



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΡΕΥΝΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΕΠΕΤ ΙΙ

ΕΡΓΟ: 98ΑΜΕΑ 19

ΑΙΝΕΙΑΣ: ΑΝΑΠΤΥΞΗ ΕΥΕΛΙΚΤΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΕΝΑΛΛΑΚΤΙΚΗΣ ΚΑΙ ΕΠΑΥΞΗΤΙΚΗΣ
ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕΣΩ ΥΠΟΛΟΓΙΣΤΩΝ
ΚΑΙ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ

Παραδοτέο Π4.2α

Υλοποίηση Πιλοτικού Αυτόνομου Βοηθήματος
Διαπροσωπικής Επικοινωνίας:

Επεκτάσεις του πλαισίου ΟΔΥΣΣΕΑΣ

Αλέξανδρου Πίνο και Γεωργίου Κουρουπέτρογλου

ΑΘΗΝΑ

15 ΜΑΡΤΙΟΥ 2001

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΡΕΥΝΑΣ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ
ΕΠΕΤ ΙΙ

ΕΡΓΟ: 98ΑΜΕΑ 19

**ΑΙΝΕΙΑΣ: ΑΝΑΠΤΥΞΗ ΕΥΕΛΙΚΤΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΕΝΑΛΛΑΚΤΙΚΗΣ ΚΑΙ ΕΠΑΥΞΗΤΙΚΗΣ
ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕΣΩ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ
ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ**

Ανάδοχος φορέας: Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Συνεργαζόμενοι φορείς: 01 ΠΛΗΡΟΦΟΡΙΚΗ

Κέντρο Αποκατάστασης Ατόμων με Ειδικές Ανάγκες
(Κ.Α.Α.Μ.Ε.Α.) Σερρών

Επιστημονικός Υπεύθυνος: Γεώργιος Κουρουπέτρογλου
Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής,
Πανεπιστημιούπολη, Ιλίσια,
Αθήνα 15784

Τηλέφωνο: (01) 7275305

Fax: (01) 6018677

Ηλεκτρονικό Ταχυδρομείο: koupe@di.uoa.gr

Περιεχόμενα

Περιεχόμενα	3
1. Εισαγωγή.....	5
2. Για τους Τελικούς Χρήστες (Θεραπευτές – ΑΜΕΑ).....	6
2.1. Εγκατάσταση της εφαρμογής.....	6
2.2. Εκκίνηση της εφαρμογής	7
2.3. Ρυθμίσεις των τεχνικών σάρωσης.....	7
3. Για τους Πωλητές.....	10
3.1. Σύνθεση εφαρμογών	10
3.2. Προετοιμασία CD-ROM εγκατάστασης εφαρμογών.....	13
4. Για τους Προγραμματιστές.....	15
4.1. Αρχείο ρυθμίσεων (Settings.ini)	15
4.2. Λογισμικό διαχείρισης αρχείου ρυθμίσεων (IniEditor.dll).....	18
4.3. Λογισμικό ανάλυσης αλφαριθμητικών σειρών (StringUtil.dll).....	20
4.4. Λογισμικό ανάκτησης του Class ID (ClassIDUtil.dll).....	21
4.5. Η προγραμματιστική διεπαφή ελέγχου (Interface.Configuration).....	22
4.6. Συστατικό σάρωσης	26
4.7. Οι κλάσεις των συστατικών (Activate, Scannable, Hidden).....	33
4.8. Παράδειγμα υλοποίησης και κώδικας αναφοράς.....	35
Αναφορές	60

1. Εισαγωγή

Κατά τη διάρκεια της ανάπτυξης και υλοποίησης των βοηθημάτων διαπροσωπικής επικοινωνίας, προέκυψαν νέες απαιτήσεις για τις δυνατότητες και τα χαρακτηριστικά του πλαισίου «ΟΔΥΣΣΕΑΣ» [4], [5]. Το πλαίσιο εμπλουτίστηκε με νέα τμήματα κώδικα, νέες βιβλιοθήκες (DLLs) και νέες προγραμματιστικές διεπαφές. Επίσης προστέθηκαν νέες λειτουργίες και δυνατότητες στο πρόγραμμα εκτέλεσης των εφαρμογών που βασίζονται στο πλαίσιο και επεκτάθηκε η χρήση του εκτελέσιμου αυτού αρχείου, τόσο για τους τελικούς χρήστες όσο και για τους πωλητές ολοκληρωμένων συστημάτων επικοινωνίας. Τέλος ενσωματώθηκαν οι τεχνικές σάρωσης και έγιναν τροποποιήσεις στη διεπαφή χρήσης. Οι τελικές εκδόσεις του πλαισίου «ΟΔΥΣΣΕΑΣ» και ειδικότερα οι επεκτάσεις και του λογισμικού και των προδιαγραφών που το αποτελούν θα παρουσιαστούν σε αυτό το κεφάλαιο. Επειδή το πλαίσιο με όλα τα συστατικά του απευθύνεται σε τρεις ομάδες χρηστών θα περιγραφεί από τρεις διαφορετικές πλευρές, αυτές δηλαδή που αφορούν τους τελικούς χρήστες (Θεραπευτές – ΑΜΕΑ, τους πωλητές και τους προγραμματιστές.

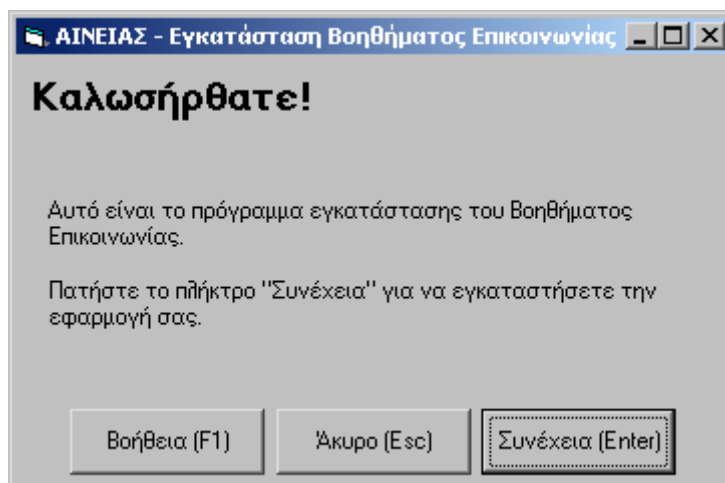
Όσον αφορά τους τελικούς χρήστες (ΑΜΕΑ ή θεραπευτές) οι επεκτάσεις αφορούν κυρίως τη διεπαφή χρήσης και τις ρυθμίσεις της σάρωσης. Για του πωλητές τροποποιήθηκε τόσο ο τρόπος σύνθεσης όσο και ο τρόπος διανομής των ολοκληρωμένων εφαρμογών με τη βοήθεια προγραμμάτων εγκατάστασης και γρήγορης ρύθμισης για τους πωλητές.

Ο κύριος όγκος της πληροφορίας αφορά τους προγραμματιστές οι οποίοι πρέπει να συμμορφωθούν με νέες προδιαγραφές και κανόνες. Το παράδειγμα σχεδιασμού και υλοποίησης ενός συστατικού που υποστηρίζει το πλαίσιο ΟΔΥΣΣΕΑΣ στο τέλος της τεχνικής έκθεσης βοηθά με τον καλύτερο τρόπο στην κατανόηση και τη διαδικασία συμμόρφωσης με τον ΟΔΥΣΣΕΑ.

2. Για τους Τελικούς Χρήστες (Θεραπευτές – ΑΜΕΑ)

Οι τελικοί χρήστες θα πρέπει σύμφωνα με το μοντέλο του πλαισίου ΟΔΥΣΣΕΑΣ να προμηθεύονται ένα CD εγκατάστασης από τον πωλητή βοηθημάτων διαπροσωπικής επικοινωνίας, το οποίο θα περιέχει την εφαρμογή που συντέθηκε από τον πωλητή σύμφωνα με τις ανάγκες τους. Θα περιγραφούν στη συνέχεια τα βήματα εγκατάστασης της εφαρμογής στον προσωπικό υπολογιστή του χρήστη, καθώς και η διαδικασία εκκίνησης της εφαρμογής. Τέλος θα αναφερθούν όλες οι ρυθμίσεις του πλαισίου ΟΔΥΣΣΕΑΣ που είναι διαθέσιμες στους χρήστες. Στις περισσότερες περιπτώσεις, οι χρήστες των αποτελεσμάτων του έργου ΑΙΝΕΙΑΣ, δηλαδή τα άτομα με ειδικές ανάγκες δεν θα είναι σε θέση να εκτελέσουν αυτές τις διαδικασίες, οπότε οι ενέργειες θα εκτελούνται από τους θεραπευτές τους.

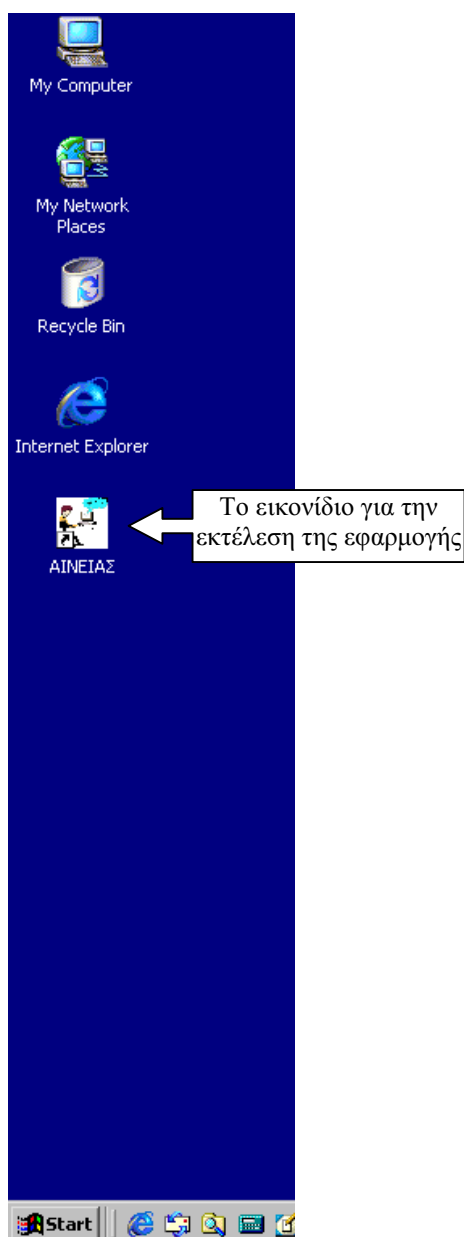
2.1. Εγκατάσταση της εφαρμογής



Σχήμα 1: Οθόνη εισαγωγής

Αφού ο χρήστης (ή ο θεραπευτής) ανοίξει τον υπολογιστή του και βάλει το CD-ROM εγκατάστασης στον οδηγό CD-ROM θα πρέπει να εκτελέσει το πρόγραμμα εγκατάστασης `install.exe` που βρίσκεται στον κύριο κατάλογο του CD-ROM. Αυτό το πρόγραμμα θα εγκαταστήσει την εφαρμογή στα Component Services και θα αντιγράψει όλα τα αναγκαία αρχεία στο δίσκο του χρήστη και συγκεκριμένα στον κατάλογο `C:\Program Files\Communicator`. Τέλος θα δημιουργηθεί μία συντόμευση στην επιφάνεια εργασίας με το όνομα ΑΙΝΕΙΑΣ η οποία με διπλό κλικ θα εκτελεί την εφαρμογή.

2.2. Εκκίνηση της εφαρμογής



Οι τελικοί χρήστες μπορούν εκτελούν το βοήθημα διαπροσωπικής επικοινωνίας πατώντας το εικονίδιο του έργου ΑΙΝΕΙΑΣ με διάφορους τρόπους. Εκτός από τον τυπικό τρόπο εκτέλεσης ενός προγράμματος που βρίσκεται κάπου στον σκληρό δίσκο του υπολογιστή και απαιτεί την εύρεση και εκτέλεση ενός εκτελέσιμου προγράμματος μέσω της επιλογής Start>>Run ή από τον Windows Explorer, το εκτελέσιμο αρχείο (Communicator.exe) ή κάποια συντόμευση προς αυτό, μετά την εγκατάσταση της εφαρμογής μπορεί να βρίσκεται στα εξής σημεία:

1. Στο μενού εκκίνησης των Windows. Στην περίπτωση αυτή με κάθε εκκίνηση του υπολογιστή θα εκτελείται αυτόματα η εφαρμογή. Για να γίνει αυτό θα πρέπει ο θεραπευτής να βάλει χειροκίνητα μια συντόμευση του εκτελέσιμου αρχείου στον κατάλογο εκκίνησης.
2. Στην επιφάνεια εργασίας. Τότε θα πρέπει ο ίδιος ο χρήστης ή ο θεραπευτής του να εκτελεί την εφαρμογή με διπλό κλικ επάνω στο εικονίδιο (βλέπε διπλανό Σχήμα).

Σχήμα 2: Το εικονίδιο του βοηθήματος διαπροσωπικής επικοινωνίας στην επιφάνεια εργασίας των Windows.

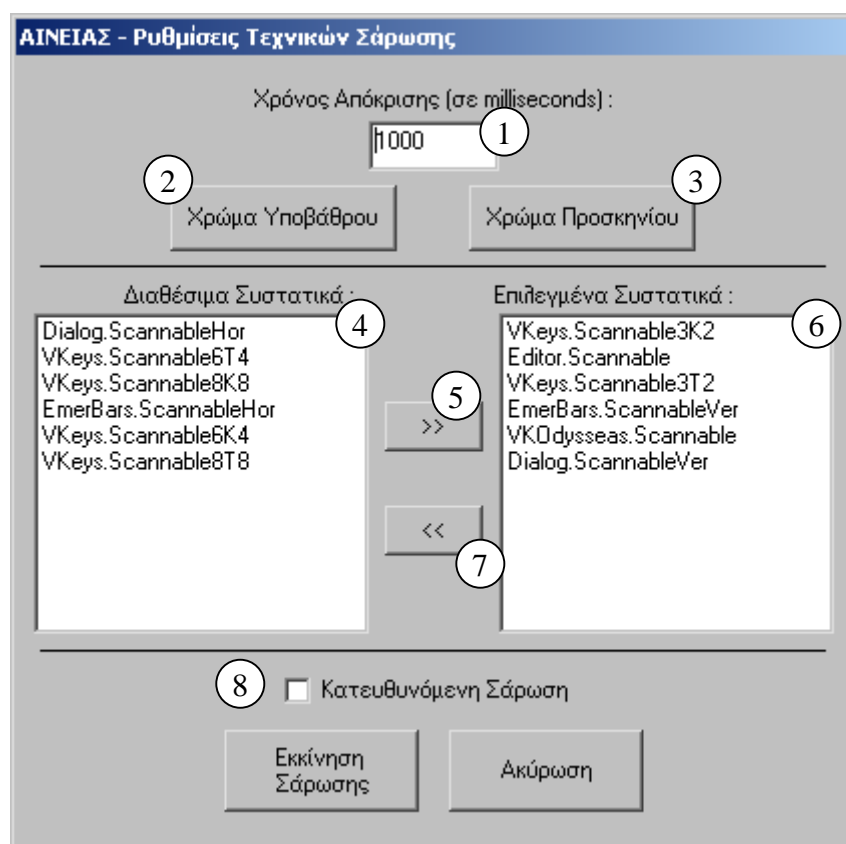
2.3. Ρυθμίσεις των τεχνικών σάρωσης

Το συστατικό σάρωσης που ενσωματώθηκε στο πλαίσιο ΟΔΥΣΣΕΑΣ έχει τις εξής ρυθμίσεις διαθέσιμες για τους χρήστες και τους θεραπευτές τους:

Ρύθμιση	Περιγραφή (βλέπε Σχήμα 3)
Χρόνος Απόκρισης	Το χρονικό διάστημα για το οποίο θα μένει ο φωτισμός της σάρωσης σε κάθε στοιχείο αλληλεπίδρασης. Όσο πιο καλές είναι οι κινητικές ικανότητες του χρήστη και τα αντανακλαστικά του τόσο πιο μικρός θα είναι αυτός ο χρόνος. Μπορεί να ρυθμιστεί από 100 msec (0,1 sec) έως και 10000 msec (10 sec). Αν ο χρήστης ή ο θεραπευτής θέλει να αλλάξει αυτήν τη ρύθμιση η οποία χοντρικά καθορίζει την ταχύτητα της σάρωσης αρκεί να πληκτρολογήσει τον αριθμό σε milliseconds που επιθυμεί στο αντίστοιχο πλαίσιο κειμένου (1).
Χρώμα υποβάθρου	Η ρύθμιση αυτή καθορίζει το χρώμα που θα έχει το υπόβαθρο (το φόντο) όλων των διεπαφών χρήσης. Αυτό το χρώμα θα έχουν γύρω τους όλα τα στοιχεία αλληλεπίδρασης που δεν είναι φωτισμένα. Με την πίεση του πλήκτρου (2) εμφανίζεται μία φόρμα επιλογής χρώματος. Ο χρήστης επιλέγει το χρώμα και πατά OK.
Χρώμα προσκηνίου	Η ρύθμιση αυτή καθορίζει το χρώμα που θα έχει το προσκηνίο (ο φωτισμός) όλων των διεπαφών χρήσης. Αυτό το χρώμα θα έχουν γύρω τους όλα τα στοιχεία αλληλεπίδρασης που είναι φωτισμένα. Με την πίεση του πλήκτρου (3) εμφανίζεται μία φόρμα επιλογής χρώματος. Ο χρήστης επιλέγει το χρώμα και πατά OK.
Διαθέσιμα συστατικά	Σε αυτήν τη λίστα (4) φαίνονται όλα τα συστατικά που είναι διαθέσιμα στο βοήθημα επικοινωνίας. Συνίσταται να μην γίνεται αλλαγή των ρυθμίσεων αυτής της λίστας από τον χρήστη ή τον θεραπευτή χωρίς την καθοδήγηση του πωλητή. Τα συστατικά αυτής της λίστας δεν είναι ενεργά. Για να ενεργοποιηθούν θα πρέπει να επιλεγούν με το ποντίκι και να πατηθεί το πλήκτρο >> (5) ώστε να περάσουν στη διπλανή λίστα των επιλεγμένων συστατικών.
Επιλεγμένα συστατικά	Σε αυτήν τη λίστα (6) φαίνονται όλα τα συστατικά που είναι επιλεγμένα στο βοήθημα επικοινωνίας. Συνίσταται να μην γίνεται αλλαγή των ρυθμίσεων αυτής της λίστας από τον χρήστη ή τον θεραπευτή χωρίς την καθοδήγηση του πωλητή. Τα συστατικά αυτής της λίστας είναι ενεργά. Για να απενεργοποιηθούν θα πρέπει να επιλεγούν με το ποντίκι και να πατηθεί το πλήκτρο << (7) ώστε να περάσουν στη διπλανή λίστα των διαθέσιμων συστατικών.
Κατευθυνόμενη σάρωση	Η ρύθμιση αυτή καθορίζει το είδος της τεχνικής σάρωσης που θα χρησιμοποιηθεί. Όπως αναφέρεται και στη συνέχεια της παρούσας τεχνικής έκθεσης οι τεχνικές σάρωσης που υποστηρίζονται είναι η αυτόματη και η κατευθυνόμενη. Ο χρήστης μπορεί να επιλέξει αυτόματη σάρωση όταν το τετραγωνάκι (8) δεν είναι τσεκαρισμένο και κατευθυνόμενη όταν είναι τσεκαρισμένο.

