

UNIVERSITA' DEGLI STUDI DI PADOVA

FACOLTA' DI SCIENZE STATISTICHE

CORSO DI LAUREA IN STATISTICA E TECNOLOGIE INFORMATICHE



TESI DI LAUREA

**PROGETTAZIONE E REALIZZAZIONE DI UN SISTEMA
PER LA SOMMINISTRAZIONE DI QUESTIONARI ONLINE**

RELATORE: CH.MO PROF. LUIGI FABBRIS

CORRELATORE: DOTT. GIUSEPPE CASTIELLO

LAUREANDA: SIMONETTA PARISI

ANNO ACCADEMICO: 2006 / 2007

Dedicata ad:

A. M.

*È solo con il cuore che si può vedere nel modo giusto,
ciò che è essenziale è invisibile agli occhi.*

Antoine de Saint Exupéry

INDICE:

1. CAPITOLO 1: OBIETTIVI DEL PROGETTO	1
1.0. Il progetto	1
1.1. Introduzione	3
1.2. Il questionario	3
1.2.1. Fasi di un'indagine tramite questionario	3
1.2.2. Fonti d'errore	4
1.2.3. Le domande nel questionario	5
1.3. Tipologia di controlli	8
1.3.1. Quando eseguire i controlli	9
1.3.2. Caso pratico	9
1.4. Metodi ' <i>computer assisted</i> '	14
1.4.1. Metodi ' <i>computer assisted</i> ' in sigle	14
1.4.2. Vantaggi e svantaggi dei metodi ' <i>computer assisted</i> '	16
2. CAPITOLO 2: METODOLOGIA PER LA CREAZIONE DI QUESTIONARI ONLINE	19
2.0. Elementi che costituiscono il questionario Web	19
2.1. Elementi che costituiscono le domande	20
2.1.1. Codifica dei vari tipi di domande	21
2.2. Scelta di XML	22
2.2.1. Struttura del Questionario	24
2.2.2. Il DTD (Document Type Declaration)	33
2.2.3. XML Schema	41

2.3. Il Parser	42
2.4. Metodi e tecnologie	43
3. CAPITOLO 3: DESCRIZIONE DEL SISTEMA CREATO, POTENZIALITA' E LIMITI	45
3.0. Potenzialità del sistema	45
3.1. Lettura del file XML e validazione tramite DTD	46
3.1.1. Creazione delle classi	46
3.1.2. Salvataggio dei dati letti dal file XML	52
3.2. Creazione delle pagine Web	53
3.2.1. Creazione del contenuto dei controlli	53
3.3. Creazione del database	56
3.3.1. Salvataggio dei dati nel database	58
3.3.2. Inserimento delle risposte già date	59
3.4. La grafica	59
3.5. Limiti del sistema	61
4. CAPITOLO 4: CONCLUSIONI PROPOSITIVE	63
<i>Bibliografia</i>	<i>I</i>
<i>Sitografia</i>	<i>II</i>
<i>Ringraziamenti</i>	<i>III</i>

Capitolo 1: Obiettivi del progetto

1.0 Il progetto:

Il progetto descritto in questa tesi si propone di realizzare un sistema per la creazione di questionari on-line per indagini computer assisted.

Questo obiettivo si colloca all'interno di un ampio e complesso progetto di sistema di gestione per indagini con tecniche miste (telefoniche, on-line e via fax) dell'azienda Questlab s.r.l. di Mestre. Questa società si occupa, da anni, di indagini campionarie rivolte alle imprese, alle istituzioni, alle organizzazioni e alla popolazione, utilizzando metodologie e tecnologie 'computer assisted'.

Le fasi principali del sistema di gestione saranno:

1. la creazione del questionario e del campione,
2. l'invito a rispondere, il contatto e il sollecito ai rispondenti,
3. il data entry dei questionari,
4. il monitoraggio dell'andamento,
5. l'esportazione dati,
6. l'analisi finale dei dati.

La parte riguardante la mia applicazione sarà corrispondente al terzo punto dell'elenco precedente e in particolare si propone di creare il questionario web, a partire da un file XML e il relativo database dove salvare le risposte.

Nello specifico si vuole creare un'applicazione desktop che comprenda la lettura del file scritto in sintassi XML¹ (le cui caratteristiche principali saranno: gli elementi del questionario, le domande, i controlli di validità, i salti logici e i filtri), la creazione del database con i campi del questionario, dove verranno salvate le risposte (costruito grazie all'uso del Parser XML che creerà o modificherà la tabella del questionario relativa all'indagine) e l'impostazione del questionario web con i relativi controlli (la creazione del file .html o .aspx collegati ad un file .js che conterrà il codice in javascript per effettuare i controlli).

Ci sono molti sistemi e relativi software per la creazione di questionari web, che prendono comunemente il nome di Web System Interview, tra i quali si possono citare alcuni nomi come: '2ASK'² e 'CONFIRM'³ per metodi CAWI, 'ASKIA'⁴, 'BLAISE'⁵ e 'BELLVIEW'⁶ che supportano metodi CAPI, CATI e CAWI e non ultimo 'CAPTOR', che è stato sviluppato in collaborazione con la Facoltà di Scienze Statistiche dell'Università di Padova a partire dal 1996, per indagini CASI e la sua evoluzione 'CAPTOR 2' che supporta indagini CASI via internet. Ovviamente ne esistono molti altri, altrettanto validi, che non sono stati riportati.

