata
A kezdeményező mindig ellátó	nem	nem
kapcsolat	automatázás	automatázás
azonosító	éhségoltás	szomjoltás

SzendvicsÉvés
szerződés

Kávészomjoltás
szerződés

ROS-objektumok



Szerződések

Kezdeményez	SzendvicsEvés, KávéIvás	-	-	-
Reagál	-	SzendvicsEvés	KávéIvás	SzendvicsEvés, KávéIvás
Kezdeményez és reagál	-	-	-	-
azonosító	egyetemista	kajaGép	kávéGép	kajaKávéGép

Információs objektumok az ASN.1-ben

- X.681 – Information object specification

Formalizáljuk a kommunikációt:

Információs objektumok (IO)

osztály definíció

MŰVELET	műveleti kód	Hívás paraméter	Kapcsolódó művelet	Van válasz?	Eredmény paraméter
	[kód érték]	[típus]	[művelet]	[logikai érték]	[típus]
Szendvics Kiadás	1	-	{ Szendvics Választás, Fizetés }	igaz	-
Szendvics Választás	2	Szendvics Lista	-	igaz	szendvics ListaElem
Fizetés	3	Egész szám	-	igaz	-

Inf.Obj. Set (készlet)

példány definíció

Information object class – field types

**CAPITAL letter,
“-” and digits**

keyword CLASS

```
MY-CLASS-1 ::= CLASS {  
  &TypeField, -- OPTIONAL | DEFAULT <ASN1-Type>  
    -- accepts: any ASN.1 type  
  &fixedTypeValueField ASN1-Type, -- UNIQUE OPTIONAL | DEFAULT <value>  
    -- accepts: a value of the specified type/subtype; UNIQUE -> if the field type is used as an  
    -- identifier field: assigned values shall be unique within any ObjectSet  
  &variableTypeValueField &TypeField, -- OPTIONAL | DEFAULT <value>  
    -- accepts: a value of the type of the referenced field; if the referenced TypeField is OPTINAL,  
    -- shall be OPTIONAL, if has DEFAULT, shall have a DEFAULT of the same type  
  &FixedTypeValueSetField ASN1-Type, -- OPTIONAL | DEFAULT <ValueSet>  
    -- accepts: a ValueSet of the specified type/subtype  
  &VariableTypeValueSetField &TypeField, -- OPTIONAL | DEFAULT <ValueSet>  
    -- accepts: a ValueSet of the type of the referenced field; if the referenced TypeField is OPTINAL,  
    -- shall be OPTIONAL, if has DEFAULT, shall have a DEFAULT as a subset of that  
  &objectField OBJECTCLASS, -- OPTIONAL | DEFAULT <object>  
    -- accepts: an object of the specified class  
  &ObjectSetField OBJECTCLASS -- OPTIONAL | DEFAULT <ObjectSet>  
    -- accepts: an ObjectSet of the specified class (it may consist a single object!)  
}
```

**field name must start
with ampersand (&)**

Példák - Osztálydefiníció

```
OPERATION ::= CLASS
```

```
{  
    &ArgumentType OPTIONAL,  
    &ResultType OPTIONAL,  
    &Errors ERROR OPTIONAL,  
    &Linked OPERATION OPTIONAL,  
    &resultReturned BOOLEAN DEFAULT TRUE,  
    &code INTEGER UNIQUE  
}
```

```
ERROR ::= CLASS
```

```
{  
    &ParameterType OPTIONAL,  
    &code INTEGER UNIQUE  
}
```

Példák – Szintaxis megadás

```
OPERATION ::= CLASS
{
    &ArgumentType    OPTIONAL,
    &ResultType      OPTIONAL,
    &Errors           ERROR OPTIONAL,
    &Linked           OPERATION OPTIONAL,
    &resultReturned  BOOLEAN DEFAULT TRUE,
    &operationCode   INTEGER UNIQUE
}
WITH SYNTAX
{
    [ARGUMENT &ArgumentType]
    [RESULT &ResultType]
    [RETURN RESULT &resultReturned]
    [ERRORS &Errors]
    [LINKED &Linked]
    CODE &operationCode
}
}
```

Példák – Szintaxis megadás

ERROR ::= CLASS

{

&ParameterType

OPTIONAL,

&errorCode

INTEGER UNIQUE

}

WITH SYNTAX

{

[PARAMETER

&ParameterType]

CODE

&errorCode

}

Formalizáljuk a kommunikációt:

Információs objektumok (IO)

osztály definíció

MŰVELET	műveleti kód	Hívás paraméter	Kapcsolódó művelet	Van válasz?	Eredmény paraméter
	[kód érték]	[típus]	[művelet]	[logikai érték]	[típus]
Szendvics Kiadás	1	-	{ Szendvics Választás, Fizetés }	igaz	-
Szendvics Választás	2	Szendvics Lista	-	igaz	szendvics ListaElem
Fizetés	3	Egész szám	-	igaz	-

Inf.Obj. Set (készlet)

példány definíció

Példák – Példányosítás (default syntax)

```
invertMatrix OPERATION ::=
{
    &ArgumentType Matrix,
    &ResultType Matrix,
    &Errors {determinantIsZero},
    &operationCode 7
}
```

```
determinantIsZero ERROR ::=
{
    &errorCode 1
}
```

Példák – Példányosítás (user defined syntax)

```
invertMatrix OPERATION ::=  
  {  
    ARGUMENT Matrix  
    RESULT Matrix  
    ERRORS {determinantIsZero}  
    CODE 7  
  }  
determinantIsZero ERROR ::=  
  {  
    CODE 1  
  }
```


Formalizáljuk a kommunikációt:

Információs objektumok (IO)

osztály definíció

MŰVELET	műveleti kód	Hívás paraméter	Kapcsolódó művelet	Van válasz?	Eredmény paraméter
	[kód érték]	[típus]	[művelet]	[logikai érték]	[típus]
Szendvics Kiadás	1	-	{ Szendvics Választás, Fizetés }	igaz	-
Szendvics Választás	2	Szendvics Lista	-	igaz	szendvics ListaElem
Fizetés	3	Egész szám	-	igaz	-

Inf.Obj. Set (készlet)

példány definíció

Példák – Information object set

```
MatrixOperations OPERATION ::=  
{  
  invertMatrix  
| addMatrices  
| subtractMatrices  
| multiplyMatrices  
}
```

Példák - Mezőhivatkozások

"ValueFromObject"

<code>invertMatrix.&operationCode</code>	7
<code>determinantIsZero.&errorCode</code>	1

"TypeFromObject"

<code>invertMatrix.&ArgumentType</code>	Matrix
---	--------

"ValueSetFromObjects"

<code>invertMatrix.&Errors.&errorCode</code>	{ 1 }
<code>MatrixOperations.&operationCode</code>	{7 <i>and others</i> }

"ObjectSetFromObjects"

<code>invertMatrix.&Errors</code>	{determinantIsZero}
<code>MatrixOperations.&Errors</code>	{determinantIsZero <i>and others</i> }

Komplex példa - OPERATION

```
OPERATION ::= CLASS
{
    &ArgumentType    OPTIONAL,
    &ResultType      OPTIONAL,
    &Errors           ERROR OPTIONAL,
    &Linked           OPERATION OPTIONAL,
    &resultReturned  BOOLEAN DEFAULT TRUE,
    &operationCode   INTEGER UNIQUE
}
WITH SYNTAX
{
    [ARGUMENT &ArgumentType]
    [RESULT &ResultType]
    [RETURN RESULT &resultReturned]
    [ERRORS &Errors]
    [LINKED &Linked]
    CODE &operationCode
}
}
```

Komplex példa - ERROR

ERROR ::= CLASS

{

&ParameterType

OPTIONAL,

&errorCode

INTEGER UNIQUE

}

WITH SYNTAX

{

[PARAMETER

&ParameterType]

CODE

&errorCode

}

Komplex példa – object set

2 OPERATION-nel

```
My-Operations OPERATION ::= { operationA | operationB }
```

```
operationA OPERATION ::= {  
  ARGUMENT INTEGER  
  ERRORS { { PARAMETER INTEGER CODE 1000 } | { CODE 1001 } }  
  CODE 1  
}
```

```
operationB OPERATION ::= {  
  ARGUMENT IA5String  
  RESULT BOOLEAN  
  ERRORS { { CODE 1002 } | { PARAMETER IA5String CODE 1003 } }  
  CODE 2  
}
```

ERROR-ral definiálhatnánk...



- Extraction of the set of the **ERROR** objects from the object set above is done as follows:

```
My-OperationErrors ERROR ::=
                               { My-Operations.&Errors }
```

- The resulting object set is:

```
My-OperationErrors ERROR ::= {
    { PARAMETER INTEGER CODE 1000 } |
    { CODE 1001 } |
    { CODE 1002 } |
    { PARAMETER IA5String CODE 1003 }
}
```

-
- Extraction of the set of error codes of the errors of the operations is done as follows:

```
My-OperationErrorCodes INTEGER ::=
    { My-Operations.&Errors.&errorCode }
```

- The resulting value set is:

```
My-OperationErrorCodes INTEGER ::=
    { 1000 | 1001 | 1002 | 1003 }
```