Η διεπαφή χρήσης για τις ρυθμίσεις της σάρωσης εμφανίζεται όταν κατά τη λειτουργία του βοηθήματος διαπροσωπικής επικοινωνίας πατηθεί το πλήκτρο F12. Αφού ο χρήστης κάνει τις ρυθμίσεις του θα πρέπει να πατήσει το πλήκτρο «Εκκίνηση Σάρωσης» και να επανεκκινήσει την εφαρμογή του για να εφαρμοστούν οι αλλαγές.

Σε περίπτωση που δεν θέλει να κάνει αλλαγές πρέπει να πατήσει το πλήκτρο «Ακύρωση».



Σχήμα 3: Διεπαφή χρήσης για τις ρυθμίσεις της σάρωσης

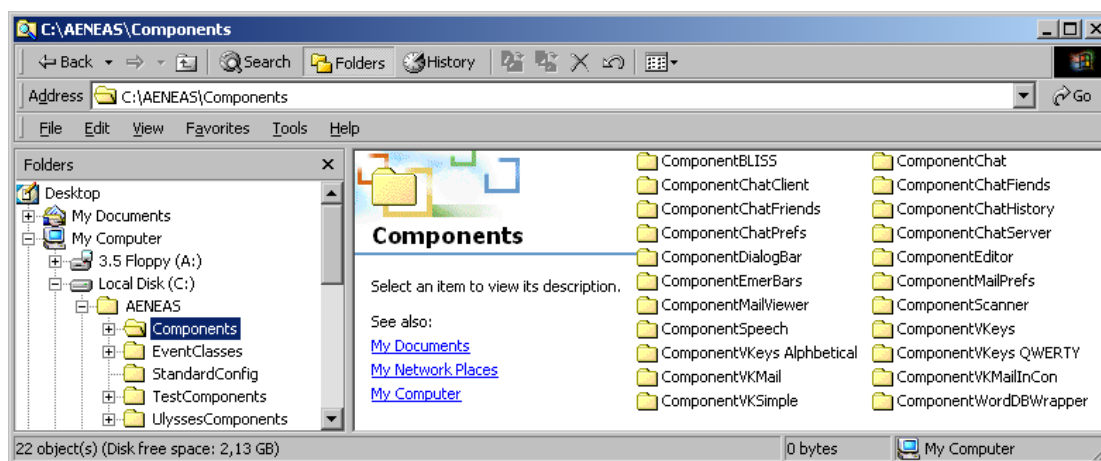
3. Για τους Πωλητές

Ένα τμήμα του πλαισίου ΟΔΥΣΣΕΑΣ απευθύνεται στους πωλητές βοηθημάτων διαπροσωπικής επικοινωνίας. Έγιναν επεκτάσεις στο τμήμα αυτό του πλαισίου όσον αφορά τη διαδικασία δημιουργίας και εγκατάστασης ολοκληρωμένων εφαρμογών.

3.1. Σύνθεση εφαρμογών

Η διαδικασία σύνθεσης εφαρμογών μετά τις αλλαγές που έγιναν στο εκτελέσιμο πρόγραμμα του πλαισίου ΟΔΥΣΣΕΑΣ (Communicator.exe) αλλά και στις κλάσεις των συστατικών έχει τροποποιηθεί. Ο πωλητής θα πρέπει να έχει στο σκληρό του δίσκο αρχειοθετημένα τα συστατικά των τρίτων κατασκευαστών, αλλά και τα απαραίτητα αρχεία για τη λειτουργία του πλαισίου ΟΔΥΣΣΕΑΣ σε ειδικούς καταλόγους, για παράδειγμα στο C:/Aeneas/Components (Σχήμα 4). Αφού εκτελέσει τα παρακάτω βήματα της διαδικασίας σύνθεσης και προετοιμασίας της εγκατάστασης, θα πρέπει να παραδώσει το CD-ROM που προκύπτει στον χρήστη, να κρατήσει ένα αντίγραφο με όλες τις ρυθμίσεις, τα απαραίτητα αρχεία και τη διαμόρφωση της εφαρμογής, ίδιο με αυτό του πελάτη και να απεγκαταστήσει την εφαρμογή από το σύστημά του. Η απεγκατάσταση από το μηχάνημα του πωλητή γίνεται σε δύο βήματα:

1. διαγράφοντας την εφαρμογή AENEAS από τα Component Services και
2. διαγράφοντας όλα τα περιεχόμενα του φακέλου C:/Program Files/Communicator.



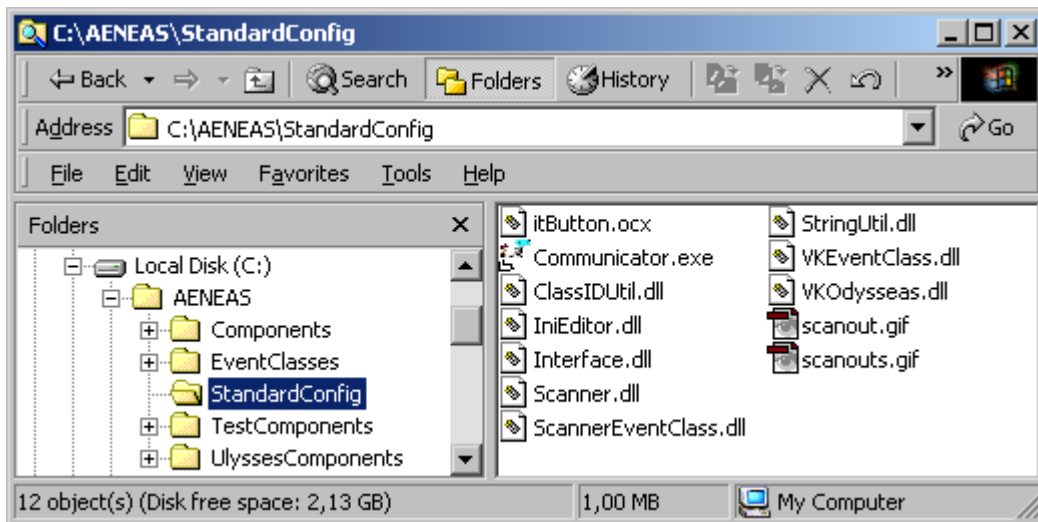
Σχήμα 4: Παράδειγμα αρχειοθέτησης των συστατικών του ΑΙΝΕΙΑ στον δίσκο του πωλητή

Τα βήματα που πρέπει να ακολουθήσει ο πωλητής για τη σύνθεση μιας νέας εφαρμογής είναι τα εξής:

1. Αρχικά το σύστημα του πωλητή θα πρέπει μην έχει καμία εγκατεστημένη εφαρμογή του πλαισίου ΟΔΥΣΣΕΑΣ, και ο κατάλογος C:/Program Files/Communicator να είναι κενός ή να μην υπάρχει. Αυτό σημαίνει

ότι κάθε φορά που ο πωλητής διαμορφώνει μία εφαρμογή για κάποιον πελάτη και την παραδίδει, θα πρέπει να κρατά ένα αντίγραφο ασφαλείας της εφαρμογής σε CD-ROM και να την αφαιρεί από το σύστημά του.

2. Το πρώτο βήμα είναι δημιουργηθεί ο κατάλογος C:/Program Files/Communicator και να αντιγραφούν εκεί τα απαραίτητα αρχεία για τη λειτουργία του πλαισίου ΟΔΥΣΣΕΑΣ. Αυτά είναι όσα φαίνονται στο Σχήμα 5. Συνίσταται να κρατούνται αυτά τα αρχεία σε ένα φάκελο (π.χ. τον C:/Aeneas/Components/StandardConfig) ώστε να είναι όλα μαζί διαθέσιμα για άμεση πρόσβαση και μαζική αντιγραφή στον κατάλογο της εφαρμογής. Στην περίπτωση που η εφαρμογή δεν πρόκειται να χρειάζεται την υποστήριξη σάρωσης, τα αρχεία Scanner.dll, VKEventClass.dll και δεν είναι απαραίτητο να συμπεριλαμβάνονται.
3. Στη συνέχεια θα πρέπει να εκτελεστεί το C:/Program Files/Communicator/Communicator.exe το οποίο θα ρωτήσει αν η εφαρμογή πρόκειται να υποστηρίξει σάρωση ή όχι. Ο πωλητής θα πρέπει να απαντήσει σε αυτό το μήνυμα για να συνεχιστεί η εγκατάσταση.
 - Αν η απάντηση είναι θετική για τη υποστήριξη σάρωσης τότε το εκτελέσιμο αρχείο θα εγκαταστήσει στο Component Services την εφαρμογή ΑΙΝΕΙΑΣ και όλα τα αρχεία και τα Event Classes που απαιτεί το πλαίσιο ΟΔΥΣΣΕΑΣ για εφαρμογές με υποστήριξη σάρωσης.
 - Αν η απάντηση είναι θετική για τη υποστήριξη σάρωσης τότε το εκτελέσιμο αρχείο θα εγκαταστήσει στο Component Services την εφαρμογή ΑΙΝΕΙΑΣ και όλα τα αρχεία και τα Event Classes που απαιτεί το πλαίσιο ΟΔΥΣΣΕΑΣ για εφαρμογές χωρίς υποστήριξη σάρωσης.

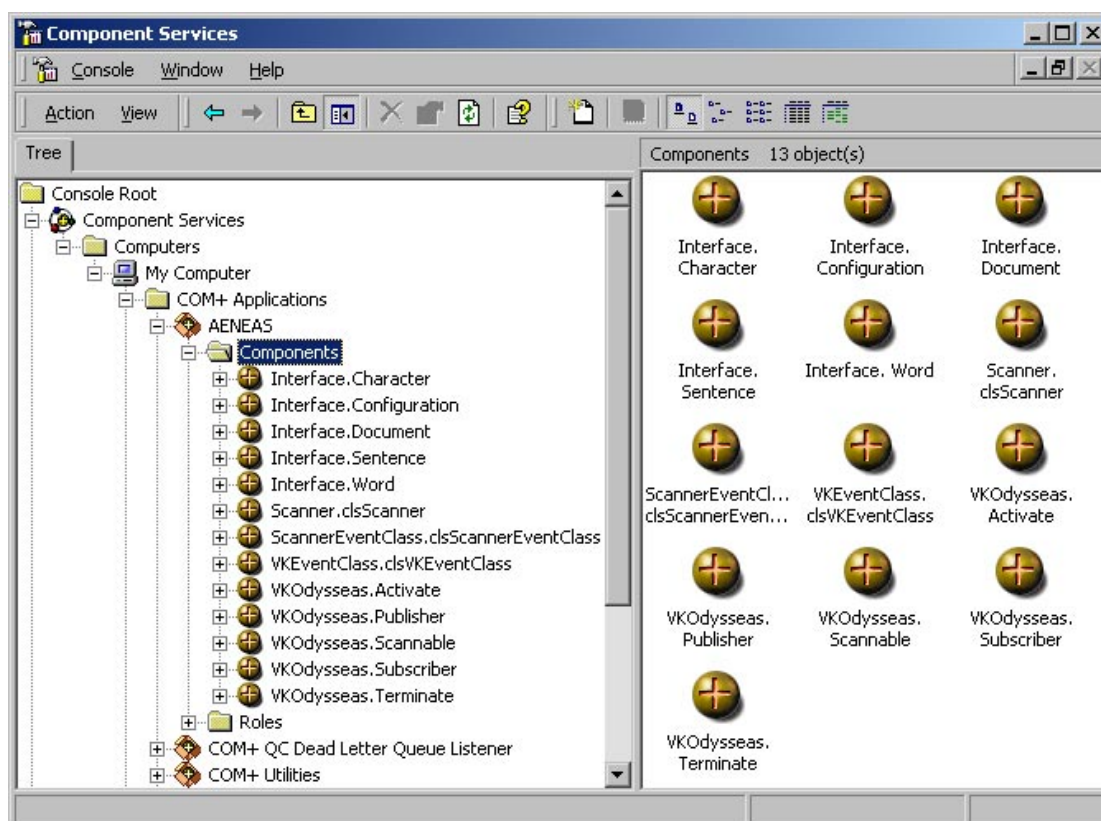


Σχήμα 5: Τα παραπάνω αρχεία θα πρέπει να είναι τα μόνα περιεχόμενα του C:/Program Files/Communicator κατά την εκκίνηση της σύνθεσης

4. Σε αυτό το σημείο μπορεί ο πωλητής να ελέγξει αν η ελάχιστη εγκατάσταση του πλαισίου ΟΔΥΣΣΕΑΣ λειτουργεί κανονικά εκτελώντας το Communicator.exe μία ακόμα φορά. Το αποτέλεσμα θα πρέπει να είναι η εμφάνιση μόνο του κεντρικού διακόπτη της εφαρμογής, με υποστήριξη ή όχι

σάρωσης, ανάλογα την επιλογή που προηγήθηκε. Θα πρέπει επίσης να λειτουργούν τα hotkeys του ΟΔΥΣΣΕΑ (F12 για τις ρυθμίσεις της σάρωσης και F11 για την μεταβολή της θέσης και του μεγέθους του παραθύρου).

5. Στη συνέχεια θα πρέπει να ανοίξει τα Component Services και να εγκαταστήσει όλα τα υπόλοιπα συστατικά που απαιτούν οι ανάγκες του χρήστη για την εφαρμογή του. Η οθόνη των Component Services σε αυτό το σημείο θα πρέπει να είναι όπως στο Σχήμα 6. Αν έχει επιλεγθεί λειτουργία χωρίς σάρωση, απλά θα λείπουν οι κλάσεις των Scanner.dll, VKEventClass.dll και ScannerEventClass.dll. Όλα τα απαραίτητα αρχεία για κάθε συστατικό θα αντιγράφονται πρώτα στον κατάλογο της εφαρμογής (C:/Program Files/Communicator) και στους υποκαταλόγους του, ανάλογα με τις οδηγίες του κάθε κατασκευαστή και στη συνέχεια θα εγκαθίστανται στα Component Services.
6. Στη συνέχεια, το ουσιαστικότερο βήμα της διαχείρισης του πωλητή είναι ο συγχρονισμός και ο καθορισμός της αλληλουχίας της επικοινωνίας μεταξύ των συστατικών με τη βοήθεια των συνδρομών και των φίλτρων τους στα Component Services. Η επέμβαση του πωλητή θα γίνεται στις κλάσεις Subscriber που χρειάζεται, με τη δημιουργία συνδρομών. Αυτές οι συνδρομές θα πρέπει να συγχρονίζονται μέσω φίλτρων με αντίστοιχες κλάσεις Publisher τις οποίες θα πρέπει να ακούν σύμφωνα με τη σύνθεση της εφαρμογής και με όσα έχουν περιγραφεί στα [5] και [6].



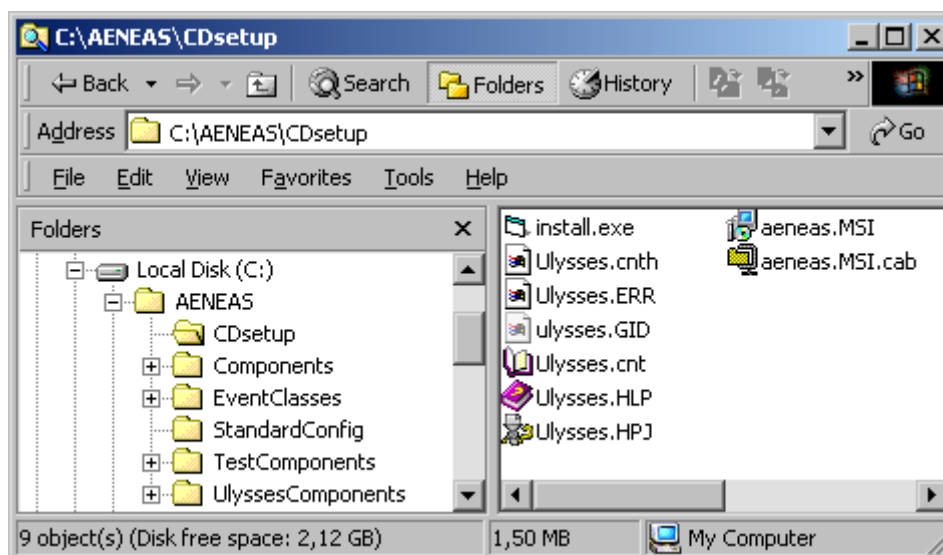
Σχήμα 6: Κατάσταση των Component Services μετά την εγκατάσταση του ΟΔΥΣΣΕΑ (με υποστήριξη σάρωσης)

7. Το τελικό βήμα είναι ο έλεγχος από τον πωλητή ότι η εφαρμογή λειτουργεί στον υπολογιστή όπως αναμένεται και η διεξαγωγή των απαραίτητων ρυθμίσεων στη διεπαφή χρήσης και τη λειτουργία των συστατικών, σύμφωνα με τις οδηγίες των κατασκευαστών αλλά και τις προδιαγραφές του ΟΔΥΣΣΕΑ.

3.2. Προετοιμασία CD-ROM εγκατάστασης εφαρμογών

Όταν όλα τα βήματα της προηγούμενης ενότητας ολοκληρωθούν με επιτυχία και η εφαρμογή είναι πλήρως λειτουργική, μπορεί ο πωλητής να προχωρήσει στη δημιουργία του CD-ROM εγκατάστασης της εφαρμογής στο μηχάνημα του χρήστη. Τα βήματα είναι τα εξής:

1. Θα πρέπει ο πωλητής να έχει έναν κατάλογο (π.χ. C:\AENEAS\CDsetup), ο οποίος θα περιέχει τα απαραίτητα αρχεία για το πρόγραμμα εγκατάστασης του βοηθήματος διαπροσωπικής επικοινωνίας στον υπολογιστή του τελικού χρήστη (Σχήμα 7). Το λογισμικό αυτό έχει αναπτυχθεί επίσης ως μέρος του πλαισίου ΟΔΥΣΣΕΑΣ.



Σχήμα 7: Περιεχόμενα του καταλόγου των απαραίτητων αρχείων για τη δημιουργία του CD-ROM Εγκατάστασης του βοηθήματος επικοινωνίας.

2. Από τα Component Services, θα πρέπει να γίνει export (με δεξί κλικ στην εφαρμογή AENEAS) ολόκληρη η εφαρμογή στον παραπάνω κατάλογο. Θα δημιουργηθούν τα δύο αρχεία aeneas.msi και aeneas.msi.cab.
3. Στη συνέχεια ο πωλητής αντιγράφει τα περιεχόμενα του παραπάνω καταλόγου στο root directory του CD-ROM. Επίσης αντιγράφει και ολόκληρο τον κατάλογο C:\Program Files\Communicator ως ξεχωριστό directory (D:\Communicator/) στο ίδιο CD-ROM.