¹ In questa prima fase, non verranno implementati dei metodi per creare il file XML in modo guidato. Verrà lasciato a sviluppi futuri, un metodo di creazione guidata con l'utilizzo di Wizard (una procedura informatica che permetterà all'utente di eseguire la creazione del file XML tramite una serie di passi successivi).

² Sviluppato dalla società tedesca '2ask'. (www.2ask.net)

³ Prodotto dalla società australiana 'CITEC Confirm'. (www.confirm.com)

⁴ Realizzato dalla società francese 'Askia'. (www.askia.com)

⁵ Sviluppato da 'Statistics Netherlands' (istituzione governativa olandese). (www.blaise.com)

⁶ Frutto dell'esperienza della società inglese 'Pulse Train Ltd'. (www.pulsetr.co.uk)

Nel nostro caso, è stato deciso di crearne uno *ad hoc* che risponda alle necessità dell'azienda, in base ai bisogni riscontrati in questi anni.

È necessaria però un'accurata fase di analisi e di studio delle metodologie e delle principali tecniche utilizzate nelle indagini statistiche.

1.1 Introduzione:

Ci sono due metodi per realizzare un'indagine e raccogliere informazioni ponendo delle domande, il primo è condurre un'intervista e il secondo è somministrare un questionario da far auto compilare.

1.2 Il questionario:

Il questionario è un insieme strutturato di domande, registrato su un supporto adeguato per la somministrazione (Fabbris 2003).

I questionari possono essere autosomministrati, se non c'è bisogno di un intervistatore e il rispondente compila da solo il questionario, o per rilevazioni tramite intervista, nelle quali un rilevatore compilerà il questionario per conto del rispondente, in base alle risposte date dall'intervistato. Inoltre si possono distinguere questionari con struttura: lineare, nei quali non sono presenti salti logici e quindi domande filtro, e gerarchica (o con forma arborea), caratterizzati dalla presenza di sotto-domande o domande filtro.

Questo metodo è spesso preferito in quanto meno costoso in termini di denaro e tempo impiegato per la realizzazione e la successiva raccolta e analisi dei dati.

1.2.1 Fasi di un'indagine tramite questionario:

Analizziamo nello specifico l'indagine tramite questionario. Questa si compone di diverse fasi: innanzitutto c'è bisogno di un'analisi esplorativa per approfondire

contenuti e caratteristiche tecniche, come tipo e numero delle domande e modalità di risposta.

Si dovranno poi individuare il campione e le procedure per raccogliere i dati (questionario telefonico, via web, postale ecc.) e successivamente costruire il questionario vero e proprio verificandone l'adeguatezza con un'indagine pilota.

Dopo la somministrazione e la raccolta delle risposte si dovrà provvedere a salvarle in seguito ad un'opportuna codifica. Infine ci sarà la vera e propria analisi dei dati con relativa interpretazione (estendendo i risultati ottenuti sul campione a tutto l'universo di riferimento) e produzione del report.

1.2.2 Fonti d'errore:

Gli errori attinenti alla rilevazione si possono classificare in tre gruppi:

- errori nel rilevare i dati:
 - mancata rilevazione di un'unità: è il più grave tra gli errori di questo gruppo, in quanto rappresenta un fallimento totale nel tentativo di ottenere una risposta;
 - mancata risposta ad un quesito: certe volte si possono recuperare informazioni di una certa domanda (alla quale non è stata data una risposta), dalle risposte avute ad altri quesiti, purtroppo però, non sempre ciò è possibile. Per questo motivo, questo errore, rappresenta un fallimento, ma solo parziale;
 - errore nella risposta ad un quesito: questo è rappresentato dalla differenza tra la risposta data e quella che risulterebbe essere vera (molte volte, il "valore vero" non è però riscontrabile);
- gli errori di risposta si possono ulteriormente suddividere in:

- sistematici: errore che si riscontra in tutte le domande della stessa direzione (ad esempio tutte le domande sul reddito) e che quindi non si compensa nell'insieme;
- casuali: errore che si manifesta casualmente nelle varie unità e si compensa nel campione;
- Distinguendoli secondo la fonte, gli errori nella formazione dei dati, possono derivare da:
 - rilevatore: questi, inconsapevolmente, può esercitare un'azione distorsiva nei confronti dell'intervistato, anche solo con il proprio atteggiamento;
 - rispondente: questi, può rifiutarsi di collaborare oppure dare risposte accidentalmente o, addirittura, volutamente errate;
 - codificatori (o eventuali supervisori): questi possono codificare erroneamente le risposte date dagli intervistati.

1.2.3 Le domande nel questionario:

Il testo delle domande del questionario deve essere di semplice interpretazione e non dare adito ad ambiguità. Inoltre, per evitare errori dalle gravi conseguenze, le domande e le modalità di risposta devono essere pertinenti rispetto agli obiettivi prefissi (Fabbris 2003)

Iniziamo con una breve panoramica tra le due principali tipologie di domande. Bisogna innanzitutto differenziare le domande aperte dalle domande chiuse.

Si dicono aperte, domande che non prevedono anticipatamente delle risposte precodificate. Questa modalità lascia all'individuo libertà di risposta e d'espressione nella scelta del contenuto e della forma. Viene preferita per domande particolarmente delicate o delle quali non si possono definire a priori le modalità. Gli svantaggi di questo tipo di domande sono che: verranno compilate a fatica, o addirittura non compilate, da coloro i quali hanno problemi di concettualizzazione o problemi nello

scrivere, si rischia di cadere in errori di interpretazione e implicano molto lavoro per la registrazione e la codifica.