Με την ολοκλήρωση αυτών των βημάτων τελειώνει η δημιουργία του CD-ROM εγκατάστασης του βοηθήματος διαπροσωπικής επικοινωνίας. Είναι σημαντικό να

σημειωθεί ότι η επιτυχία της εγκατάστασης στον υπολογιστή του τελικού χρήστη εξαρτάται άμεσα από τον σωστό σχεδιασμό των επιμέρους συστατικών. Οι προγραμματιστές θα πρέπει να έχουν συμπεριλάβει στην τεκμηρίωσή τους όλο το αναγκαίο λογισμικό (dlls) που χρησιμοποιείται από το συστατικό τους, καθώς και τη διαδικασία εγκατάστασής του στα Component Services. Αν ο πωλητής έχει περιλάβει στον κατάλογο της εφαρμογής και στα Component Services όλα τα αναγκαία αρχεία και βιβλιοθήκες, η τελική εγκατάσταση θα είναι επιτυχημένη. Επίσης θα πρέπει να δοθεί προσοχή από τους κατασκευαστές και τους πωλητές στις αναγκαίες δηλώσεις των διαφόρων dlls που χρησιμοποιούνται από τα συστατικά. Καλό θα είναι όποιο dll χρειάζεται δήλωση (registration) να εγκαθίσταται στα Component Services στο μηχάνημα του πωλητή, ώστε να δηλώνεται αυτόματα και στο μηχάνημα του τελικού χρήστη.

4. Για τους Προγραμματιστές

4.1. Αρχείο ρυθμίσεων (Settings.ini)

Το πλαίσιο «ΟΔΥΣΣΕΑΣ» [4], [5] εμπλουτίστηκε με ένα αρχείο ρυθμίσεων γενικής χρήσης με το όνομα Settings.ini. Το αρχείο αυτό χρησιμεύει για την αποθήκευση χρήσιμων δεδομένων για τα στοιχεία και τις ιδιότητες του χρήστη του βοηθήματος διαπροσωπικής επικοινωνίας, αλλά και δεδομένων απαραίτητων για τη λειτουργία των επιμέρους συστατικών που αποτελούν την ολοκληρωμένη εφαρμογή.

Η ανάγκη για την προσθήκη αυτού του αρχείου στο πλαίσιο «ΟΔΥΣΣΕΑΣ» προέκυψε από το γεγονός ότι σε ολοκληρωμένες εφαρμογές, οι οποίες αποτελούνταν από πολλά συστατικά υπήρχε η ανάγκη αποθήκευσης πληροφοριών σε κάποιο αρχείο για το κάθε συστατικό ξεχωριστά. Αυτό είχε ως αποτέλεσμα την παρουσία στο δίσκο του τελικού χρήστη και συγκεκριμένα στον κατάλογο της εφαρμογής C:\Program Files\Communicator, πολλών αρχείων ρυθμίσεων που αυξάνουν την πολυπλοκότητα και μειώνουν την απόδοση και την ευκολία ελέγχου και αποσφαλμάτωσης.

Το αρχείο ρυθμίσεων γενικής χρήσης χρησιμοποιείται για την αποθήκευση δεδομένων όπως:

- οι θέσεις των παραθύρων των διεπαφών χρήσης των επιμέρους συστατικών
- τα μεγέθη των παραθύρων των διεπαφών χρήσης των επιμέρους συστατικών
- η υποστήριξη ή μη τεχνικής σάρωσης
- το είδος της σάρωσης που υποστηρίζεται από την εφαρμογή
- η γλώσσα του χρήστη
- το λεξιλόγιο του χρήστη
- οι φιλικοί συνομιλητές του χρήστη
- το ονοματεπώνυμο του χρήστη
- το ψευδώνυμο του χρήστη (για τις εφαρμογές ηλεκτρονικής κουβέντας και ηλεκτρονικής αλληλογραφίας)
- η δικτυακή διεύθυνση του μηχανήματος του χρήστη και στοιχεία απαραίτητα για τη σύνδεσή του με τους εξυπηρετές ηλεκτρονικής κουβέντας και ηλεκτρονικής αλληλογραφίας
- το μεσοδιάστημα της σάρωσης
- το χρώμα υποβάθρου
- το χρώμα προσκηνίου (εστίαση σάρωσης)
- οι κλάσεις των διεπαφών χρήσης των συστατικών που υποστηρίζουν σάρωση

Τα δεδομένα στο αρχείο ρυθμίσεων αποθηκεύονται με τη μορφή κειμένου και την εξής διαμόρφωση:

```
[Όνομα Ενότητας #1]
Όνομα Πεδίου #1 = Τιμή Πεδίου #1
Όνομα Πεδίου #2 = Τιμή Πεδίου #2
Όνομα Πεδίου #3 = Τιμή Πεδίου #3
.
```

.
.
[Όνομα Ενότητας #2]
Όνομα Πεδίου #1 = Τιμή Πεδίου #1
Όνομα Πεδίου #2 = Τιμή Πεδίου #2
Όνομα Πεδίου #3 = Τιμή Πεδίου #3
.
.
.

Το Όνομα Ενότητας μπορεί να είναι οποιαδήποτε αλφαριθμητική σειρά που αντιπροσωπεύει την γενική ονομασία ενός συνόλου ομαδοποιημένων δεδομένων. Το Όνομα Πεδίου είναι το όνομα του συγκεκριμένου δεδομένου και η Τιμή Πεδίου είναι η τιμή του. Για παράδειγμα το συστατικό τεχνικών σάρωσης που είναι ενσωματωμένο στο πλαίσιο «ΟΔΥΣΣΕΑΣ» έχει τα εξής δεδομένα μέσα στο αρχείο ρυθμίσεων:

```
[Scanner]  
TimeInterval=1000  
FgColor=255  
BgColor=0  
DirectedScanning=1
```

Ομαδοποιούνται λοιπόν δεδομένα με το κριτήριο ότι όλα αποτελούν στοιχεία ρυθμίσεων του ίδιου συστατικού με το όνομα Scanner. Τα δεδομένα αυτά είναι με τη σειρά που εμφανίζονται το μεσοδιάστημα της σάρωσης, το χρώμα υποβάθρου, το χρώμα προσκηνίου και το είδος της σάρωσης (1 = κατευθυνόμενη σάρωση, 2 = αυτόματη σάρωση).

Προτείνεται η χρήση του αρχείου αυτού από όλα τα συστατικά που ενσωματώνονται στο βοήθημα επικοινωνίας, ώστε να υπάρχει ένα κεντρικό σημείο αναφοράς από όπου θα μπορούν να αντλούνται δεδομένα, αλλά και να αποφεύγεται η συσσώρευση μεγάλου αριθμού αρχείων ρυθμίσεων στον δίσκο του χρήστη. Συγκεκριμένα θα πρέπει να αναφέρεται στις προδιαγραφές του κάθε συστατικού ποιο όνομα ενότητας ή ποια ονόματα ενότητων χρησιμοποιούνται, καθώς και ποια ονόματα πεδίων και με ποιες προεπιλεγμένες τιμές (αν υπάρχουν). Συνίσταται η χρήση μιας ενότητας για την αποθήκευση όλων των δεδομένων κάθε συστατικού για να μην υπάρχουν διενέξεις στα ονόματα των ενότητων. Όσον αφορά τα ονόματα των μεταβλητών, αυτά αφήνονται ελεύθερα μια και δεν υπάρχει ο κίνδυνος της διένεξης λόγω ίδιου ονόματος μεταβλητών μια και αυτές θα ανήκουν σε διαφορετικές ενότητες.

Όνομα Ενότητας	Περιγραφή	Όνομα Πεδίου	Περιγραφή
General	Γενικές πληροφορίες για το χρήστη και την εφαρμογή του βοηθήματος επικοινωνίας	SupportsScanning	Πληροφορία για το αν η εφαρμογή υποστηρίζει τεχνικές σάρωσης ή όχι. Παίρνει την τιμή 1 αν υποστηρίζεται η σάρωση και την τιμή 0 αν όχι
		Name	Το όνομα του χρήστη
		Surname	Το επώνυμο του χρήστη
		Nickname	Το ψευδώνυμο του χρήστη
		Photo	Η διαδρομή στο δίσκο για το αρχείο εικόνας που περιέχει τη φωτογραφία του χρήστη
		UserLang	Η γλώσσα του χρήστη
		CountryLang	Η γλώσσα της χώρας του χρήστη.
Scanner	Δεδομένα ρυθμίσεων που αφορούν τις τεχνικές σάρωσης	TimeInterval	Το μεσοδιάστημα για τη σάρωση σε msec
		FgColor	Το χρώμα υποβάθρου σε τύπο δεδομένων long
		BgColor	Το χρώμα προσκηνίου σε τύπο δεδομένων long
		DirectedScanning	Τύπος σάρωσης (0 για αυτόματα, 1 για κατευθυνόμενη)
VKOdysseas	Δεδομένα ρυθμίσεων που αφορούν το παράθυρο του πλήκτρου τερματισμού της εφαρμογής	Top	Απόσταση του παραθύρου από την κορυφή της οθόνης
		Left	Απόσταση του παραθύρου από το αριστερό άκρο της οθόνης
		Height	Ύψος του παραθύρου
		Width	Πλάτος του παραθύρου
About	Δεδομένα ρυθμίσεων που αφορούν το παράθυρο του πλήκτρου εμφάνισης πληροφοριών για την εφαρμογή	Top	Απόσταση του παραθύρου από την κορυφή της οθόνης
		Left	Απόσταση του παραθύρου από το αριστερό άκρο της οθόνης
		Height	Ύψος του παραθύρου
		Width	Πλάτος του παραθύρου

Πίνακας 1: «Κοινά» δεδομένα του αρχείου ρυθμίσεων

Σημειώνεται ότι η ενότητα γενικής χρήσης [General], όπως και οι ενότητες που χρησιμοποιούνται από συστατικά ενσωματωμένα στο πλαίσιο «ΟΔΥΣΣΕΑΣ», και τα πεδία τους μπορούν να χρησιμοποιούνται ελεύθερα από οποιαδήποτε συστατικά χρειάζονται τις τιμές τους για την προσαρμογή της λειτουργίας τους. Αυτά τα «κοινής χρήσης» δεδομένα παρατίθενται στον παραπάνω Πίνακα 1.

Θα πρέπει να τονιστεί ότι δεν είναι πάντα σίγουρο ότι όλα αυτά τα δεδομένα θα είναι διαθέσιμα. Αυτό σημαίνει ότι οι προγραμματιστές που ίσως χρησιμοποιούν αυτά τα δεδομένα θα πρέπει πρώτα με τον κώδικά τους να ελέγχουν αν όντως αυτά υπάρχουν μέσα στο αρχείο Settings.ini.

4.2. Λογισμικό διαχείρισης αρχείου ρυθμίσεων (IniEditor.dll)

Στο πλαίσιο «ΟΔΥΣΣΕΑΣ» [4], [5] προστέθηκε σε μορφή μιας ξεχωριστής βιβλιοθήκης dll κώδικας για τη διαχείριση αρχείου ρυθμίσεων. Με τη χρήση της βιβλιοθήκης ο προγραμματιστής έχει τη δυνατότητα δημιουργίας ενός αρχείου ρυθμίσεων, πρόσθεσης και αφαίρεσης ενοτήτων και πεδίων, καθώς και ανάγνωσης των τιμών των πεδίων που επιθυμεί. Το όνομα του αρχείου βιβλιοθήκης είναι IniEditor.dll και περιέχει τις εξής ρουτίνες και ιδιότητες που είναι διαθέσιμες στους προγραμματιστές:

Property IniFileName () As String

Η ρύθμιση αυτής της ιδιότητας είναι πάντα το πρώτο βήμα μετά τη δήλωση και τη δημιουργία του αντικειμένου του IniEditor για να ξεκινήσει οποιαδήποτε διαχείριση του αρχείου ρυθμίσεων. Είναι η ιδιότητα του αντικειμένου που το πληροφορεί για τη διαδρομή του δίσκου για το αρχείο ρυθμίσεων. Πριν λοιπόν από οποιαδήποτε χρήση των ρουτινών διαχείρισης θα πρέπει να προηγείται ο κώδικας που φαίνεται στο παράδειγμα που ακολουθεί.

Παράδειγμα:

```
Dim iniObj As Object
Set iniObj = CreateObject ("IniEditor.clsIniEditor")
iniObj.IniFileName = App.Path & "\Testing.ini"
```

Κώδικας 1: Αρχικοποίηση για τη χρήση του IniEditor

όπου Testing.ini είναι το όνομα του αρχείου ρυθμίσεων (στην περίπτωση μας το Settings.ini).

Sub ProfileSaveItem(lpSectionName As String, lpKeyName As String, lpValue As String)

Χρησιμοποιείται για την εγγραφή ενός πεδίου στο αρχείο ρυθμίσεων ή για την τροποποίηση της τιμής ενός ήδη υπάρχοντος πεδίου. Απαιτεί ως ορίσματα το όνομα της Ενότητας (lpSectionName), το όνομα του Πεδίου (lpKeyName) και την Τιμή του πεδίου (lpValue). Αν δε βρεθεί το αρχείο, η ενότητα ή το πεδίο, η ρουτίνα τα δημιουργεί.

Παράδειγμα:

```
iniObj.ProfileSaveItem "Section#1", "Item#1", "Value#1"
```

Κώδικας 2: Χρήση της ρουτίνας ProfileSaveItem

Function ProfileGetItem(lpSectionName As String, lpKeyName As String, defaultValue As String) As String

Επιστρέφει την τιμή ενός πεδίου ως αλφαριθμητική σειρά. Απαιτεί ως ορίσματα το όνομα της Ενότητας (lpSectionName), το όνομα του Πεδίου (lpKeyName) και

την προκαθορισμένη Τιμή του πεδίου (defaultValue). Και τα τρία ορίσματα πρέπει να δίνονται ως αλφαριθμητικές σειρές. Αν δε βρεθεί το αρχείο, η ενότητα ή το πεδίο, η ρουτίνα επιστρέφει την προκαθορισμένη τιμή (defaultValue).

Παράδειγμα:

```
Dim itemValue As String
itemValue = iniObj.ProfileGetItem("Section#1", "Item#1", "Default Value")
```

Κώδικας 3: Χρήση της ρουτίνας ProfileGetItem

Sub ProfileDeleteItem(lpSectionName As String, lpKeyName As String)

Χρησιμοποιείται για τη διαγραφή ενός πεδίου. Το πεδίο και η τιμή του χάνονται για πάντα. Πρέπει να δοθεί ως όρισμα το όνομα της Ενότητας (lpSectionName) και το όνομα του Πεδίου (lpKeyName) σε μορφή αλφαριθμητικής σειράς. Αν κάποιος από τα δύο δεν υπάρχει, η ρουτίνα απλά δεν κάνει τίποτα. Αν δεν υπάρχει το αρχείο στο οποίο αναφέρεται ο κώδικας, η ρουτίνα το δημιουργεί (άδειο).

Παράδειγμα:

```
iniObj.ProfileDeleteItem "Section#1", "Item#1"
```

Κώδικας 4: Χρήση της ρουτίνας ProfileDeleteItem

Function ProfileGetSection(lpSectionName As String) As String

Επιστρέφει τα περιεχόμενα μιας ενότητας ως αλφαριθμητική σειρά. Απαιτεί ως όρισμα το όνομα της Ενότητας (lpSectionName) ως αλφαριθμητική σειρά. Αν δε βρεθεί η ενότητα ή το αρχείο, η ρουτίνα επιστρέφει μια κενή αλφαριθμητική σειρά. Τα πεδία της ενότητας διαχωρίζονται με χαρακτήρα αλλαγής γραμμής.

Παράδειγμα:

```
Dim sectionValue As String
sectionValue = iniObj.ProfileGetSection("Section#1")
```

Κώδικας 5: Χρήση της ρουτίνας ProfileGetSection

Sub ProfileDeleteSection(lpSectionName As String, inifile As String)

Χρησιμοποιείται για τη διαγραφή μιας Ενότητας. Η Ενότητα και τα Πεδία της χάνονται για πάντα. Πρέπει να δοθεί ως όρισμα το όνομα της Ενότητας (lpSectionName) σε μορφή αλφαριθμητικής σειράς. Αν η ενότητα δεν υπάρχει, η ρουτίνα απλά δεν κάνει τίποτα. Αν δεν υπάρχει το αρχείο στο οποίο αναφέρεται ο κώδικας, η ρουτίνα το δημιουργεί (άδειο).

Παράδειγμα:

```
iniObj.ProfileDeleteSection "Section#1"
```

Κώδικας 6: Χρήση της ρουτίνας ProfileDeleteSection

4.3. Λογισμικό ανάλυσης αλφαριθμητικών σειρών (StringUtil.dll)

Η συγκεκριμένη βιβλιοθήκη που προστέθηκε στο πλαίσιο «ΟΔΥΣΣΕΑΣ» [4], [5] συμπληρώνει τη λειτουργικότητα της προηγούμενης (IniEditor.dll) όσον αφορά την ανάλυση (parsing) αλφαριθμητικών σειρών. Συγκεκριμένα χρησιμοποιείται για να μας δίνει όλα τα πεδία μιας ενότητας μαζί με τις τιμές τους. Στην IniEditor.dll απαιτείται να ξέρει ο προγραμματιστής το όνομα του πεδίου και της ενότητας στην οποία ανήκει για να πάρει ή να θέσει την τιμή του. Η StringUtil.dll καλύπτει την περίπτωση όπου δεν είναι γνωστός ο αριθμός των πεδίων που περιέχει μια ενότητα, αλλά θέλουμε τις τιμές όλων αυτών των πεδίων. Το λογισμικό αυτό εκτελεί λειτουργίες ανάλυσης βασισμένο σε μερικές ιδιότητες και μία ρουτίνα. Αναλυτικότερα:

Property Let StringToParse(value As String)

Η ιδιότητα αυτή δίνεται ως είσοδος από τον προγραμματιστή. Είναι στην ουσία ολόκληρη η αλφαριθμητική σειρά που πρόκειται να αναλυθεί και μπορεί να αποτελείται από πολλαπλά πεδία και τιμές ή αντίστοιχα γραμμές κειμένου. Προτείνεται η συνεργασία με τη βιβλιοθήκη IniEditor.dll για να πάρει ο προγραμματιστής την αλφαριθμητική σειρά που θέλει να αναλύσει μέσα από το αρχείο ρυθμίσεων. Συνήθως αυτή η αλφαριθμητική σειρά είναι τα περιεχόμενα μιας ολόκληρης Ενότητας.

Property Let TokenChar(value As String)

Βασική ιδιότητα για το είδος της ανάλυσης που θέλουμε είναι ένας χαρακτήρας κλειδί (η σκυτάλη). Αυτός ο χαρακτήρας καθορίζει το διαχωρισμό των πεδίων μεταξύ τους σε πρώτη φάση και το διαχωρισμό των πεδίων από τις τιμές τους στη δεύτερη. Στην ουσία χρησιμοποιείται αυτός ο χαρακτήρας ως ένας δείκτης μέσα στην αρχική αλφαριθμητική σειρά για να λαμβάνει το πρόγραμμα «σημάδια» που απαιτούνται για την ανάλυση (parsing) και στην ουσία το διαχωρισμό των στοιχείων της. Σύμφωνα με τον ορισμό και τη μορφή του αρχείου ρυθμίσεων που δόθηκε προηγουμένως, χαρακτήρες κλειδιά για την ανάλυση είναι α χαρακτήρας αλλαγής γραμμής, ο οποίος δηλώνεται ως char(0) στην τιμή value της ιδιότητας TokenChar και διαχωρίζει μεταξύ τους τα πεδία της ενότητας και ο χαρακτήρας «=» που διαχωρίζει ένα πεδίο από την τιμή του.