Le domande chiuse, invece, forniscono una serie di alternative già stabilite per le risposte. Questa modalità riduce i tempi di registrazione e codifica, fornisce un aiuto al rispondente e rende possibile una standardizzazione della distribuzione della risposta. Bisogna però, prestare attenzione al numero di opzioni fornite, in quanto troppe alternative concentrano l'attenzione sulla prima o sull'ultima (per le interviste telefoniche è quindi opportuno un metodo di rotazione delle risposte) e poche opzioni possono trascurare fatti importanti, infine le domande chiuse possono portare a rispondere a caso.

Analizziamo ora nello specifico le varie tipologie di domande che si possono presentare.

In base al tipo di struttura, le domande chiuse possono essere:

1. a risposta unica: che prevedono di poter scegliere una sola opzione tra quelle presenti (come ad esempio il sesso del rispondente)
2. a risposta multipla o multiresposta: che prevedono di poter scegliere una o più alternative tra quelle proposte (eventualmente vincolando il numero massimo di preferenze che possono essere selezionate)
3. in sequenza strutturata: nelle quali viene chiesto di definire l'ordine di preferenza o di importanza delle modalità di un certo fenomeno (ad esempio la Scala di Likert)

In base al tipo di risposte che prevedono abbiamo:

- a) *Domande qualitative*: sono domande le cui risposte prevedono delle modalità qualitative
- b) *Domande quantitative*: sono domande le cui risposte sono numeriche

In base alla loro funzionalità si possono distinguere:

- A. *Domande semplici*: sono le domande, sia aperte che strutturate, che non prevedono collegamenti con altre domande all'interno del questionario

- B. *Domande composte*: sono domande composte di due o più domande semplici
- C. *Domande concatenate*: al rispondere di una domanda viene risposto automaticamente alla domanda ad essa collegata
- D. *Domande filtro (o selettive)*: che permettono di suddividere i rispondenti in base a determinati criteri e formulare solo per loro specifiche domande al fine di evitare di scendere in domande dettagliate quando ciò è inutile (ad esempio chiedere informazioni riguardanti il coniuge ad un single) e di evitare di ricadere nel fenomeno della desiderabilità sociale (il rispondente cerca cioè di aggiustare la propria risposta in modo da non fare una “brutta figura”).
- E. *Domande di controllo (o trappola)*: servono per controllare la significatività e l'attendibilità di una risposta ottenuta precedentemente
- F. *Domande nascoste*: sono domande non visibili che si compilano automaticamente ricavando i dati da altre risposte
- G. *Domande dinamiche*: sono domande che dipendono dalla domanda principale (quantitativa). In particolare, deve essere possibile costruire in maniera automatica, tante domande quante indicate nella risposta della domanda principale.

Problema del “non so”: bisogna decidere se prevedere per determinate domande la possibilità della risposta ‘non so’ . Essa può sì rappresentare un ‘escamotage’ per coloro che non hanno voglia e/o tempo di rispondere alla domanda, ma evita le risposte forzate. Rispetto a una risposta sbagliata, o forzata, è preferibile ottenere una mancata risposta o un ‘non so’, infatti la prima “pesa” sui valori delle statistiche, mentre la seconda non “sposta” le tendenze emerse dalle risposte.

Una volta raccolte le risposte, è necessario salvarle in un database dove ogni rispondente è identificato univocamente da un codice e rappresenta un record, mentre ogni risposta è contenuta in una colonna differente. Le unità campionate saranno quindi le righe del nostro data set mentre le variabili saranno individuate dalle colonne. Le basi di dati statistiche hanno il vantaggio di essere sostanzialmente stabili, d'altra parte, però sono temporali e per loro natura molto complesse. E proprio per la loro natura non sarà

tanto necessario modificare i dati in esse contenuti, quanto effettuare controlli ed eventuali correzioni (anche prima di salvare i dati nel caso di controlli run time) .

In seguito al salvataggio, sarà necessario codificare alcune risposte (e prevedere un codice per i dati mancanti) in modo da dare alle informazioni raccolte la forma di dati che saranno poi analizzabili con tecniche statistiche appropriate. Bisognerà procedere con l'accorpamento dei dati in caso di modalità poco frequenti e solo a questo punto si potrà procedere con l'analisi e la successiva stesura del report finale.

1.3 Tipologia di controlli:

Esaminiamo ora i vari tipi di controlli che si possono effettuare sulle risposte.

Distinguiamo inizialmente le due tipologie principali: controlli di validità e controlli di coerenza.