Property Get HasMoreTokens() As Boolean

Η ιδιότητα αυτή βγαίνει ως έξοδος από τη βιβλιοθήκη και πληροφορεί το πρόγραμμα και το βρόχο της ανάλυσης (όπως αυτός φαίνεται στο παράδειγμα παρακάτω, για το αν η αρχική αλφαριθμητική σειρά έχει μέσα της ακόμα τον χαρακτήρα κλειδί (τη

σκυτάλη). Στην πραγματικότητα αυτό που κάνει το λογισμικό που περιγράφουμε είναι ψάχνει τους χαρακτήρες κλειδιά μέσα στην αρχική αλφαριθμητική σειρά και να τους χρησιμοποιεί ως σημεία διαχωρισμού των πεδίων και των τιμών. Η ιδιότητα αυτή καθορίζει πότε θα σταματήσει το πρόγραμμα να ψάχνει για πρόσθετα σημεία διαχωρισμού (όταν τελειώσουν οι χαρακτήρες κλειδιά, δηλαδή όταν γίνει αυτή η ιδιότητα 0).

Function GetToken() As String

Η μοναδική ρουτίνα τη βιβλιοθήκης επιστρέφει το τμήμα της αρχικής αλφαριθμητικής σειράς που βρίσκεται μεταξύ δύο χαρακτήρων κλειδιών. Πιο συγκεκριμένα, όπως φαίνεται και στο παρακάτω παράδειγμα, χρησιμοποιούνται δύο χαρακτήρες κλειδιά (το chr(0) και το =) και δύο αντικείμενα StringUtil.clsStringUtil (η μοναδική κλάση της βιβλιοθήκης). Το πρώτο αντικείμενο strLines αναλύει μια ολόκληρη ενότητα του αρχείου ρυθμίσεων και μέσω της GetToken επιστρέφει κάθε φορά μια γραμμή κειμένου από τη συγκεκριμένη ενότητα (δηλαδή το κείμενο πριν από το χαρακτήρα αλλαγής γραμμής). Το δεύτερο αντικείμενο strKeys αναλύει την έξοδο του strLines και επιστρέφει κάθε φορά το κείμενο που βρίσκεται πριν από το «=» κάθε γραμμής, δηλαδή τα ονόματα των πεδίων. Τελικά, ο παρακάτω κώδικας μας γεμίζει μια λίστα με τα ονόματα των πεδίων της ενότητας «ScannableKeyboards».

Παράδειγμα:

```
Dim strLines As Object
Dim strKeys As Object

Set strLines = CreateObject("StringUtil.clsStringUtil")
Set strKeys = CreateObject("StringUtil.clsStringUtil")

strSections.StringToParse = iniObj.ProfileGetSection("ScannableKeyboards")

While strLines.HasMoreTokens
    strLines.TokenChar = Chr(0)
    strKeys.TokenChar = "="
    strKeys.StringToParse = strLines.GetToken
    frmObj.List.AddItem strKeys.GetToken
Wend
```

Κώδικας 7: Παράδειγμα χρήσης της StringUtil

4.4. Λογισμικό ανάκτησης του Class ID (ClassIDUtil.dll)

Η βιβλιοθήκη αυτή ενσωματώθηκε στο πλαίσιο «ΟΔΥΣΣΕΑΣ» [4], [5] εξαιτίας της πολύ συχνής ανάγκης για όλα τα συστατικά να γνωρίζουν το Class ID τους. Όπως έχει ήδη αναφερθεί το Class ID είναι μια αλφαριθμητική σειρά που χαρακτηρίζει μοναδικά την κάθε κλάση που εντάσσεται στα Component Services στην εφαρμογή ΑΙΝΕΙΑΣ, αλλά και όλες τις κλάσεις γενικότερα. Αυτή η βιβλιοθήκη μας προσφέρει μία και μοναδική ρουτίνα η οποία δέχεται σαν είσοδο το όνομα της κλάσης της οποίας θέλουμε το Class ID και επιστρέφει το Class ID.

Function GetClassID(value As String) As String

Δίνοντας ως είσοδο στη μεταβλητή value το όνομα της κλάσης με τη μορφή, για παράδειγμα VKOdysseas.Publisher, η ρουτίνα επιστρέφει το Class ID της με τη μορφή για παράδειγμα {38727D9A-E5F2-47BE-8003-94F35CA01A7A}.

Παράδειγμα:

```
Dim PubID As String
Dim objCLSID As New ClassIDUtil.clsClassID
PubID = objCLSID.GetClassID("VKOdysseas.Publisher")
```

Κώδικας 8: Παράδειγμα χρήσης της ClassIDUtil

4.5. Η προγραμματιστική διεπαφή ελέγχου (Interface.Configuration)

Όπως έχει ήδη αναλυθεί υπάρχουν τέσσερις διεπαφές για την επικοινωνία των συστατικών που αποτελούν τις ολοκληρωμένες εφαρμογές που βασίζονται στο πλαίσιο «ΟΔΥΣΣΕΑΣ» [4], [5]. Οι διεπαφές αυτές (Character, Word, Sentence, Document) που βρίσκονται μέσα στη βιβλιοθήκη Interface.dll χρησιμοποιούνται για την κυκλοφορία και μεταφορά των δεδομένων του χρήστη και μπορεί κάποιος να τις διαχειριστεί μέσω των κλάσεων Έκδοσης και Συνδρομής που διαθέτουν τα διάφορα συστατικά και μέσα από την κονσόλα των Component Services.

Εκτός όμως τα δεδομένα του χρήστη, κατά την υλοποίηση βοηθημάτων διαπροσωπικής επικοινωνίας παρατηρήθηκε η ανάγκη για μεταφορά μηνυμάτων ελέγχου στα διάφορα συστατικά είτε από το κεντρικό πρόγραμμα (Communicator.exe) είτε από συστατικό σε συστατικό. Για παράδειγμα υπήρξαν περιπτώσεις στις οποίες μόνο μια συγκεκριμένη διεπαφή χρήσης ενός συστατικού θα έπρεπε να είναι εμφανής στην οθόνη του χρήστη, και η αλληλεπίδρασή του με αυτή τη διεπαφή χρήσης να καθορίζει την επόμενη κατάσταση λειτουργίας της ολοκληρωμένης εφαρμογής. Αυτό σημαίνει ότι όλες οι υπόλοιπες διεπαφές χρήσης των άλλων συστατικών θα έπρεπε να κρυφτούν από το χρήστη και να μείνει μόνο η κρίσιμη για τη συνέχιση της εφαρμογής διεπαφή χρήσης ώστε να δώσει ο χρήστης αναγκαστικά την απαιτούμενη είσοδο (input) στην εφαρμογή. Χρειάστηκε λοιπόν με κάποιο τρόπο να ενημερωθούν όλα τα συστατικά ότι πρέπει να κρύψουν τις διεπαφές χρήσης τους. Η λύση ήταν μια πρόσθετη προγραμματιστική διεπαφή για την κυκλοφορία και μεταφορά μηνυμάτων ελέγχου όπως για παράδειγμα το μήνυμα «Hide all» για την απόκρυψη των παραθύρων. Η προγραμματιστική διεπαφή ονομάστηκε Configuration και ενσωματώθηκε στη βιβλιοθήκη των βασικών προγραμματιστικών διεπαφών του πλαισίου «ΟΔΥΣΣΕΑΣ» Interface.dll.

Στη συνέχεια παρατίθεται ο απλός κώδικας του ορισμού αυτής της προγραμματιστικής διεπαφής, όπου φαίνεται ότι, όπως και στις υπόλοιπες υπάρχει δυνατότητα αποστολής αλφαριθμητικών σειρών που αποτελούν τα μηνύματα ελέγχου μέσω του ορίσματος Data, αλλά και της ταυτότητας του Εκδότη, του αποστολέα

δηλαδή των δεδομένων για λόγους συγχρονισμού και φιλτραρίσματος μέσω του ορίσματος PubID.

```
Public Sub Configure(ByVal Data As String, ByVal PubID As String)
End Sub
```

Κώδικας 9: Δήλωση της Configuration στο Interface.dll

Για να επιτευχθεί ο σκοπός της διεπαφής αυτής θα πρέπει βέβαια να την «ακούν» όλα τα συστατικά του βοηθήματος διαπροσωπικής επικοινωνίας και φυσικά να είναι γνωστό το πρωτόκολλο επικοινωνίας σε αυτή τη διεπαφή. Αυτό σημαίνει ότι θα πρέπει να καθοριστούν τα τυπικά μηνύματα ελέγχου που θα εκδίδονται στη διεπαφή, ώστε να έχουν τη δυνατότητα οι προγραμματιστές των συστατικών να εκμεταλλεύονται ή όχι την παραλαβή των μηνυμάτων αυτών.

Το κεντρικό πρόγραμμα εκκίνησης του βοηθήματος επικοινωνίας βοηθά στο να έχουν όλα τα συστατικά μια Συνδρομή στη διεπαφή μηνυμάτων ελέγχου. Αυτό επιτυγχάνεται με την αυτόματη καταχώριση κάθε κλάσης Scannable και Activate αντίστοιχα για λειτουργία με σάρωση και χωρίς σάρωση, αλλά και όλων των κλάσεων Hidden για λειτουργία χωρίς διεπαφή χρήσης στη λίστα των Συνδρομητών στη διεπαφή Interface.Configuration. Παρατίθεται και το τμήμα του κώδικα του Communicator.exe για τις Συνδρομές αυτές:

```
'..... oComs ... .. Components ... Communicator
Dim oComs As COMAdminCatalogCollection
'..... oCom ... .. Component
Dim oCom As COMAdminCatalogObject
'..... Class IDs ..... Components
Dim strConfiguration As String
'..... Objects ... .. scannable .....
Dim Component As Collection
'..... scannable .....
Set Component = New Collection
'..... ..
Dim comp As Object
'..... Objects ... .. hidden .....
Dim ComponentH As Collection
'..... hidden .....
Set ComponentH = New Collection

'.....

'..... scannable .....
For Each oCom In oComs
If Mid(Right(oCom.Name, 12), 1, 9) = "Scannable" Or Right(oCom.Name, 10) =
".Scannable" Then
Component.Add Item:=CreateObject(oCom.Name)
End If
Next

'..... transient subscriptions
For Each comp In Component
CreateTransientSubscription strConfiguration, comp
Next

'..... .. Hidden Components
For Each oCom In oComs
```

```

If Right(oCom.Name, 7) = ".Hidden" Then
    ComponentH.Add Item:=CreateObject(oCom.Name)
End If
Next

'..... subscriptions .. ... .. Components .. ... .. .....
For Each comp In ComponentH
    CreateTransientSubscription strConfiguration, comp
Next

'.....

'..... .. ..... CLSID .. ... .. ..... .. ..... CreateTransientSubscription
For Each oCom In oComs
If oCom.Name = "Interface.Configuration" Then
    strConfiguration = oCom.Value("CLSID")
    Exit For
End If
Next

'..... .. .. user interfaces
For Each oCom In oComs
If Right(oCom.Name, 9) = ".Activate" Or Right(oCom.Name, 7) = ".Hidden" Then
    Component.Add Item:=CreateObject(oCom.Name)
End If
Next

'..... subscriptions .. ... .. Components .. ... .. .....
For Each comp In Component
    CreateTransientSubscription strConfiguration, comp
Next

```

Κώδικας 10: Κώδικας δημιουργίας προσωρινών συνδρομών όλων των συστατικών στην Configuration

Η ρουτίνα CreateTransientSubscription που εμφανίζεται σε διάφορα σημεία στον παραπάνω κώδικα είναι επίσης μια νέα ρουτίνα ενσωματωμένη στο εκτελέσιμο της εφαρμογής, η οποία παίρνοντας ως είσοδο το Class ID της προγραμματιστικής διεπαφής όπου θα γίνει μια συνδρομή και το αντικείμενο της κλάσης που θα γίνει συνδρομητής φτιάχνει μια transient (προσωρινή) συνδρομή στα Component Services, η οποία διαρκεί για όσο χρόνο η εφαρμογή λειτουργεί.

Τέλος, αν και το πρωτόκολλο μηνυμάτων ελέγχου από τη φύση του θα πρέπει να εξελίσσεται και να ενημερώνεται διαρκώς με νέα μηνύματα ελέγχου που θα πρέπει να δημοσιεύονται, σύμφωνα με τις μεταβαλλόμενες ανάγκες των εφαρμογών διαπροσωπικής επικοινωνίας και των συστατικών τους, παραθέτουμε στον παρακάτω πίνακα τα πρώτα μηνύματα ελέγχου που χρησιμοποιούνται ήδη.

Μήνυμα	Περιγραφή
--------	-----------

End Communicator	Είναι το μήνυμα τερματισμού ολόκληρης της εφαρμογής. Όλα τα συστατικά θα πρέπει να προγραμματιστούν έτσι ώστε να τερματίζουν άμεσα τη λειτουργία τους, ελευθερώνοντας με το σωστό τρόπο όλους τους πόρους που έχουν δεσμεύσει, όταν λαμβάνουν αυτό το μήνυμα από τη διεπαφή των μηνυμάτων ρυθμίσεων. Το μήνυμα αποστέλλεται από το συστατικό VKOdysseas όταν πιέζεται το πλήκτρο τερματισμού.
Resize	Η αποστολή αυτού του μηνύματος αποστέλλεται στη διεπαφή Configuration από το εκτελέσιμο πρόγραμμα της εφαρμογής όταν ανιχνεύεται η πίεση του πλήκτρου F11 από τον πωλητή (και συναρμολογητή) των βοηθημάτων διαπροσωπικής επικοινωνίας. Όταν τα συστατικά που διαθέτουν διεπαφές χρήσης λαμβάνουν αυτό το μήνυμα θα πρέπει να μετατρέπουν την κανονική κατάσταση λειτουργία της ενεργής κάθε φορά διεπαφής χρήσης τους σε κατάσταση μεταβολής της θέσης και του μεγέθους του παραθύρου.
Invisible	Ο ρόλος του συγκεκριμένου μηνύματος είναι να ειδοποιεί όλα τα συστατικά ότι θα πρέπει να «κρύψουν» τα παράθυρα των διεπαφών χρήσης τους από την οθόνη του χρήστη. Οι προγραμματιστές θα πρέπει να υπακούν σε αυτό το μήνυμα για την ομαλή και εύχρηστη λειτουργία των βοηθημάτων διαπροσωπικής επικοινωνίας. Ένα συστατικό θα πρέπει λοιπόν να εξαφανίζει τη διεπαφή χρήσης του όταν λαμβάνει αυτό το μήνυμα. Σε βοηθήματα που υποστηρίζουν την ενσωματωμένη τεχνική σάρωσης του ΟΔΥΣΣΕΑ, μέρος αυτής της διαδικασίας είναι και η ενημέρωση του συστατικού ελέγχου της σάρωσης για το κρύψιμο του παραθύρου μέσω της μεθόδου VKRemoveMe της κλάσης γεγονόςτος (Event Class) VKEventClass.clsVKEventClass. Το μήνυμα μπορούν να στείλουν όλα τα συστατικά που απαιτούν την απόκρυψη όλων των άλλων για τη λειτουργία τους.
Visible	Την ακριβώς αντίθετη λειτουργικότητα από το προηγούμενο έχει αυτό το μήνυμα. Τα συστατικά που με την παραλαβή του προηγούμενου μηνύματος έκρυψαν τα παράθυρά τους θα πρέπει να τα επανεμφανίσουν με την παραλαβή αυτού του μηνύματος. Σε βοηθήματα που υποστηρίζουν την ενσωματωμένη τεχνική σάρωσης του ΟΔΥΣΣΕΑ, μετά την εμφάνιση του παραθύρου θα πρέπει να ενημερωθεί και το συστατικό ελέγχου της σάρωσης ότι επανήλθε η διεπαφή χρήσης μέσω της μεθόδου VKInit του Event Class VKEventClass.clsVKEventClass. Το μήνυμα μπορούν να στείλουν όλα τα συστατικά που αποστέλλουν και το προηγούμενο.
Labels On	Πολλά συστατικά των βοηθημάτων διαπροσωπικής επικοινωνίας αναφέρονται σε άτομα που επικοινωνούν μέσω μιας γλώσσας που βασίζεται σε εικόνες ή σύμβολα. Για παράδειγμα οι Πίνακες Επιλογής Συμβόλων είναι συστατικά των οποίων η διεπαφή χρήσης είναι απλά κάποια πλήκτρα με σύμβολα επάνω τους, τα οποία όταν πιέζονται θα πρέπει να

	εκφωνείται για παράδειγμα η αντίστοιχη έννοια. Το παρόν μήνυμα χρησιμεύει για την αξιοποίηση της δυνατότητας που μπορεί να έχουν τέτοια συστατικά να εμφανίζουν εκτός από τα σύμβολα, και την έννοια που αντιστοιχεί σε κάθε σύμβολο με τη μορφή ετικέτας. Το μήνυμα αποστέλλεται από το κεντρικό εκτελέσιμο πρόγραμμα του βοηθήματος (Communicator.exe) με την πίεση των πλήκτρων CTRL+L.
Labels Off	Ενώ με το προηγούμενο μήνυμα οι ετικέτες (λεζάντες) των συμβόλων εμφανίζονται, με αυτό εδώ θα πρέπει να απενεργοποιούνται. Η εμφάνιση ή όχι των ετικετών έχει κάποιες φορές ιδιαίτερη σημασία κατά την εκμάθηση των συμβολικών συστημάτων επικοινωνίας ή κατά τη μετάβαση από αυτά σε φυσική γλώσσα. Το μήνυμα αποστέλλεται από το κεντρικό εκτελέσιμο πρόγραμμα του βοηθήματος (Communicator.exe) με την πίεση των πλήκτρων CTRL+L.
Mute On	Το μήνυμα αυτό ειδοποιεί τα συστατικά που παράγουν ήχο και κυρίως συστατικά σύνθεσης ομιλίας ή εκφώνησης να σταματήσουν τη λειτουργία τους. Υπάρχουν περιπτώσεις και καταστάσεις λειτουργίας των βοηθημάτων επικοινωνίας, όπου ο χρήστης δεν επιθυμεί την παραγωγή ήχου. Δυνατότητα αποστολής αυτού του μηνύματος θα πρέπει να έχουν όλα τα συστατικά των οποίων η λειτουργία μπορεί να απαιτήσει τέτοιες καταστάσεις λειτουργίας.
Mute Off	Χρησιμοποιείται για την ειδοποίηση των συστατικών που έχουν «σιγήσει» να αποκαταστήσουν την κανονική λειτουργία τους με την παραγωγή ήχου. Τα συστατικά που αποστέλλουν το προηγούμενο μήνυμα θα μπορούν να στέλνουν και το παρόν, εφόσον δεν απαιτούν πια τη σιγή.

Πίνακας 2: Τυποποιημένα μηνύματα της διεπαφής Configuration

4.6. Συστατικό σάρωσης

Βασική χαρακτηριστική δυνατότητα του πλαισίου ΟΔΥΣΣΕΑΣ [4], [5] είναι η υποστήριξη τεχνικών σάρωσης [6], [7]. Η λειτουργία αυτή απευθύνεται κυρίως σε ΑΜΕΑ με προβλήματα στην κίνηση, τα οποία δεν μπορούν να χρησιμοποιήσουν τις συμβατικές συσκευές εισόδου/εξόδου όπως το πληκτρολόγιο και το ποντίκι. Αυτές οι συσκευές απαιτούν κάποιες κινητικές δεξιότητες, τις οποίες, ένα μεγάλο ποσοστό από τους χρήστες-στόχους του έργου ΑΙΝΕΙΑΣ δεν διαθέτουν. Μια μεγάλη κατηγορία χρηστών έχουν μόνο τη διεξοδική λύση της αλληλεπίδρασης με το υπολογιστικό σύστημα μέσω διακοπών, οι οποίοι πρέπει να συνδυάζονται με δυνατότητες ειδικής σάρωσης της οθόνης για να επιτευχθεί ο έλεγχος στα σημεία αλληλεπίδρασης που αυτή περιέχει. Είναι τόσο σημαντικό αυτό το χαρακτηριστικό, ώστε κρίθηκε σκόπιμη η ενσωμάτωσή του, ως εγγενούς ιδιότητας στο πλαίσιο ΟΔΥΣΣΕΑΣ για το σχεδιασμό και υλοποίηση συστατικών βοηθημάτων διαπροσωπικής επικοινωνίας.

Οι τεχνικές σάρωσης που υλοποιήθηκαν, έχουν σχεδιαστεί έτσι ώστε να καλύπτουν ανάγκες χρηστών που έχουν τη δυνατότητα χρήσης ενός, τριών ή πέντε διακοπών. Οι διακόπτες αυτοί μπορεί να είναι απλοί διακόπτες πίεσης οι οποίοι συνδέονται στον υπολογιστή με τη βοήθεια ειδικού συμπληρωματικού εξοπλισμού (Switch Box) και παρεμβάλλονται συνήθως στη λειτουργία του πληκτρολογίου, δίνοντας στον υπολογιστή επιλεγμένα σήματα πίεσης πλήκτρων (keystrokes). Τα σήματα αυτά ανιχνεύει το συστατικό σάρωσης του ΟΔΥΣΣΕΑ και τα χρησιμοποιεί για τον έλεγχο της σάρωσης της οθόνης του υπολογιστή όπως θα περιγραφεί παρακάτω.

Αυτόματη σάρωση

Η αυτόματη σάρωση είναι σχεδιασμένη ώστε να ικανοποιεί τις ανάγκες αλληλεπίδρασης με το βοήθημα διαπροσωπικής επικοινωνίας με τη βοήθεια ενός μόνο διακόπτη. Αυτό σημαίνει ότι ο τελικός χρήστης είναι ικανός για την αποστολή στον υπολογιστή μόνο ενός απλού σήματος πίεσης ενός διακόπτη και με αυτή μόνο τη δυνατότητα θα πρέπει να μπορεί να χειρίζεται το βοήθημά του. Στοιχεία αλληλεπίδρασης του βοηθήματος σε αυτήν την περίπτωση είναι απλά πλήκτρα πίεσης σε διάφορα παράθυρα επάνω στην οθόνη. Η αυτόματη σάρωση αναλαμβάνει να δίνει με αυτόματο τρόπο διαδοχικά και κυκλικά την εστίαση σε αυτά τα στοιχεία αλληλεπίδρασης. Εκτός από τα μεμονωμένα πλήκτρα, στοιχεία αλληλεπίδρασης μπορούν να θεωρηθούν και ομάδες πλήκτρων, αλλά και ολόκληρα παράθυρα συστατικών.

Κατ' αρχή πρέπει να πούμε ότι στον τελικό χρήστη, η όλη λειτουργία της σάρωσης θα πρέπει να φαίνεται ως ο διαδοχικός φωτισμός για ένα προκαθορισμένο χρονικό διάστημα, ενός-ενός των στοιχείων αλληλεπίδρασης. Η πίεση του εξωτερικού διακόπτη του χρήστη κατά το χρονικό διάστημα που κάποιο σημείο αλληλεπίδρασης είναι φωτισμένο, θα προκαλέσει την επιλογή αυτού του στοιχείου αλληλεπίδρασης. Αυτή η επιλογή σημαίνει ότι θα εκτελεστεί η λειτουργία που εκτελεί αυτό το στοιχείο αλληλεπίδρασης. Οι λειτουργίες ανά στοιχείο αλληλεπίδρασης περιγράφονται στον παρακάτω πίνακα.

Στοιχείο αλληλεπίδρασης	Λειτουργικότητα κατά την επιλογή
Σάρωση στο επίπεδο των παραθύρων	
Παράθυρο συστατικού	Με την επιλογή ενός φωτισμένου παραθύρου συστατικού, η σάρωση θα εισέλθει σε αυτό το παράθυρο, φωτίζοντας στη συνέχεια διαδοχικά τα πλήκτρα ή τις ομάδες πλήκτρων που αυτό περιέχει (πέρασμα σε επίπεδο ομάδων, αν υπάρχουν, ή σε επίπεδο πλήκτρων).
Σάρωση στο επίπεδο των ομάδων πλήκτρων	
Ομάδα πλήκτρων	Με την επιλογή μιας φωτισμένης ομάδας πλήκτρων, η σάρωση θα εισέλθει στην ομάδα, φωτίζοντας στη συνέχεια διαδοχικά όλα τα πλήκτρα της (πέρασμα σε επίπεδο πλήκτρων). Αν δεν πατηθεί ο διακόπτης αφού φωτιστούν όλες οι ομάδες, τότε θα φωτιστεί το παράθυρο δίνοντας τη δυνατότητα εξόδου από αυτό και αν και πάλι δεν υπάρξει πίεση διακόπτη θα ξεκινήσει από την αρχή η σάρωση των ομάδων.
Παράθυρο ενεργού συστατικού	Με την επιλογή του φωτισμένου παραθύρου του ενεργού συστατικού, η σάρωση θα εξέλθει από αυτό το παράθυρο, φωτίζοντας στη συνέχεια διαδοχικά τα παράθυρα όλων των συστατικών του βοηθήματος (πέρασμα σε επίπεδο παραθύρων).
Σάρωση στο επίπεδο των πλήκτρων	
Πλήκτρο	Με την επιλογή ενός φωτισμένου πλήκτρου, θα εκτελεστεί η λειτουργία που αντιστοιχεί σε αυτό το πλήκτρο (σαν να πιεζόταν με το ποντίκι). Μετά την πίεση και τη λειτουργία του πλήκτρου, φωτίζεται πάλι η ομάδα του πλήκτρου αν υπάρχει (και περνάμε πάλι σε επίπεδο ομάδων) ή το παράθυρο του συστατικού για να δοθεί η δυνατότητα εξόδου από το παράθυρο.
Παράθυρο ενεργού συστατικού	Με την επιλογή του φωτισμένου παραθύρου του ενεργού συστατικού, η σάρωση θα εξέλθει από αυτό το παράθυρο, φωτίζοντας στη συνέχεια διαδοχικά τα παράθυρα όλων των συστατικών του βοηθήματος (πέρασμα σε επίπεδο παραθύρων). Αν δεν πατηθεί ο διακόπτης κατά τη διάρκεια που είναι το παράθυρο ενεργό, η σάρωση ξεκινά πάλι σε επίπεδο ομάδων αν υπάρχουν ή σε επίπεδο πλήκτρων.

Πίνακας 3: Περιγραφή της λειτουργικότητας των στοιχείων αλληλεπίδρασης κατά την σάρωση

Όπως φαίνεται και από τον παραπάνω πίνακα υπάρχουν τρία επίπεδα της λειτουργίας της σάρωσης (παραθύρων, ομάδων, πλήκτρων), στα οποία το πάτημα του διακόπτη έχει και διαφορετικό αποτέλεσμα. Η όλη διαδικασία και ακολουθία της σάρωσης έχει περιγραφεί στις προδιαγραφές του συστατικού [6], αλλά εδώ έχει σημασία να φανούν τα αποτελέσματά της και να δοθούν οι οδηγίες υποστήριξης στους προγραμματιστές. Ο τρόπος εμφάνισης της σάρωσης όπως αναφέρθηκε είναι ο φωτισμός του ενεργού κάθε φορά στοιχείου αλληλεπίδρασης. Αυτό επιτυγχάνεται ορίζοντας ένα χρώμα

υποβάθρου και ένα χρώμα προσκηνίου (φωτισμού). Αυτός ο ορισμός γίνεται από τη διεπαφή χρήσης του συστατικού σάρωσης και τα δεδομένα (οι τιμές των δύο χρωμάτων) είναι διαθέσιμα στο αρχείο ρυθμίσεων του βοηθήματος `Settings.ini` όπως αναφέρθηκε και προηγουμένως. Ο κώδικας που πρέπει να ενσωματωθεί στα συστατικά ώστε να λειτουργούν ομαλά και σε συνεργασία με τον κεντρικό έλεγχο του συστατικού σάρωσης θα περιγραφεί παρακάτω.

Στο σχήμα που ακολουθεί απεικονίζεται μια απλή περίπτωση της εμφάνισης της οθόνης του χρήστη, με δύο παράθυρα με πλήκτρα, από τα οποία το ένα έχει ομαδοποιημένα τα πλήκτρα του σε ομάδες (γραμμές). Με τη βοήθεια αυτού του παραδείγματος θα περιγράψουμε ποια είναι η επιθυμητή και αναμενόμενη συμπεριφορά του συστήματος με τη βοήθεια του συστατικού σάρωσης, του εξωτερικού διακόπτη και των συστατικών που υποστηρίζουν αυτήν την κατάσταση λειτουργίας.

Στο πρώτο στιγμιότυπο (α) φαίνεται η σάρωση στο επίπεδο των παραθύρων, στο δεύτερο (β) στο επίπεδο των ομάδων και στο τρίτο (γ) στο επίπεδο των πλήκτρων. Στην πρώτη περίπτωση (α) διακρίνεται το κόκκινο πλαίσιο φωτισμού γύρω από ολόκληρο το μεγάλο παράθυρο, στη δεύτερη (β) γύρω από την τρίτη σειρά-ομάδα και στην τρίτη γύρω από το τρίτο πλήκτρο της ομάδας αυτής.

Η σάρωση ξεκινά αυτόματα όταν ξεκινά το βοήθημα και συνεχίζει την αυτόματη μετάβαση του φωτισμού στα κατάλληλα στοιχεία αλληλεπίδρασης μέχρι το τέλος της λειτουργίας της εφαρμογής.

Κατευθυνόμενη σάρωση

Το συστατικό σάρωσης είναι έτσι σχεδιασμένο ώστε να υποστηρίζει και τη λεγόμενη κατευθυνόμενη σάρωση. Κατά συνέπεια, τα συστατικά που υποστηρίζουν την αυτόματη σάρωση θα υποστηρίζουν και την κατευθυνόμενη. Η κατευθυνόμενη σάρωση δεν περιλαμβάνει αυτόματη μετακίνηση του φωτισμού των στοιχείων αλληλεπίδρασης και δεν περιλαμβάνει επίπεδο ομάδων. Περιορίζεται λοιπόν στα επίπεδα των παραθύρων και των πλήκτρων και η κίνηση σε αυτά γίνεται αποκλειστικά με τη παρέμβαση του χρήστη. Ακόμη και το επίπεδο των παραθύρων είναι εκφυλισμένο με την έννοια ότι ένα ειδικό πλήκτρο αναλαμβάνει την μετακίνηση μεταξύ των παραθύρων και αντί ολόκληρου του παραθύρου, φωτίζεται μόνο το πλήκτρο αυτό (πλήκτρο επόμενου παραθύρου). Ο χρήστης ελέγχει την κίνηση του φωτισμού με δύο (επάνω, δεξιά) ή με τέσσερις (επάνω, κάτω, δεξιά, αριστερά) διακόπτες και με έναν πρόσθετο διακόπτη εκτελεί την επιλογή του στοιχείου αλληλεπίδρασης. Έτσι λοιπόν, η αλληλεπίδραση γίνεται είτε με τρεις είτε με πέντε διακόπτες. Κύρια στοιχεία αλληλεπίδρασης είναι τα πλήκτρα των πινάκων επιλογής συμβόλων που έχουν περιγραφεί στο [6] και η κίνηση σε αυτά είναι κυκλική (μετά το τελευταίο κάθε σειράς ή γραμμής η σάρωση συνεχίζει στο πρώτο). Η δράση των πλήκτρων είναι στην ευχέρεια του προγραμματιστή: μπορεί να στέλνουν τις έννοιες που τους αντιστοιχούν σε επόμενα συστατικά, ή να τις εκφωνούν, ή να εκτελούν οποιαδήποτε άλλη λειτουργία. Μόνο το ειδικό πλήκτρο «επόμενου παραθύρου» έχει αυστηρά καθορισμένη λειτουργία ώστε να μεταθέτει τη σάρωση μέσω του κεντρικού ελέγχου στο επόμενο παράθυρο.

Όπως και στην αυτόματη σάρωση, το ενεργό κάθε φορά στοιχείο αλληλεπίδρασης είναι εμφανές από το χρώμα προσκηνίου που το περιβάλλει σε αντίθεση με το χρώμα παρασκηνίου που περιβάλλει όλα τα υπόλοιπα μη ενεργά στοιχεία αλληλεπίδρασης του παραθύρου. Στο σχήμα που ακολουθεί φαίνεται η ίδια οθόνη με αυτήν της αυτόματης σάρωσης, μόνο που τώρα η εφαρμογή λειτουργεί σε κατάσταση κατευθυνόμενης σάρωσης. Από το στιγμιότυπο α) με πάτημα του διακόπτη επιλογής η σάρωση περνά στο επόμενο παράθυρο όπως φαίνεται στο στιγμιότυπο β). Εκεί με επιλογή 2 φορές του διακόπτη «επάνω» και 3 φορές του διακόπτη «δεξιά» η σάρωση έχει φτάσει στο πλήκτρο που φαίνεται στο στιγμιότυπο γ).

Στην κατευθυνόμενη σάρωση λοιπόν δεν έχουμε αυτόματη μετάβαση του φωτισμού στα στοιχεία αλληλεπίδρασης, αλλά μόνο με ενέργεια του χρήστη και τον κατάλληλο χειρισμό των ειδικών διακοπών η σάρωση μπορεί να φτάσει στην επιθυμητή θέση και να φωτιστεί (να γίνει ενεργό) το στοιχείο αλληλεπίδρασης που θέλει ο χρήστης. Περισσότερες τεχνικές λεπτομέρειες και παραδείγματα κώδικα υποστήριξης αυτών των τεχνικών θα δοθούν στο τέλος του κεφαλαίου.

4.7. Οι κλάσεις των συστατικών (Activate, Scannable, Hidden)

Για να είναι τα συστατικά των βοηθημάτων διαπροσωπικής επικοινωνίας συμβατά με το πλαίσιο ΟΔΥΣΣΕΑΣ και μεταξύ τους, θα πρέπει να υλοποιούν τρεις κλάσεις και να δίνουν σε εξωτερικά προγράμματα τη δυνατότητα δημιουργίας αυτών και των αντικειμένων τους. Κάθε μία από αυτές τις κλάσεις έχει και διαφορετική λειτουργικότητα και υπάρχει για διαφορετικό σκοπό. Κάθε συστατικό είναι επιθυμητό να διαθέτει τις δύο πρώτες κλάσεις, πράγμα που σημαίνει ότι υποστηρίζει λειτουργία με ή χωρίς τεχνικές σάρωσης αλλά μπορεί να διαθέτει μόνο μία από τις δύο, που σημαίνει ότι θα λειτουργεί σε μία από τις δύο καταστάσεις.

```
'..... oComs ... .. Components ... Communicator
Dim oComs As COMAdminCatalogCollection
'..... oCom ... .. Component
Dim oCom As COMAdminCatalogObject
'..... Objects ... .. scannable .....
Dim Component As Collection
'..... scannable .....
Set Component = New Collection
'.....
Dim comp As Object
'..... Objects ... .. hidden .....
Dim ComponentH As Collection
'..... hidden .....
Set ComponentH = New Collection

'.....

'..... scannable .....
For Each oCom In oComs
If Mid(Right(oCom.Name, 12), 1, 9) = "Scannable" Or Right(oCom.Name, 10) =
".Scannable" Then
Component.Add Item:=CreateObject(oCom.Name)
End If
Next

'..... .. Hidden Components
For Each oCom In oComs
If Right(oCom.Name, 7) = ".Hidden" Then
ComponentH.Add Item:=CreateObject(oCom.Name)
End If
Next

'..... .. user interfaces
For Each oCom In oComs
If Right(oCom.Name, 9) = ".Activate" Or Right(oCom.Name, 7) = ".Hidden" Then
Component.Add Item:=CreateObject(oCom.Name)
End If
Next
```

Κώδικας 11: Κώδικας δημιουργίας στιγμιότυπων των αντικειμένων που ορίζουν οι κλάσεις Scannable, Activate και Hidden.

Η τρίτη κλάση απευθύνεται σε συστατικά ή αντικείμενα τα οποία δε διαθέτουν διεπαφή χρήσης, λειτουργούν στο παρασκήνιο και δεν έχει νόημα να ελέγχεται αν υποστηρίζουν ή όχι σάρωση.

Τον κεντρικό έλεγχο για το ποιές κλάσεις των συστατικών θα ενεργοποιούνται σε κάθε βοήθημα τον έχει το κεντρικό εκτελέσιμο πρόγραμμα (Communicator.exe) το οποίο δημιουργεί τα αντικείμενα με τον κώδικα που δίνεται παρακάτω. Φαίνεται ότι κατά τη λειτουργία με υποστήριξη σάρωσης «σηκώνονται» οι κλάσεις Scannable και Hidden των συστατικών, ενώ σε λειτουργία χωρίς υποστήριξη σάρωσης «σηκώνονται» οι κλάσεις Activate και Hidden.

Activate

Η κλάση Activate είναι η πρώτη που τυποποιήθηκε στην πορεία της ανάπτυξης του πλαισίου ΟΔΥΣΣΕΑΣ. Η λειτουργία και οι προδιαγραφές της όμως, έχουν αλλάξει από την πρώτη περιγραφή της [4], [5] και ο κύριος λόγος είναι η ενσωμάτωση του συστατικού σάρωσης στο πλαίσιο, που επέφερε πρόσθετες απαιτήσεις στα πρωτόκολλα επικοινωνίας μεταξύ των συστατικών και στις διαφορετικές κλάσεις που πρέπει αυτά να υλοποιούν.

Η κλάση αυτή χρησιμοποιείται για να δημιουργούνται τα αντικείμενα των συστατικών του βοηθήματος διαπροσωπικής επικοινωνίας τα οποία διαθέτουν μία διεπαφή χρήσης που δεν υποστηρίζει τεχνικές σάρωσης. Αυτό δε σημαίνει ότι και το συστατικό δεν θα υποστηρίζει τη σάρωση, μια και όπως αναφέρθηκε θα μπορεί να υλοποιεί και τη διεπαφή Scannable συγχρόνως, η οποία ενεργοποιεί τη διεπαφή χρήσης που υποστηρίζει σάρωση.

Προφανή περιεχόμενα της κλάσης είναι για παράδειγμα η ενεργοποίηση φορμών, πλαισίων διαλόγου και γενικά στοιχείων αλληλεπίδρασης και διεπαφής χρήσης τα οποία λειτουργούν μεν με βάση ενέργειες του χρήστη αλλά δεν απαιτούν ούτε υποστηρίζουν σάρωση. Η αλληλεπίδραση γίνεται με τους συμβατικούς τρόπους, δηλαδή με το ποντίκι και το πληκτρολόγιο.