Controlli di validità dei dati immessi: sono quei controlli che indicano se il dato immesso è valido, questi si dividono in:

- i) *Controlli di compatibilità*: i dati immessi in risposta devono essere dello stesso tipo di quelli previsti (ad esempio se viene chiesto l'anno di nascita, la risposta dovrà essere di tipo numerico)
- ii) *Controlli sul range*: i dati immessi devono essere contenuti nell'intervallo (ad esempio se si vuole sapere il numero di giorni lavorativi della precedente settimana, la risposta dovrà essere un numero compreso tra 0 e 7)
- iii) *Controlli di correttezza del formato*: servono per verificare che la risposta fornita appartenga al formato predefinito (ad esempio l'indirizzo e-mail o il codice fiscale)
- iv) *Controlli di completezza*: bisogna decidere se e quali campi possono essere lasciati vuoti e quindi non necessitano obbligatoriamente di una risposta

Controlli di coerenza tra i dati immessi: indicano se il dato immesso è compatibile con le altre risposte fornite e si dividono in:

- a) Controlli su domande composte: viene verificata la coerenza all'interno di domande composte e quindi contigue (ad esempio più percentuali che in totale diano 100)
- b) Controlli su domande sparse: viene verificata la coerenza tra domande diverse anche localizzate in punti diversi del questionario (ad esempio per le domande di controllo che devono obbligatoriamente essere distanti fra loro)

Sono inoltre possibili combinazioni dei vari tipi di controlli, come il controllo di completezza legato a quello di coerenza su domande sparse (ad esempio se la condizione professionale = occupato , la posizione della professione non deve essere mancante).

Infine deve essere possibile suddividere il questionario su più pagine non solo in base a salti logici, ma anche a discrezione del committente.

1.3.1 Quando eseguire i controlli:

Generalmente i controlli di validità vengono fatti in run time dal lato client, è infatti inutile inviare dei dati che sono sicuramente errati, notificando l'errore al rispondente. Questi controlli verranno implementati in Javascript.

I controlli di coerenza invece, possono essere affrontati in modo diverso, ovvero facendo presente al rispondente del riscontro dell'incoerenza nel questionario, ma lasciandogli la libertà di risolvere o meno l'errore riscontrato. In caso di mancata correzione, i dati verranno salvati ugualmente, ma verrà anche segnalato nel database che quel questionario andrà riverificato in un secondo momento parlando e ragionando direttamente con il rispondente. Questi controlli, invece, verranno fatti nel lato server.

1.3.2 Caso pratico:

Indagine ISTAT sui consumi delle famiglie italiane 1997. Classificazione delle regole di compatibilità per il controllo delle variabili qualitative (Manzari 1999)

1. Le variabili relative ai componenti dichiarati non devono avere valori mancanti
2. Esistenza della persona di riferimento
3. Unicità della persona di riferimento
4. La persona di riferimento può avere solo un coniuge
5. La persona di riferimento e il suo coniuge devono avere sesso diverso
6. Il coniuge della persona di riferimento deve essere coniugato se la persona di riferimento è coniugata
7. Il coniuge della persona di riferimento non deve essere coniugato se la persona di riferimento non è coniugata
8. L'ascendente della persona di riferimento deve avere almeno 35 anni
9. Il componente coniuge / convivente della persona di riferimento deve avere almeno 16 anni
10. Il componente non celibe deve avere almeno 16 anni
11. Il componente divorziato deve avere almeno 19 anni
12. Il componente con il dottorato deve avere almeno 23 anni
13. Il componente con la laurea deve avere almeno 21 anni
14. Il componente con il diploma universitario o laurea breve deve avere almeno 20 anni
15. Il componente con il diploma che permette l'accesso all'università deve avere almeno 17 anni
16. Il componente con la qualifica o licenza che non permette l'accesso all'università deve avere almeno 15 anni
17. Il componente con la licenza media deve avere almeno 13 anni
18. Il componente con la licenza elementare deve avere almeno 10 anni

19. Per il componente con età ≤ 14 anni non deve essere rilevata la condizione professionale
20. Per il componente con età ≤ 14 anni non deve essere rilevata la branca di attività economica prevalente
21. Per il componente con età ≤ 14 anni non deve essere rilevata la posizione nella professione
22. Per il componente con età ≥ 15 anni non deve essere mancante la condizione professionale
23. Il componente con condizione professionale = casalinga non deve essere di sesso maschile
24. Il componente con condizione professionale = pensionato non deve avere età ≤ 40 anni
25. Il componente con condizione professionale = in servizio di leva o servizio civile sostitutivo non deve essere di sesso femminile
26. Il componente con condizione professionale = in servizio di leva o servizio civile sostitutivo non deve avere età < 18 anni o età > 30 anni
27. Il componente con età ≥ 65 anni non deve avere condizione professionale disoccupato o in cerca di prima occupazione
28. Per il componente con condizione professionale \neq occupato / disoccupato non deve essere rilevata la branca di attività economica prevalente
29. Per il componente con condizione professionale \neq occupato non deve essere rilevata posizione nella professione
30. Per il componente con condizione professionale = occupato / disoccupato non deve essere mancante la branca di attività economica prevalente
31. Per il componente con condizione professionale = occupato non deve essere mancante la posizione nella professione

Classificazione personale delle regole a seconda delle varie tipologie di controlli e pseudo - codifica dei controlli:

1. Controllo di completezza → ciascuna variabile $\neq \emptyset$
2. Controllo di completezza → relazione con la persona di riferimento $\neq \emptyset$
3. Controllo sul range (vincolo) → numero persone di riferimento = 1
4. Controllo su domande sparse → $0 \leq \text{numero coniuge} \leq 1$
5. Controllo di coerenza su domande sparse → sesso persona di riferimento \neq sesso coniuge
6. Controllo di coerenza su domande sparse → se stato civile(persona di riferimento) = coniugato allora stato civile(coniuge) = coniugato
7. Controllo di coerenza su domande sparse → se stato civile(persona di riferimento) \neq coniugato allora stato civile(coniuge) \neq coniugato
8. Controllo di coerenza su domande sparse → età(ascendente della persona di riferimento) ≥ 35 anni
9. Controllo di coerenza su domande sparse → età(convivente / coniuge della persona di riferimento) ≥ 16 anni
10. Controllo di coerenza su domande sparse → se componente \neq celibe allora età ≥ 16 anni
11. Controllo di coerenza su domande sparse → se componente = divorziato allora età ≥ 19 anni
12. Controllo di coerenza su domande sparse → se titolo di studio = dottorato allora età ≥ 23 anni
13. Controllo di coerenza su domande sparse → se titolo di studio = laurea allora età ≥ 21 anni

14. Controllo di coerenza su domande sparse → se titolo di studio = diploma universitario / laurea breve allora età ≥ 20 anni
15. Controllo di coerenza su domande sparse → se titolo di studio = diploma allora età ≥ 17 anni
16. Controllo di coerenza su domande sparse → se titolo di studio = qualifica / licenza allora età ≥ 15 anni
17. Controllo di coerenza su domande sparse → se titolo di studio = licenza media allora età ≥ 13 anni
18. Controllo di coerenza su domande sparse → se titolo di studio = licenza elementare allora età ≥ 10 anni
19. Domanda Filtro → se età ≤ 14 anni allora condizione professionale = \emptyset
20. Domanda Filtro → se età ≤ 14 anni allora branca di attività economica prevalente = \emptyset
21. Domanda Filtro → se età ≤ 14 anni allora posizione nella professione = \emptyset
22. Controllo di completezza e coerenza su domande sparse → se età ≥ 15 anni allora condizione professionale $\neq \emptyset$
23. Controllo di coerenza su domande sparse → se professione = casalinga allora sesso = F
24. Controllo di coerenza su domande sparse → se professione = pensionato allora età ≥ 40 anni
25. Controllo di coerenza su domande sparse → se professione = in servizio di leva / servizio civile sostitutivo allora sesso = M
26. Controllo di coerenza su domande sparse → se professione = in servizio di leva / servizio civile sostitutivo allora $18 \text{ anni} \leq \text{età} \leq 30 \text{ anni}$
27. Controllo di coerenza su domande sparse → se età ≥ 65 anni allora professione \neq disoccupato / in cerca di prima occupazione

28. Domanda Filtro → se professione \neq occupato / disoccupato allora attività economica prevalente = \emptyset
29. Domanda Filtro → se professione \neq occupato allora posizione nella professione = \emptyset
30. Controllo di completezza e coerenza su domande sparse → se professione = occupato / disoccupato allora attività economica prevalente \neq \emptyset
31. Controllo di completezza e coerenza su domande sparse → se professione = occupato allora posizione nella professione \neq \emptyset

1.4 Metodi 'computer assisted':

In questi ultimi anni le rilevazioni dei dati sono sempre più "computer assisted" ovvero fatte mediante l'uso di strumenti elettronici e in modo automatizzato.

In questo modo, vengono gestiti automaticamente i vari passi dell'indagine statistica, dalla composizione del questionario all'analisi dei risultati.

Il questionario stesso sarà un programma che verrà visualizzato sullo schermo del proprio computer e che, man mano, registrerà in tempo reale sul database le risposte fornite dall'utente, tramite la connessione Internet.

1.4.1 Metodi 'computer assisted' in sigle:

I sistemi CASIC (Computer Assisted Survey Information Collection) sono l'insieme delle nuove tecniche d'indagine per le rilevazioni assistite da computer e si dividono in:

- Tecniche che prevedono la presenza dell'intervistatore:
 - o CAPI (Computer Assisted Personal Interviewing): le interviste fatte con questa tecnica vengono effettuate faccia a faccia da un rilevatore dotato di un computer che gestisce il questionario elettronico, direttamente nella casa dell'intervistato;

- CATI (Computer Assisted Telephone Interviewing): sono interviste telefoniche con l'ausilio di computer. L'intervistatore propone le domande che si visualizzano sullo schermo del suo computer e registra le risposte dell'intervistato.
- Tecniche CASI (Computer Assisted Self-administered Interviewing) basate sull'auto compilazione del questionario:
 - TDE (Touch-tone Data Entry): questa tecnica si basa sull'utilizzo di un telefono multifrequenza collegato ad un computer e prevede che l'intervistato risponda a delle domande registrate, digitando la risposta sulla tastiera del proprio telefono. Le risposte verranno poi rilette dal computer per chiedere conferma all'utente. Questa tecnica può essere valida quando i dati da rilevare sono numerici o codificabili numericamente;
 - VRE (Voice Recognition Entry): è analoga al TDE, ma in questo caso l'intervistato risponderà alle domande poste dal computer in modo vocale, pronunciando le cifre e la conferme o meno della propria risposta. È sconsigliata l'applicazione di questo metodo a interviste medio – lunghe;
 - DBM (Disk By Mail): con questo metodo l'intervistato riceve per posta il floppy con il codice del questionario, che dovrà installare nel proprio computer per poi rispondere alle domande proposte;
 - EMS (E-mail Survey): questa tecnica è un'evoluzione di quella precedente, infatti l'intervistato riceve e ritrasmette il questionario non più per posta, ma per e-mail;
 - WBS (Web Based Survey): sono indagini basate sul web, che si svolgono on line, grazie all'utilizzo di internet. Sul server viene allestito il questionario web al quale gli intervistati potranno accedere mediante connessione ad internet, completarlo e inviare in tempo reale le proprie

risposte. È questo il settore emergente dei metodi CASI e probabilmente anche il più promettente.

- Tecniche di acquisizione dati non correlate all'intervista:
 - o ODE (Optical Data Entry): questa tecnica prevede la scansione e riconoscimento ottico dei codici o dei caratteri. La lettura ottica dei modelli è però più lenta dell'imputazione manuale;
 - o EDI (Electronic Data Interchange): è un metodo che, grazie allo scambio elettronico dei dati, consente il superamento del questionario e può realizzare un dialogo diretto dei sistemi informativi, sollevando il rispondente da molti oneri.

1.4.2 Vantaggi e svantaggi dei metodi 'computer assisted':

Esaminiamo nello specifico i vantaggi di questo metodo:

- Il processo è quasi interamente automatizzato: la generazione e la somministrazione del questionario, il controllo delle risposte fornite, la registrazione dei dati, il controllo della qualità e l'elaborazione degli indici di sintesi;
- I costi sono meno elevati perché non sono necessari spostamenti dei rilevatori (ad eccezione dei metodi CAPI);
- Maggior tempestività dei risultati: la trasmissione dei dati avviene in tempo reale tramite la connessione Internet, così come la registrazione delle risposte su database;
- Diminuzione dei tempi per la conduzione dell'intera indagine;
- Semplicità delle operazioni: la prima revisione viene effettuata in modo automatico;

- Controlli (che approfondiremo in seguito): effettuati su alcune risposte, fornendo avvisi al rispondente in run time o a posteriori dopo una rivisita delle risposte. I controlli si dividono in due categorie:
 - o Di validità: ovvero che i dati forniti in risposta siano compatibili con i tipi di dati richiesti (ad esempio che nel campo età sia contenuto un valore compreso tra 0 e 120);
 - o Di coerenza: serve ad escludere che i dati rivelino problemi di qualità, confrontandoli con altre domande presenti nel questionario o con il sistema delle statistiche già disponibili;
- Controllo della qualità delle informazioni raccolte: gli errori vengono ridotti grazie alla possibilità di fornire un help in linea o finestre per ulteriori spiegazioni, possono essere individuati già in fase di rilevazione e venir corretti in modo automatico, inoltre, non essendoci più un rilevatore, verranno evitati gli errori di trascrizione o di registrazione delle risposte fornite.
- Minor numero di domande visualizzate per il rispondente: nel caso di presenza di domande filtro o salti logici, il programma provvederà in modo automatico alla visualizzazione di tutte e sole le domande per quel dato rispondente, evitando di gravare inutilmente sulla sua attenzione e il suo tempo.

Passiamo ora ad analizzare i problemi e gli svantaggi del metodo *'computer assisted'*:

- Le tecniche CASI (in particolare EMS e WBS), possono essere un po' rischiose, principalmente per le indagini sulle famiglie, dato che prevede il possesso di un computer collegato ad Internet (largamente in uso per la maggior parte delle famiglie, ma difficile se si pensa a nuclei familiari composti solo di persone anziane, pertanto, questi ultimi dovranno essere raggiunti tramite indagine CATI, CAPI, ...);
- Un problema non indifferente è quello del layout, che deve essere compatibile con tutti i browser web e le diverse interfacce grafiche dei computer in modo da evitare una diversa organizzazione delle domande rispetto a quella prevista, portando così ad una difficile lettura del questionario da parte del rispondente;

- La sicurezza del trasferimento delle informazioni e la privacy (informativa ai sensi dell'art. 13 del DLGS 196/03);
- Consentire l'accesso al questionario solo agli utenti presenti nel campione. Ciò è realizzabile mediante l'uso di uno username e una password generati casualmente per ogni rispondente nel campione.

Capitolo 2: Metodologia per la creazione di questionari online

Per poter creare un questionario online dobbiamo innanzitutto sapere di quali elementi si compone e successivamente quali sono le caratteristiche principali delle domande di cui è costituito.

2.0 Elementi che costituiscono il questionario Web:

- *Intestazione:*
 - Loghi dei committenti e dell'azienda incaricata di effettuare l'indagine⁷
 - Link sulle precedenti indagini e rilevazioni
 - Informazioni sul questionario e sull'indagine statistica
 - In automatico dal sistema viene presa la data di compilazione

- *Corpo:*
 - Testi di introduzione
 - Domande
 - Eventuali help, spiegazioni o esempi
 - Risposte
 - Controlli associati alle risposte

⁷ La determinazione dell'intervistato a rispondere, è tanto maggiore, quanto più è ampio il consenso percepito dal campione designato e quanto più è elevato il senso di appartenenza alle istituzioni che promuovono l'indagine (Fabbris 2003).

- *Footer:*
 - o Funzione di invio dei dati inseriti nel questionario
 - o Informativa sulla privacy
 - o Ringraziamenti
 - o Loghi

2.1 Elementi che costituiscono le domande:

- Codice o numero della domanda
- Introduzione
- Testo della domanda
- Eventuali spiegazioni
- Avvertenze
- Testo delle eventuali modalità di risposta
- Help, spiegazioni o esempi
- Randomizzazione delle risposte, dove previste, per ogni singola domanda
- Controlli associati alle risposte
- Localizzazione e internazionalizzazione dei testi⁸
- Icone o immagini

⁸ Metodi utilizzati per consentire l'accessibilità del sistema in varie lingue e che verranno trattati anche in seguito.