Προτείνεται οι φόρμες που ενεργοποιούνται για να υλοποιήσουν τη διεπαφή χρήσης του κάθε συστατικού, στην περίπτωση της μη υποστήριξης να δημιουργούνται ως αντικείμενα στην κλάση Activate (τουλάχιστο η φόρμα που εμφανίζεται με την ενεργοποίηση του συστατικού). Επίσης, όπως έχει ήδη αναφερθεί, η συγκεκριμένη κλάση θα έχει ούτως ή άλλως μία Transient Subscription στην προγραμματιστική διεπαφή Interface.Configuration. Καλό θα είναι ο προγραμματιστής να εκμεταλλευτεί αυτή τη συνδρομή και να υλοποιήσει τη διεπαφή (Implements Interface.Configuration), ώστε να γίνεται η διαχείριση των μηνυμάτων ελέγχου σε αυτήν την κλάση. Στο παράδειγμα του κώδικα στο τέλος του κεφαλαίου θα δοθεί ένα πλήρες παράδειγμα ενός πίνακα επιλογής συμβόλων που θα περιέχει και όλες τις κλάσεις που περιγράφουμε εδώ.

Scannable

Η κλάση Scannable υλοποιεί την ενεργοποίηση των διεπαφών χρήσης που υποστηρίζουν σάρωση, τόσο αυτόματη, όσο και κατευθυνόμενη. Είναι επίσης, το τμήμα του κώδικα όπου λαμβάνει χώρα η υποστήριξη της σάρωσης, με δηλώσεις μεταβλητών, συναρτήσεων και ρουτινών που αφορούν αυτή τη λειτουργία. Στη

συνήθη περίπτωση, η αρχική φόρμα ενός συστατικού που υποστηρίζει σάρωση ενεργοποιείται από αυτήν την κλάση. Προτείνεται τα αντικείμενα της διεπαφής χρήσης (για παράδειγμα οι φόρμες) να δημιουργούνται ως αντικείμενα σε αυτήν την κλάση όπως θα φανεί και από το παράδειγμα του κώδικα στο τέλος του κεφαλαίου. Επίσης συνίσταται σε αυτήν την κλάση να υλοποιούνται και τα διάφορα γεγονότα (events) που μπορεί να καλούνται από τα στοιχεία της διεπαφής χρήσης.

Σημειώνεται ότι το κεντρικό εκτελέσιμο πρόγραμμα έχει τη δυνατότητα ενεργοποίησης κατά την εκκίνηση του βοηθήματος επικοινωνίας όλων των κλάσεων που καταλήγουν σε `.Scannable???`. Αυτό σημαίνει ότι οι τρεις τελευταίοι χαρακτήρες της κλάσης μπορεί να είναι οποιοδήποτε, δίνοντας τη δυνατότητα στον προγραμματιστή να περιλαμβάνει περισσότερες από μία κλάσεις `Scannable` στο ίδιο συστατικό, αριθμώντας τις στους τρεις τελευταίους χαρακτήρες (π.χ. `Scannable001`, `Scannable002`, `Scannable 003` κτλ.) ή να δώσει στις `Scannable` κλάσεις του οποιοδήποτε αναγνωριστικές καταλήξεις τον βολεύουν (π.χ. `ScannableNew`, `ScannableOld` κτλ.).

Τέλος, όπως και οι κλάσεις `Activate` έτσι και οι `Scannable` έχουν από την αρχή της δημιουργίας τους, κάθε φορά που εκτελείται η εφαρμογή μια συνδρομή (που δημιουργείται αυτόματα) στη διεπαφή `Interface.Configuration`. Και σε αυτήν την περίπτωση προτείνεται στους προγραμματιστές να χρησιμοποιείται αυτή η συνδρομή για την παραλαβή των μηνυμάτων ελέγχου της προγραμματιστικής διεπαφή και για τη διαχείρισή τους.

Hidden

Αυτή η κλάση χρησιμοποιείται στην περίπτωση των συστατικών τα οποία δεν έχουν φανερή διεπαφή χρήσης, τουλάχιστον κατά την εκκίνηση της εφαρμογής. Συστατικά, όπως για παράδειγμα ένας Συνθέτης Ομιλίας, λειτουργούν στο παρασκήνιο χωρίς εμφανείς φόρμες ή πλήκτρα ή άλλα στοιχεία αλληλεπίδρασης και αυτή η κλάση είναι που τα ενεργοποιεί και τα αρχικοποιεί. Φυσικά με την έλλειψη γραφικής διεπαφής χρήσης δεν υπάρχει νόημα να συζητούμε για υποστήριξη ή μη υποστήριξη σάρωσης και από αυτό το γεγονός δημιουργήθηκε η ανάγκη τυποποίησης αυτής της διεπαφής. Ο ΟΔΥΣΣΕΑΣ αλληλεπιδρά με τη `Scannable` για να επιτύχει τη σάρωση σε κατάσταση λειτουργίας με σάρωση και αγνοεί την `Activate`. Σε κατάσταση λειτουργίας χωρίς σάρωση ενεργοποιεί την `Activate` και αγνοεί τη `Scannable`. Υπάρχουν όμως αντικείμενα που και στις δύο καταστάσεις λειτουργίας θα πρέπει να ενεργοποιούνται και αυτά περιλαμβάνονται στην `Hidden` που ενεργοποιείται ούτως ή άλλως.

4.8. Παράδειγμα υλοποίησης και κώδικας αναφοράς

Ως παράδειγμα όσον έχουν αναφερθεί για το αντικειμενοστραφές πλαίσιο σχεδιασμού και υλοποίησης βοηθημάτων διαπροσωπικής επικοινωνίας ΟΔΥΣΣΕΑΣ, στην παρούσα τεχνική έκθεση αλλά και στις [4] και [5] θα δοθεί η διαδικασία σχεδιασμού όσον αφορά τη διεπαφή χρήσης, και η υλοποίηση όσον αφορά τον κώδικα υποστήριξης του πλαισίου το παράδειγμα που ακολουθεί. Πρόκειται για έναν απλό πίνακα επιλογής συμβόλων διάστασης 2 γραμμών επί τρεις στήλες. Τα στοιχεία επιλογής είναι απλά πλήκτρα των Windows και το βάρος δίνεται στις προδιαγραφές

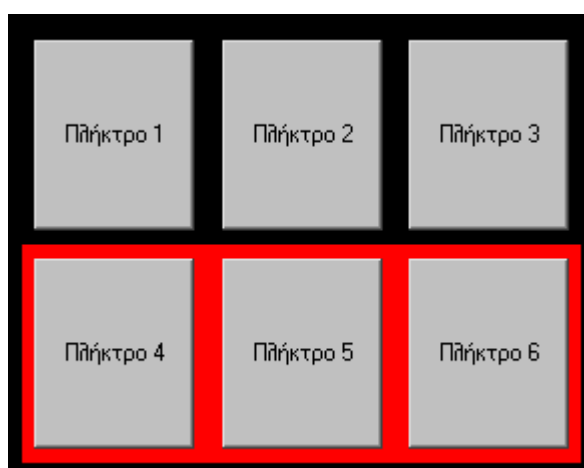
του ΟΔΥΣΣΕΑΣ και όχι στη λειτουργικότητα του συστατικού-παραδείγματος. Έτσι, όπως φαίνεται στη συνέχεια το συστατικό δεν εκτελεί καμία λειτουργία, η πίεση των πλήκτρων του δεν έχει κανένα αποτέλεσμα, παρά μόνο υποστηρίζει τεχνικές σάρωσης και τις προδιαγραφές του ΟΔΥΣΣΕΑ.

Στη συνέχεια θα παρουσιαστεί όλος ο κώδικας-υπόδειγμα και η διαδικασία δημιουργίας ενός συμβατού συστατικού. Πρέπει να σημειωθεί ότι ο κώδικας περιέχει σε σχόλια όλες τις απαραίτητες επεξηγήσεις και διευκρινήσεις.

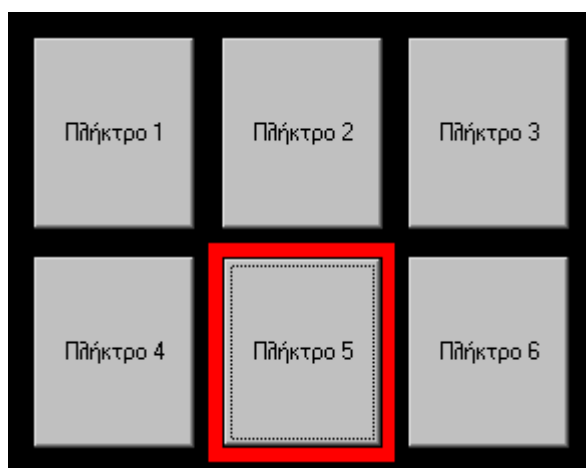
Ξεκινώντας από τη διεπαφή χρήσης του συστατικού-παραδείγματος, φαίνονται στα παρακάτω Σχήματα οι διάφορες καταστάσεις και φάσεις της λειτουργίας του στους διάφορους τύπους σάρωσης. Το παράδειγμα δεν έχει υλοποιηθεί για κατάσταση μη υποστήριξης σάρωσης μια και δεν υπάρχουν εκεί τόσες προδιαγραφές και κανόνες, ούτε ειδικός κώδικας υποστήριξης. Η ίδια διεπαφή χρήσης χωρίς τα ειδικά πλαίσια της σάρωσης και χωρίς τον αντίστοιχο κώδικα θα αποτελούσε κάλλιστα τη διεπαφή λειτουργίας χωρίς σάρωση (με όνομα φόρμας π.χ. SimpleForm). Το μόνο πρόσθετο που θα χρειαζόταν θα ήταν μια κλάση με το όνομα Activate που θα ενεργοποιούσε αυτή τη διεπαφή.



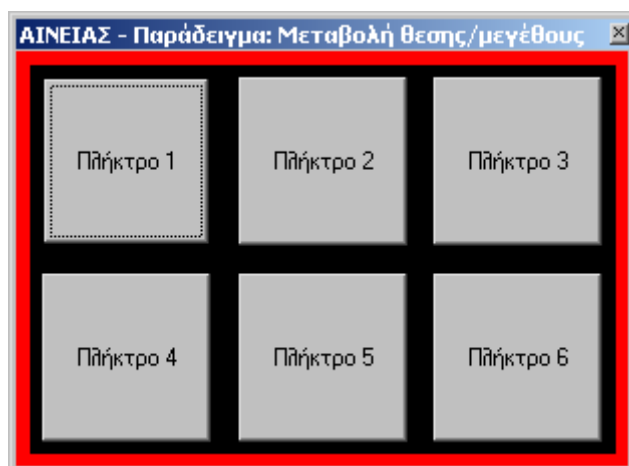
Σχήμα 10: Αυτόματη σάρωση στο επίπεδο παραθύρου



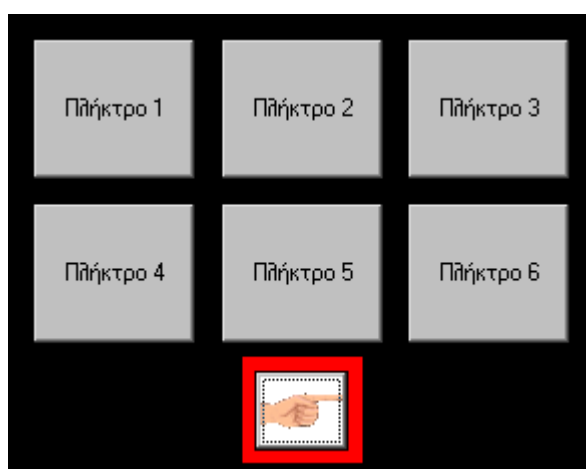
Σχήμα 11: Αυτόματη σάρωση στο επίπεδο των ομάδων



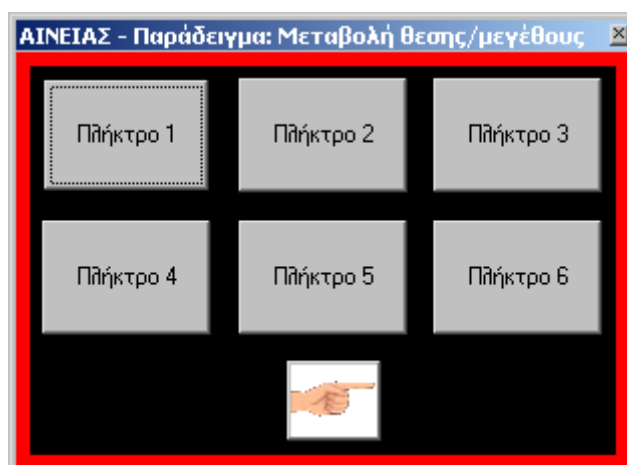
Σχήμα 12: Αυτόματη σάρωση στο επίπεδο των πλήκτρων



Σχήμα 13: Παράθυρο μεταβολής θέσης και μεγέθους για την αυτόματη σάρωση

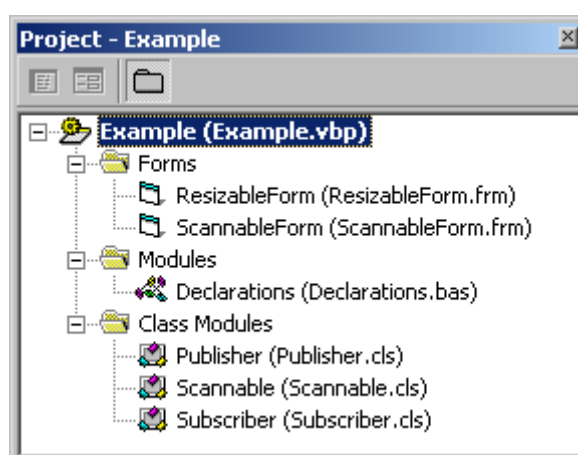


Σχήμα 14: Κατευθυνόμενη σάρωση στο επίπεδο των παραθύρων



Σχήμα 15: Παράθυρο μεταβολής θέσης και μεγέθους για την κατευθυνόμενη σάρωση

Στο επόμενο Σχήμα φαίνεται η δομή του project που υλοποιεί το παράδειγμα στη Microsoft Visual Basic 6.0. Φυσικά το πρόγραμμα θα είναι ένα Single Threaded ActiveX DLL σύμφωνα με τις προδιαγραφές του ΟΔΥΣΣΕΑ και θα μπορούσε να υλοποιηθεί σε οποιαδήποτε άλλη γλώσσα προγραμματισμού που υποστηρίζει αυτή την τεχνολογία



Σχήμα 16: Περιεχόμενα του dll παραδείγματος στη Visual Basic

Ακολουθεί ο κώδικας για το module Declarations το οποίο περιέχει τις απαραίτητες δηλώσεις μεταβλητών και τύπων για τη λειτουργία της σάρωσης. Βασικά στοιχεία που πρέπει να προσεχθούν είναι ο ορισμός τύπου δεδομένων ώστε τα πλήκτρα να αποκτήσουν πρόσθετες ιδιότητες που απαιτούνται για τη σάρωση, αλλά και ο ορισμός των keystrokes που κατευθύνουν τη σάρωση. Για τις συσκευές εισόδου (διακόπτες) που χρησιμοποιήθηκαν στο έργο ΑΙΝΕΙΑΣ, αυτά τα δεδομένα ήταν 1-κάτω (ASCII 49), 4-επάνω (ASCII 52), 2-αριστερά (ASCII 50), 3-δεξιά (ASCII 51), 0-επιλογή (ASCII 48). Οι αντιστοιχίες αυτών των πλήκτρων σύμφωνα με τον κώδικα ASCII φαίνονται στον κώδικα αυτόν.

**Κώδικας του module δήλωσης μεταβλητών απαραίτητων για τη σάρωση
(Declarations.bas)**

```
Option Explicit

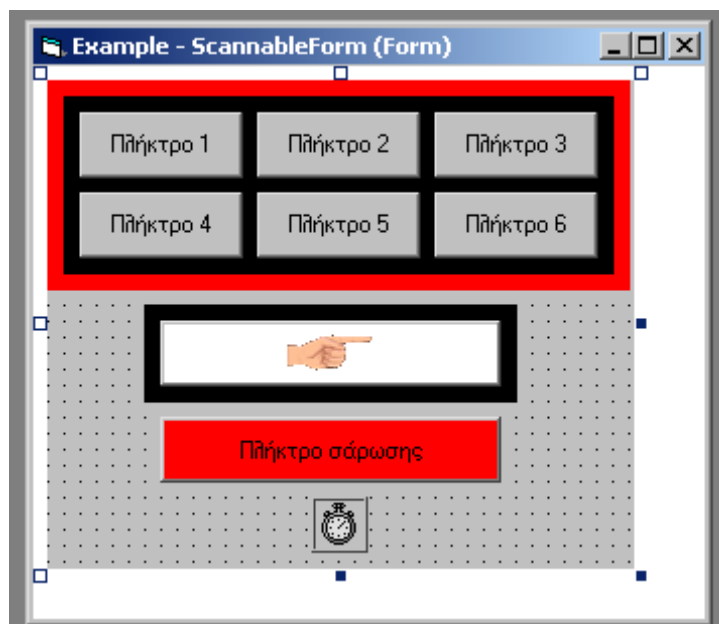
'.....
Public Type Coords
    x As Integer
    y As Integer
End Type

'.....
Public Type ButtonType
    strName As String
    cmdName As String
    nButtonID As Integer
    nGroupID As Integer
    coos As Coords
    special As Boolean
End Type

'..... keystrokes .....
Public Const BUTTONARRAYSIZE = 8
Public Const GOUP = 52
Public Const GODOWN = 49
Public Const GORIGHT = 51
Public Const GOLEFT = 50
```

Κώδικας 12: Κώδικας του module δήλωσης μεταβλητών απαραίτητων για τη σάρωση
(Declarations.bas)

Στη συνέχεια φαίνεται ο σχεδιασμός της διεπαφής χρήσης για την κανονική κατάσταση λειτουργίας.



Σχήμα 17: Φόρμα που υποστηρίζει σάρωση

Σημειώνεται η ύπαρξη των πλαισίων στα οποία στην ουσία εφαρμόζεται η σάρωση. Ξεκινώντας από το πιο πίσω επίπεδο και φτάνοντας εμπρός, τα πλαίσια είναι τα εξής: Bigframe (για τη σάρωση στο επίπεδο παραθύρων), SmallFrame (πάντα στο χρώμα υποβάθρου για να μη φαίνεται το χρώμα προσκηνίου του BigFrame σε κενούς χώρους), Frame? (ένα πλαίσιο για κάθε γύρω από κάθε πλήκτρο για το επίπεδο ομάδων και πλήκτρων). Επίσης φαίνονται στο προηγούμενο Σχήμα στοιχεία απαραίτητα για τη λειτουργία της σάρωσης όπως το πλήκτρο μετάβασης στο επόμενο πληκτρολόγιο με την εικόνα του χεριού hand.jpg και το άσπρο χρώμα (απαιτεί πλαίσιο για σάρωση) και το ειδικό, αόρατο κατά τη λειτουργία πλήκτρο σάρωσης. Σύμφωνα με τις δηλώσεις που θα ακολουθήσουν στην κλάση Scannable, η φόρμα αποτελείται από δύο ομάδες (συμπίπτουν με τις γραμμές) των τριών πλήκτρων.

Στη συνέχεια δίνεται ο κώδικας λειτουργίας της φόρμας αυτής. Θα πρέπει να δοθεί προσοχή στα σχόλια που περιέχει. Ο κώδικας αυτός είναι ο απόλυτα απαραίτητος για την υποστήριξη της σάρωσης και του πλαισίου ΟΔΥΣΣΕΑΣ και όπως είναι φανερό δεν περιέχει καμία λειτουργικότητα για τα πλήκτρα και για το συστατικό γενικότερα. Ο κώδικας για τις λειτουργίες μπορεί να προστεθεί σύμφωνα με τους κανόνες της Visual Basic σε οποιοδήποτε σημείο.

Τα γεγονότα (event) που δηλώνονται εδώ για τη φόρμα υλοποιούνται στην κλάση Scannable όπως θα φανεί στη συνέχεια. Δεδομένα όπως αυτά για τη θέση και το μέγεθος του παραθύρου, το είδος της σάρωσης και τα χρώματα υποβάθρου και προσκηνίου διαβάζονται από το αρχείο Settings.ini και ανατίθενται σε μεταβλητές.

```

..... (ScannableForm.frm)

Option Explicit
'..... events .....
Public DirectedScanning As Integer
Public Event Pressed()
Public Event Navigate(KeyCode As Integer)
Public Event GoNextVK()
Public Event StartScan()

Private Sub Button1_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = 48 Then
        '.....
        RaiseEvent Pressed
    End If
    '.....
    If DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If

End Sub

Private Sub Button2_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = 48 Then
        '.....
        RaiseEvent Pressed
    End If
    '.....
    If DirectedScanning = 1 Then

```



```
        RaiseEvent Navigate(KeyCode)
    End If
End Sub

Private Sub Button3_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = 48 Then
        '.....
        RaiseEvent Pressed
    End If
    '.....
    If DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If

End Sub

Private Sub Button4_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = 48 Then
        '.....
        RaiseEvent Pressed
    End If
    '.....
    If DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If

End Sub

Private Sub Button5_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = 48 Then
        '.....
        RaiseEvent Pressed
    End If
    '.....
    If DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If

End Sub

Private Sub Button6_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = 48 Then
        '.....
        RaiseEvent Pressed
    End If
    '.....
    If DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If

End Sub

Private Sub ButtonNext_KeyDown(KeyCode As Integer, Shift As Integer)

    '.....
    If DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If

End Sub
```

```

Private Sub ButtonNext_KeyUp(KeyCode As Integer, Shift As Integer)
    '..... (.....)
    If KeyCode = 48 Then
        RaiseEvent GoNextVK
    End If
End Sub

Private Sub Command_KeyDown(KeyCode As Integer, Shift As Integer)
    '.....
    If KeyCode = 48 And DirectedScanning = 0 Then
        RaiseEvent Pressed
    ElseIf DirectedScanning = 1 Then
        RaiseEvent Navigate(KeyCode)
    End If
End Sub

Private Sub Form_Load()
    Dim x
    '.....
    x = Arrange
    '..... timer .....
    TimerScanner.Enabled = True
End Sub

Private Sub TimerScanner_Timer()
    '.....
    RaiseEvent StartScan
    TimerScanner.Enabled = False
End Sub

'.....
Public Function Arrange()
    Dim iniObj As Object
    '..... loop
    Dim i As Integer
    '.....
    Dim t, l, h, w As Integer
    Dim v1, v2 As Integer
    '.....
    Set iniObj = CreateObject("IniEditor.clsIniEditor")
    iniObj.IniFileName = "C:\Program Files\Communicator\Settings.ini"
    DirectedScanning = CInt(iniObj.ProfileGetItem("Scanner",
"DirectedScanning", 0))
    '.....
    t = CInt(iniObj.ProfileGetItem("Example", "Top", 0))
    If t = 0 Then
        t = 1000
        iniObj.ProfileSaveItem "Example", "Top", Str(t)
    End If
    l = CInt(iniObj.ProfileGetItem("Example", "Left", 0))

```

```

    If l = 0 Then
        l = 1000
        iniObj.ProfileSaveItem "Example", "Left", Str(l)
    End If
h = CInt(iniObj.ProfileGetItem("Example", "Height", 0))
    If h = 0 Then
        h = 2000
        iniObj.ProfileSaveItem "Example", "Height", Str(h)
    End If
w = CInt(iniObj.ProfileGetItem("Example", "Width", 0))
    If w = 0 Then
        w = 2000
        iniObj.ProfileSaveItem "Example", "Width", Str(w)
    End If

'.....
With Me
    .Top = t
    .Left = l
    .Height = h
    .Width = w
End With

'.....
With BigFrame
    .Top = 0
    .Left = 0
    .Height = h
    .Width = w
End With

With SmallFrame
    .Top = 100
    .Left = 100
    .Height = h - 200
    .Width = w - 200
End With

If DirectedScanning = 0 Then
    '.....
    FrameNext.Visible = False
    ButtonNext.Visible = False

    v1 = (h - 200) / 2
    v2 = (w - 200) / 3
Else
    v1 = (h - 1000) / 2
    v2 = (w - 200) / 3
End If

'.....
With Frame1
    .Height = v1
    .Width = v2
    .Top = 100
    .Left = 100
End With
With Frame2
    .Height = v1
    .Width = v2
    .Top = 100
    .Left = Frame1.Left + Frame1.Width
End With
With Frame3
    .Height = v1
    .Width = v2

```

```
.Top = 100
.Left = Frame2.Left + Frame2.Width
End With
With Frame4
.Height = v1
.Width = v2
.Top = Frame1.Top + Frame1.Height
.Left = 100
End With
With Frame5
.Height = v1
.Width = v2
.Top = Frame4.Top
.Left = Frame4.Left + Frame4.Width
End With
With Frame6
.Height = v1
.Width = v2
.Top = Frame4.Top
.Left = Frame5.Left + Frame5.Width
End With
With FrameNext
.Height = 800
.Width = 900
.Top = Frame5.Top + Frame5.Height
.Left = (w - .Width) / 2
End With

'.....
With Button1
.Height = Frame1.Height - 200
.Width = Frame1.Width - 200
.Top = Frame1.Top + 100
.Left = Frame1.Left + 100
End With
With Button2
.Height = Frame2.Height - 200
.Width = Frame2.Width - 200
.Top = Frame2.Top + 100
.Left = Frame2.Left + 100
End With
With Button3
.Height = Frame3.Height - 200
.Width = Frame3.Width - 200
.Top = Frame3.Top + 100
.Left = Frame3.Left + 100
End With
With Button4
.Height = Frame4.Height - 200
.Width = Frame4.Width - 200
.Top = Frame4.Top + 100
.Left = Frame4.Left + 100
End With
With Button5
.Height = Frame5.Height - 200
.Width = Frame5.Width - 200
.Top = Frame5.Top + 100
.Left = Frame5.Left + 100
End With
With Button6
.Height = Frame6.Height - 200
.Width = Frame6.Width - 200
.Top = Frame6.Top + 100
.Left = Frame6.Left + 100
End With
With ButtonNext
```

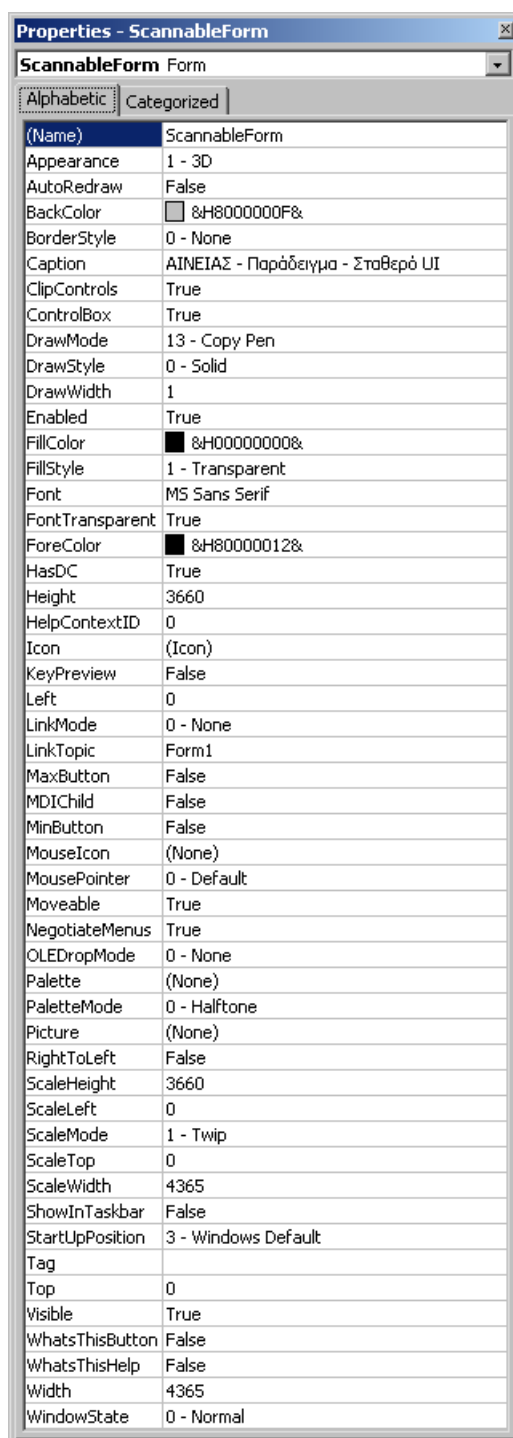
```
.Height = FrameNext.Height - 200
.Width = FrameNext.Width - 200
.Top = FrameNext.Top + 100
.Left = FrameNext.Left + 100
End With
With Command
.Height = 100
.Width = 100
.Top = h + 1000
.Left = w + 1000
End With

'.....
iniObj.ProfileSaveItem "Example", "Top", Str(t)
iniObj.ProfileSaveItem "Example", "Left", Str(l)
iniObj.ProfileSaveItem "Example", "Height", Str(h)
iniObj.ProfileSaveItem "Example", "Width", Str(w)
Set iniObj = Nothing

End Function
```

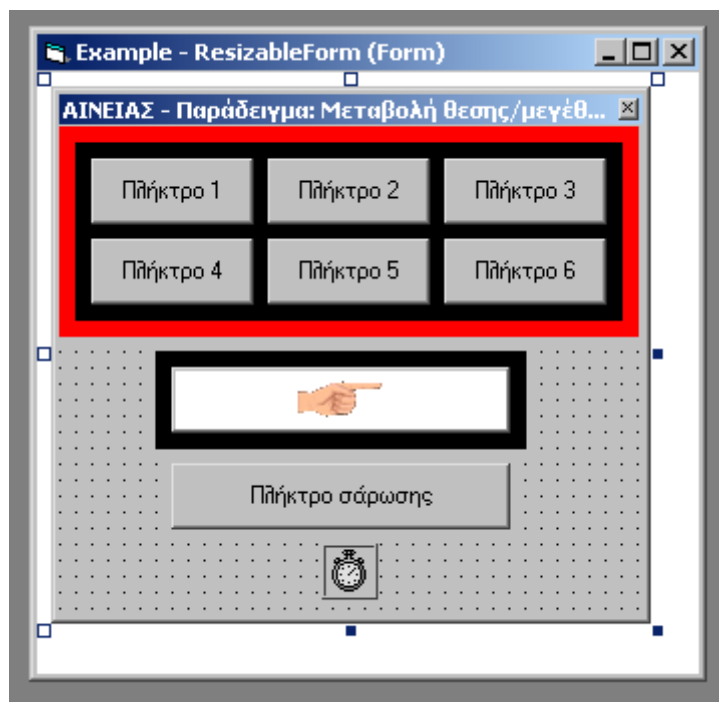
Κώδικας 13: Κώδικας φόρμας που υποστηρίζει σάρωση (ScannableForm.frm)

Στη συνέχεια απεικονίζονται οι ιδιότητες της φόρμας. Ένα στοιχείο που χρειάζεται προσοχή είναι ότι η ιδιότητα `BorderStyle` που έχει την τιμή `None`. Αυτό έχει ως αποτέλεσμα να φαίνεται το παράθυρο της διεπαφής χρήσης χωρίς κανένα πλαίσιο, χωρίς πλήκτρα ελέγχου (ελαχιστοποίηση, μεγιστοποίηση, κλείσιμο) και χωρίς μπάρα τίτλου όπως απαιτεί το πλαίσιο ΟΔΥΣΣΕΑΣ για όλες τις διεπαφές χρήσης.



Σχήμα 18: Ιδιότητες της φόρμας που υποστηρίζει σάρωση

Στη συνέχεια δίνεται ο σχεδιασμός, ο κώδικας και οι ιδιότητες για τη φόρμα με τη βοήθεια της οποίας γίνεται η μεταβολή μεγέθους και θέσης του παραθύρου. Στην ουσία είναι η ίδια με την προηγούμενη με μόνες διαφορές φυσικά στο όνομα, την ιδιότητα `BorderStyle` που εδώ είναι `Sizeable ToolWindow` και στις σταθερές για την προσαρμογή των μεγεθών και θέσεων πλαισίων και πλήκτρων λόγω της ύπαρξης εδώ πλαισίου ελέγχου και κατά συνέπεια του χώρου που αντιστοιχεί στα στοιχεία αλληλεπίδρασης.



Σχήμα 19: Φόρμα για τη μεταβολή μεγέθους και θέσης παραθύρου

```

..... (.....) .....
..... (ResizableForm.frm)

Option Explicit
'..... events .....
Public DirectedScanning As Integer

Private Sub Form_Load()

Dim iniObj As Object
Dim t, l, h, w As Integer

'.....
Set iniObj = CreateObject("IniEditor.clsIniEditor")
iniObj.IniFileName = "C:\Program Files\Communicator\Settings.ini"
DirectedScanning = CInt(iniObj.ProfileGetItem("Scanner",
"DirectedScanning", 0))
'.....
t = CInt(iniObj.ProfileGetItem("Example", "Top", 0))
If t = 0 Then
    t = 1000
    iniObj.ProfileSaveItem "Example", "Top", Str(t)
End If
l = CInt(iniObj.ProfileGetItem("Example", "Left", 0))
If l = 0 Then
    l = 1000
    iniObj.ProfileSaveItem "Example", "Left", Str(l)
End If
h = CInt(iniObj.ProfileGetItem("Example", "Height", 0))
If h = 0 Then
    h = 2000
    iniObj.ProfileSaveItem "Example", "Height", Str(h)
End If
w = CInt(iniObj.ProfileGetItem("Example", "Width", 0))
If w = 0 Then

```

```

        w = 2000
        iniObj.ProfileSaveItem "Example", "Width", Str(w)
    End If
Set iniObj = Nothing

'.....
With Me
    .Top = t
    .Left = l
    .Height = h
    .Width = w
End With

End Sub

Private Sub Form_Resize()

Dim iniObj As Object

'..... loop
Dim i As Integer

'.....
Dim t, l, h, w As Integer
Dim v1, v2 As Integer

If Me.Height < 2000 Then Me.Height = 2000
If Me.Width < 2000 Then Me.Width = 2000

'.....
With Me
    t = .Top
    l = .Left
    h = .Height
    w = .Width
End With

'.....
With BigFrame
    .Top = 0
    .Left = 0
    .Height = h - 350
    .Width = w - 160
End With

With SmallFrame
    .Top = 100
    .Left = 100
    .Height = h - 200 - 350
    .Width = w - 200 - 160
End With

If DirectedScanning = 0 Then
    '.....
    FrameNext.Visible = False
    ButtonNext.Visible = False

    v1 = (h - 200 - 350) / 2
    v2 = (w - 200 - 160) / 3
Else
    v1 = (h - 1000 - 350) / 2
    v2 = (w - 200 - 160) / 3
End If

'.....
With Frame1

```



```

        .Height = v1
        .Width = v2
        .Top = 100
        .Left = 100
    End With
    With Frame2
        .Height = v1
        .Width = v2
        .Top = 100
        .Left = Frame1.Left + Frame1.Width
    End With
    With Frame3
        .Height = v1
        .Width = v2
        .Top = 100
        .Left = Frame2.Left + Frame2.Width
    End With
    With Frame4
        .Height = v1
        .Width = v2
        .Top = Frame1.Top + Frame1.Height
        .Left = 100
    End With
    With Frame5
        .Height = v1
        .Width = v2
        .Top = Frame4.Top
        .Left = Frame4.Left + Frame4.Width
    End With
    With Frame6
        .Height = v1
        .Width = v2
        .Top = Frame4.Top
        .Left = Frame5.Left + Frame5.Width
    End With
    With FrameNext
        .Height = 800
        .Width = 900
        .Top = Frame5.Top + Frame5.Height
        .Left = (w - .Width) / 2
    End With

    ' .....
    With Button1
        .Height = Frame1.Height - 200
        .Width = Frame1.Width - 200
        .Top = Frame1.Top + 100
        .Left = Frame1.Left + 100
    End With
    With Button2
        .Height = Frame2.Height - 200
        .Width = Frame2.Width - 200
        .Top = Frame2.Top + 100
        .Left = Frame2.Left + 100
    End With
    With Button3
        .Height = Frame3.Height - 200
        .Width = Frame3.Width - 200
        .Top = Frame3.Top + 100
        .Left = Frame3.Left + 100
    End With
    With Button4
        .Height = Frame4.Height - 200
        .Width = Frame4.Width - 200
        .Top = Frame4.Top + 100
        .Left = Frame4.Left + 100
    End With

```

```

End With
With Button5
    .Height = Frame5.Height - 200
    .Width = Frame5.Width - 200
    .Top = Frame5.Top + 100
    .Left = Frame5.Left + 100
End With
With Button6
    .Height = Frame6.Height - 200
    .Width = Frame6.Width - 200
    .Top = Frame6.Top + 100
    .Left = Frame6.Left + 100
End With
With ButtonNext
    .Height = FrameNext.Height - 200
    .Width = FrameNext.Width - 200
    .Top = FrameNext.Top + 100
    .Left = FrameNext.Left + 100
End With
With Command
    .Height = 100
    .Width = 100
    .Top = h + 1000
    .Left = w + 1000
End With

End Sub

Private Sub Form_Unload(Cancel As Integer)

Dim iniObj As Object
Dim t, l, h, w As Integer
t = Me.Top
l = Me.Left
h = Me.Height
w = Me.Width

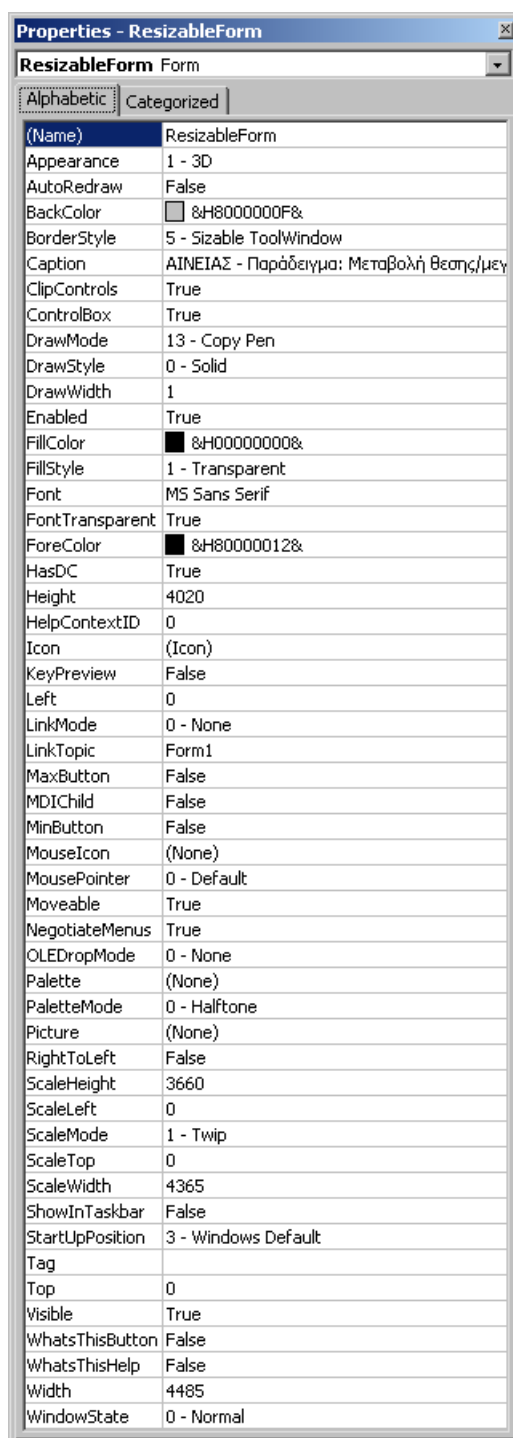
Set iniObj = CreateObject("IniEditor.clsIniEditor")
iniObj.IniFileName = "C:\Program Files\Communicator\Settings.ini"
'.....
iniObj.ProfileSaveItem "Example", "Top", Str(t)
iniObj.ProfileSaveItem "Example", "Left", Str(l)
iniObj.ProfileSaveItem "Example", "Height", Str(h)
iniObj.ProfileSaveItem "Example", "Width", Str(w)
Set iniObj = Nothing

'..... • run mode .....
ScannableForm.Arrange
ScannableForm.show

End Sub