2.1.1 Codifica dei vari tipi di domande:

1. Per le domande aperte è possibile prevedere un semplice 'TextBox', oppure un campo 'memo' (ovvero un 'TextBox' con proprietà TextMode impostata a MultiLine) nel caso di risposte molto lunghe o delle quali non si può stabilire una lunghezza massima (ad esempio commenti)
2. Per le domande strutturate, fatta eccezione per le domande gerarchizzate, si possono utilizzare 'ListBox', 'CheckBoxList', 'RadioButtonList' e 'DropDownList' (menù a tendina).

Per le domande gerarchizzate si possono utilizzare dei 'TextBox' o 'DropDownList', associati ad ogni opzione di risposta, ad ognuno dei quali è associato un controllo sulla validità dei dati immessi nei 'TextBox' e uno sull'eventuale mutua esclusione (se si decide che più opzioni non possano avere uguale posizione di gradimento).

Le domande a doppia alternativa si possono codificare come CheckBox o RadioButton (selected=TRUE se si desidera che una delle due modalità venga presa come predefinita).

Per le domande a risposta multipla una possibile codifica è fatta mediante 'CheckBoxList', mentre per le domande strutturate a scelta esclusiva abbiamo 'RadioButtonList' e 'ListBox' (per tutti è possibile definire che un'opzione sia predefinita utilizzando il comando selected=TRUE in corrispondenza della modalità scelta).

Alle domande concatenate sarà associata una procedura che al rispondere di una, completerà automaticamente anche l'altra.

Per le domande filtro bisognerà associare ad ogni modalità la domanda successiva da visualizzare impostando, ad esempio, un campo "NextQuestionID".

Le domande di controllo verranno trattate come domande semplici o composte in base alla loro struttura, solo che ad esse verranno associate dei controlli.

Per le domande nascoste basterà semplicemente impostare la proprietà Visible=FALSE e trattarle come domande concatenate.

Le domande dinamiche vengono risolte, inserendo in un ciclo, il codice per la creazione delle singole domande e fatto iterare tante volte, quante ne indica il campo risposta della domanda principale.

Infine per le domande composte si è deciso di procedere in forma tabellare. Esistono moltissime combinazioni di domande semplici, ed essendo molto difficile prevederle tutte, si è scelto di procedere creando dei modelli. Verranno quindi previsti dei modelli di base secondo i quali verrà creata la struttura della domanda composta ed in base a dei parametri inseriti in ingresso verrà creata la domanda.

2.2 Scelta di XML:

Per costruire la struttura del questionario, è stato deciso di utilizzare il meta linguaggio XML il cui acronimo sta per eXtensible Markup Language. Si tratta di uno standard di descrizione di documenti, progettato per la descrizione dei dati e caratterizzato da una struttura gerarchica, inventato dal comitato W3C (World Wide Web Consortium) guidato da Tim Berners Lee, l'inventore dell'HTML.

XML fornisce un insieme di regole sintattiche, dette specifiche, per mettere dati strutturati in un file di testo, configurando la struttura di documenti e dati.

Questo standard è utile per due tipi di applicazioni Web:

1. Creazione di documenti :

Diversamente dall'HTML, XML non ha degli elementi predefiniti, ma saranno noi a costruirli con determinati attributi e funzioni specifiche di ogni elemento. Il documento XML, come detto precedentemente, è caratterizzato da una struttura

gerarchica ed è quindi composto da elementi che possono contenere altri elementi e attributi. La sintassi XML è molto rigida e severa proprio per permettere un'interpretazione rigorosa.

2. Scambio dei dati:

XML essendo orientato alla descrizione dei dati, può essere utilizzato come piattaforma per lo scambio di informazioni tra applicazioni (ovvero la conservazione in una struttura XML di dati presi da fonti esterne o registrati nella struttura stessa).

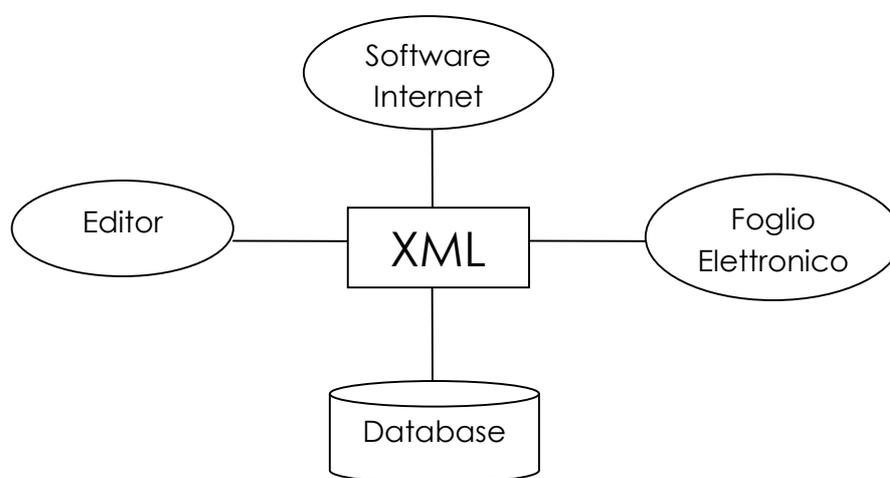


Fig. 1: XML come piattaforma per lo scambio dei dati

Un documento XML è composto da una serie di elementi, che a loro volta possono contenere altri sotto elementi o testo. Ad ogni elemento possono essere associati una serie di attributi, cioè informazioni che descrivono le proprietà dell'elemento.