```

Κώδικας 14: Κώδικας της τροποποίησής (σε θέση και μέγεθος) κατάσταση της φόρμας που υποστηρίζει σάρωση (ResizableForm.frm)



Σχήμα 20: Ιδιότητες της τροποποιήσιμης (σε θέση και μέγεθος) κατάσταση της φόρμας που υποστηρίζει σάρωση

Στη συνέχεια δίνεται ο κώδικας για τη βασική κλάση υποστήριξης της σάρωσης. Αυτή είναι και η κλάση που ενεργοποιείται από το εκτελέσιμο πρόγραμμα του βοηθήματος διαπροσωπικής επικοινωνίας και με τη σειρά της ενεργοποιεί τη φόρμα κανονικής λειτουργίας. Στο σημείο αυτό θα πρέπει να αναφερθεί ότι στο συγκεκριμένο πρόγραμμα, όπως και σε όλα τα συστατικά του πλαισίου ΟΔΥΣΣΕΑΣ θα πρέπει να προστίθενται στα References οι βιβλιοθήκες των κλάσεων γεγονότων

(event classes) της σάρωσης και της βασικής διεπαφής δεδομένων του ΟΔΥΣΣΕΑ για να λειτουργεί κανονικά ο κώδικας. Οι δύο αυτές βιβλιοθήκες είναι η ScannerEventClass.dll και η Interface.dll.

Κώδικας βασικής κλάσης για την υποστήριξη της σάρωσης (Scannable.cls)

```
Option Explicit

'.....
Implements Interface.Configuration
'.....
Implements ScannerEventClass.clsScannerEventClass
'.....
Dim WithEvents frmObj As ScannableForm
'.....
Private m_BgColor As Long
Private m_FgColor As Long
'.....
Dim buttonArray(8) As ButtonType
'.....
Dim curCoords As Coords
Dim maxX As Integer
Dim maxY As Integer
'.. Class ID ..
Dim MyScanCLSID As String

Private Sub Class_Initialize()

'..... .. Class ID ..
Dim objCLSID As Object
Set objCLSID = CreateObject("ClassIDUtil.clsClassID")
MyScanCLSID = objCLSID.GetClassID("Example.Scannable")

'.....
'.. ..
buttonArray(0).strName = "Frame1"
'.. ..
buttonArray(0).cmdName = "Button1"
'..... ID ..
buttonArray(0).nButtonID = 1
'.....
buttonArray(0).coos.x = 1
'.....
buttonArray(0).coos.Y = 1
'.. ..
buttonArray(0).nGroupID = 1
'.....
buttonArray(0).special = False

buttonArray(1).strName = "Frame2"
buttonArray(1).cmdName = "Button2"
buttonArray(1).nButtonID = 2
buttonArray(1).coos.x = 2
buttonArray(1).coos.Y = 1
buttonArray(1).nGroupID = 1
buttonArray(1).special = False

buttonArray(2).strName = "Frame3"
buttonArray(2).cmdName = "Button3"
buttonArray(2).nButtonID = 3
buttonArray(2).coos.x = 3
buttonArray(2).coos.Y = 1
buttonArray(2).nGroupID = 1
buttonArray(2).special = False
```

```

buttonArray(3).strName = "Frame4"
buttonArray(3).cmdName = "Button4"
buttonArray(3).nButtonID = 4
buttonArray(3).coos.x = 1
buttonArray(3).coos.Y = 2
buttonArray(3).nGroupID = 2
buttonArray(3).special = False

buttonArray(4).strName = "Frame5"
buttonArray(4).cmdName = "Button5"
buttonArray(4).nButtonID = 5
buttonArray(4).coos.x = 2
buttonArray(4).coos.Y = 2
buttonArray(4).nGroupID = 2
buttonArray(4).special = False

buttonArray(5).strName = "Frame6"
buttonArray(5).cmdName = "Button6"
buttonArray(5).nButtonID = 6
buttonArray(5).coos.x = 3
buttonArray(5).coos.Y = 2
buttonArray(5).nGroupID = 2
buttonArray(5).special = False

buttonArray(6).strName = "FrameNext"
buttonArray(6).cmdName = "ButtonNext"
buttonArray(6).nButtonID = 7
buttonArray(6).coos.x = 1
buttonArray(6).coos.Y = 3
'..... "....."
buttonArray(6).nGroupID = 3
'.....
buttonArray(6).special = True

'.....
buttonArray(7).strName = "FrameNext"
buttonArray(7).cmdName = "ButtonNext"
buttonArray(7).nButtonID = 8
buttonArray(7).coos.x = 2
buttonArray(7).coos.Y = 3
buttonArray(7).nGroupID = 3
buttonArray(7).special = True

buttonArray(8).strName = "FrameNext"
buttonArray(8).cmdName = "ButtonNext"
buttonArray(8).nButtonID = 9
buttonArray(8).coos.x = 3
buttonArray(8).coos.Y = 3
buttonArray(8).nGroupID = 3
buttonArray(8).special = True

'.....
curCoords.x = 1
curCoords.Y = 1
'..... (.....
maxX = 3
maxY = 3

'.....
Set frmObj = ScannableForm
frmObj.show vbModeless

End Sub

```

```

Private Sub Class_Terminate()
    '.....
    Unload frmObj
    Set frmObj = Nothing
End Sub

'.....
Public Sub frmObj_Navigate(KeyCode As Integer)

    Select Case (KeyCode)
    Case GOUP ' i
        If curCoords.Y = 1 Then
            curCoords.Y = maxY
        Else
            curCoords.Y = curCoords.Y - 1
        End If
    Case GODOWN ' m
        If curCoords.Y = maxY Then
            curCoords.Y = 1
        Else
            curCoords.Y = curCoords.Y + 1
        End If
    Case GORIGHT ' k
        If curCoords.x = maxX Then
            curCoords.x = 1
        Else
            curCoords.x = curCoords.x + 1
        End If
    Case GOLEFT ' j
        If curCoords.x = 1 Then
            curCoords.x = maxX
        Else
            curCoords.x = curCoords.x - 1
        End If
    End Select
    If frmObj.DirectedScanning = 0 Then
        frmObj_Pressed
    Else
        PaintButton
    End If
End Sub

'.....
Private Sub PaintButton()

    Dim i As Integer
    For i = 0 To BUTTONARRAYSIZE
        If buttonArray(i).coos.x = curCoords.x Then
            If buttonArray(i).coos.Y = curCoords.Y Then Exit For
        End If
    Next

    ResetAllButtons (False)
    Dim a As Object
    For Each a In frmObj.Controls
        If TypeOf a Is Frame Then
            If a.Name = buttonArray(i).strName Then
                a.ZOrder
                a.BackColor = m_FgColor
                Exit For
            End If
        End If
    Next
    For Each a In frmObj.Controls

```

```

        If TypeOf a Is CommandButton Then
            If a.Name = buttonArray(i).cmdName Then
                a.ZOrder
                a.SetFocus
            Exit For
        End If
    End If
Next

End Sub

'.....
Public Sub frmObj_Pressed()

    Dim vkEventClassObj As Object
    Set vkEventClassObj = CreateObject("VKEventClass.clsVKEventClass")
    vkEventClassObj.VKPressed MyScanCLSID
    Set vkEventClassObj = Nothing

End Sub

'.....
Public Sub frmObj_GoNextVK()

    Dim vkEventClassObj As Object
    Set vkEventClassObj = CreateObject("VKEventClass.clsVKEventClass")
    ResetAllButtons (False)
    vkEventClassObj.VKGoNext MyScanCLSID
    Set vkEventClassObj = Nothing

End Sub

'..... (.....)
.....)
Private Sub clsScannerEventClass_Timer(ByVal ID As Integer, ByVal clsID As
String, ByVal nType As Integer)

    If clsID = MyScanCLSID Then
        Dim a As Object
        Dim i As Integer
        frmObj.Command.SetFocus
        ResetAllButtons False
        Select Case nType
        Case 0
            For i = 0 To BUTTONARRAYSIZE
                If buttonArray(i).nButtonID = ID Then
                    For Each a In frmObj.Controls
                        If TypeOf a Is Frame Then
                            If a.Name = buttonArray(i).strName Then
                                a.ZOrder
                                a.BackColor = m_FgColor
                                Exit For
                            End If
                        End If
                    Next
                    For Each a In frmObj.Controls
                        If TypeOf a Is CommandButton Then
                            If a.Name = buttonArray(i).cmdName Then
                                a.ZOrder
                                a.SetFocus
                                Exit For
                            End If
                        End If
                    Next
                End If
            Next
        End If
    End If
Next

```

```

    Case 1
      For i = 0 To BUTTONARRAYSIZE
        If buttonArray(i).nGroupID = ID Then
          For Each a In frmObj.Controls
            If TypeOf a Is Frame Then
              If a.Name = buttonArray(i).strName Then
                a.ZOrder
                a.BackColor = m_FgColor
                Exit For
              End If
            End If
          Next
          For Each a In frmObj.Controls
            If TypeOf a Is CommandButton Then
              If a.Name = buttonArray(i).cmdName Then
                a.ZOrder
                Exit For
              End If
            End If
          Next
        End If
      Next
    Case 2
      ResetAllButtons True
    End Select
Else
  ResetAllButtons False
End If

End Sub

'.....
Private Sub clsScannerEventClass_SetColor(ByVal BgColor As Long, ByVal FgColor
As Long, ByVal DirectedScanning As Integer)

  m_BgColor = BgColor
  m_FgColor = FgColor
  frmObj.DirectedScanning = DirectedScanning

  If (frmObj.DirectedScanning = 0) Then
    ShowSpecialButtons (False)
  Else
    ShowSpecialButtons (True)
  End If

End Sub

'.....
Private Sub ShowSpecialButtons(show As Boolean)

  Dim i As Integer
  Dim a As Object

  For i = 0 To BUTTONARRAYSIZE
    If buttonArray(i).special Then
      For Each a In frmObj.Controls
        If TypeOf a Is Frame Then
          If a.Name = buttonArray(i).strName Then
            a.Visible = show
          End If
        End If
        If TypeOf a Is CommandButton Then
          If a.Name = buttonArray(i).cmdName Then
            a.Visible = show
          End If
        End If
      End If
    End If
  End If

```



```

        Next
    End If
Next

End Sub

'.....
Private Sub ResetAllButtons(bForeColor As Boolean)

    Dim frm As Object

    If bForeColor Then
        For Each frm In frmObj.Controls
            If TypeOf frm Is Frame And frm.Name <> "BigFrame" And frm.Name <>
"SmallFrame" Then
                frm.BackColor = m_BgColor
                frm.ZOrder 1
            End If
        Next
        frmObj.BigFrame.BackColor = m_FgColor
    Else
        For Each frm In frmObj.Controls
            If TypeOf frm Is Frame Then
                frm.BackColor = m_BgColor
            End If
        Next
    End If

End Sub

'.....
Private Sub clsScannerEventClass_NextVK(ByVal clsID As String)

    If clsID = MyScanCLSID Then
        frmObj.ButtonNext.SetFocus
        frmObj.FrameNext.BackColor = m_FgColor
        curCoords.x = 2
        curCoords.Y = 3
    Else: ResetAllButtons (False)
    End If

End Sub

'.....
Private Sub clsScannerEventClass_CloseVK()

    Unload frmObj
    Set frmObj = Nothing

End Sub

'.....
Private Sub frmObj_StartScan()

Dim vkEventClassObj As Object

    ResetAllButtons False
    Set vkEventClassObj = CreateObject("VKEEventClass.clsVKEEventClass")
    vkEventClassObj.VKInit 2, "1.2.3. 4.5.6. ", MyScanCLSID
    Set vkEventClassObj = Nothing

End Sub

'.....
Private Sub Configuration_Configure(ByVal Data As String, ByVal PubID As
String)

```

```

If Data = "Visible" Then
    frmObj.Visible = True
    frmObj_StartScan
ElseIf Data = "Invisible" Then
    frmObj.Visible = False
    Dim vkEventClassObj As Object
    Set vkEventClassObj = CreateObject("VKEventClass.clsVKEventClass")
    vkEventClassObj.VKRemoveMe MyScanCLSID
    Set vkEventClassObj = Nothing
ElseIf Data = "Resize" Then
    frmObj.Hide
    Load ResizableForm
    ResizableForm.show
    Unload frmObj
ElseIf Data = "End Communicator" Then
    Unload frmObj
    Set frmObj = Nothing
    Unload Me
End If
Exit Sub

End Sub

'.....
Private Sub clsScannerEventClass_HideVK()

    frmObj.Visible = False

End Sub

```

Κώδικας 15: Κώδικας βασικής κλάσης για την υποστήριξη της σάρωσης (Scannable.cls)

Τέλος ακολουθούν οι κλάσεις έκδοσης δεδομένων και συνδρομής σε δεδομένα.

Κώδικας για την κλάση έκδοσης δεδομένων (Publisher.cls)

```

Option Explicit

'.....
Public Function Publish(ByVal Data As String)

'.....
Dim objEventClass As Object
Dim PubID As String
Dim objCLSID As Object

'.....
'.. Class ID .....
Set objCLSID = CreateObject("ClassIDUtil.clsClassID")
PubID = objCLSID.GetClassID("Example.Publisher")

'.....
Set objEventClass = CreateObject("Interface.Character")
'..... Class ID .....
objEventClass.Communicate Data, PubID
'.....
Set objEventClass = CreateObject("Interface.Word")
'..... Class ID .....
objEventClass.Communicate Data, PubID
'.....
Set objEventClass = CreateObject("Interface.Sentence")

```

```

'..... Class ID .....
objEventClass.Communicate Data, PubID
'.....
Set objEventClass = CreateObject("Interface.Document")
'..... Class ID .....
objEventClass.Communicate Data, PubID

'.....
Set objEventClass = Nothing

End Function

```

Κώδικας 16: Κώδικας για την κλάση έκδοσης δεδομένων (Publisher.cls)

Κώδικας για την κλάση συνδρομής σε δεδομένα (Subscriber.cls)

```

Option Explicit

'.. Interface.dll
'..
'..
'.. (ScannableForm.Button1.Caption = Data)
Implements Interface.Character
Implements Interface.Word
Implements Interface.Sentence
Implements Interface.Document

Private Sub Character_Communicate(ByVal Data As String, ByVal PubID As String)
End Sub

Private Sub Configuration_Configure(ByVal Data As String, ByVal PubID As
String)
End Sub

Private Sub Document_Communicate(ByVal Data As String, ByVal PubID As String)
End Sub

Private Sub Sentence_Communicate(ByVal Data As String, ByVal PubID As String)
End Sub

Private Sub Word_Communicate(ByVal Data As String, ByVal PubID As String)
End Sub

```

Κώδικας 17: Κώδικας για την κλάση συνδρομής σε δεδομένα (Subscriber.cls)

Αναφορές

1. «ΑΙΝΕΙΑΣ - Ανάπτυξη Ευέλικτων Συστημάτων Εναλλακτικής και Επαυξητικής Διαπροσωπικής Επικοινωνίας μέσω Υπολογιστών και του Διαδικτύου», Τεχνικό Δελτίο, ΕΠΕΤ II, Αθήνα, 1998.
2. Γ. Κουρουπέτρογλου και Σ. Λιάλιου: «Τεχνική Περιγραφή και Ανάλυση των Χαρακτηριστικών και των Επικοινωνιακών Αναγκών των ΑΜΕΑ-Στόχου του Έργου – Θεωρητική Προσέγγιση», Τεχνική Έκθεση Π1.1.α., Έργο ΑΙΝΕΙΑΣ, ΕΠΕΤ II, Αθήνα, 2000.
3. Κ. Βίγλα, Γ. Κουρουπέτρογλου, Ρ. Πίτα και Μ. Αθουσάκη: «Τεχνική Περιγραφή και Ανάλυση των Χαρακτηριστικών και των Επικοινωνιακών Αναγκών των ΑΜΕΑ-Στόχου του Έργου – Πειραματική Προσέγγιση», Έργο ΑΙΝΕΙΑΣ, Τεχνική Έκθεση Π1.1.β, ΕΠΕΤ II, Αθήνα, 2000.
4. Α. Πίνο και Γ. Κουρουπέτρογλου: «Ανάλυση και Τεχνικές Προδιαγραφές του Αντικειμενοστραφούς Πλαισίου Ανάπτυξης Εφαρμογών ΟΔΥΣΣΕΑΣ», Τεχνική Έκθεση Π3.1., Έργο ΑΙΝΕΙΑΣ, ΕΠΕΤ II, 2000.
5. Α. Πίνο και Γ. Κουρουπέτρογλου: «Υλοποίηση του Πλαισίου ΟΔΥΣΣΕΑΣ και Εγχειρίδιο Χρήσης», Τεχνική Έκθεση Π3.2., Έργο ΑΙΝΕΙΑΣ, ΕΠΕΤ II, 2000.
6. Α. Πίνο και Γ. Κουρουπέτρογλου: «Τεχνικές Προδιαγραφές και Σενάριο Επίδειξης Πιλοτικού Αυτόνομου Βοηθήματος Διαπροσωπικής Επικοινωνίας», Τεχνική Έκθεση Π4.1α., Έργο ΑΙΝΕΙΑΣ, ΕΠΕΤ II, 2001.
7. Κ. Ξιπτερίδη, Ε. Μιτσόπουλου και Γ. Κουρουπέτρογλου: «Τεχνικές Πρόσβασης σε Υπολογιστικά Περιβάλλοντα», Τεχνική Έκθεση Π4.1β., Έργο ΑΙΝΕΙΑΣ, ΕΠΕΤ II, 2001.