Essendo una struttura gerarchica a forma arborea avremo necessariamente un nodo padre o radice, che nel nostro caso sarà "questionario", che conterrà tutti gli elementi figli.

Tutti i documenti XML devono essere ben formati (well done), in particolare, nel nostro caso, vogliamo che siano validi (condizione più restrittiva della precedente).

Affinché un documento sia ben formato, deve rispettare alcune regole di base: ad ogni tag aperto, deve corrispondere un tag di chiusura, ci deve essere uno ed un solo

elemento a livello radice, i valori degli attributi vanno inseriti tra virgolette o apici, gli elementi devono essere nidificati in modo opportuno (come detto precedentemente, la struttura dell'XML è gerarchica e a forma di albero, quindi tutti i tag andranno chiusi con ordine inverso rispetto a quelli di apertura) ed infine, essendo l'XML case sensitive, il nome degli elementi e degli attributi deve essere il medesimo, distinguendo maiuscole e minuscole.

Ma noi abbiamo bisogno di un documento XML valido.

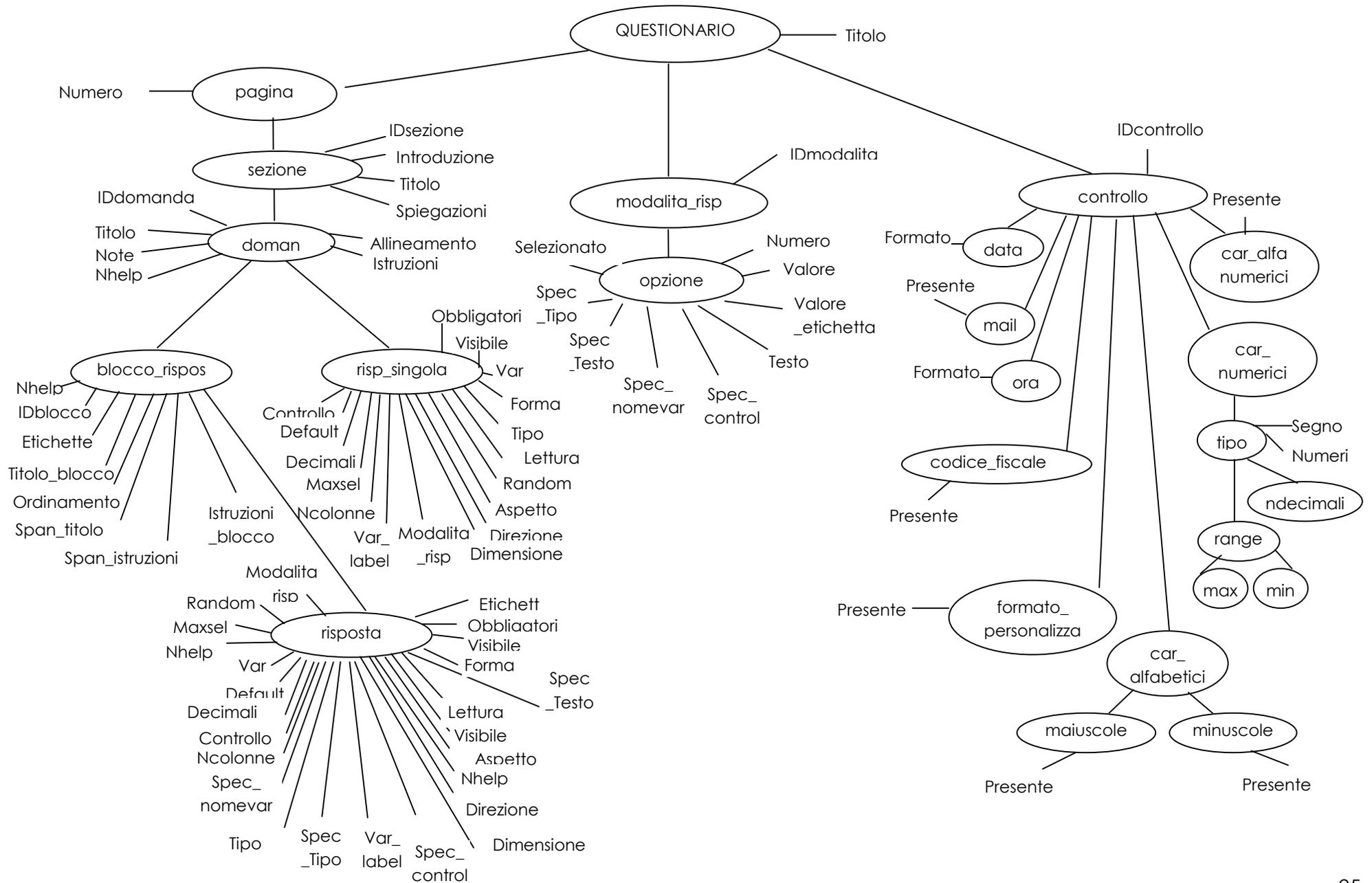
Si definisce documento XML valido, un documento XML ben formato, che è conforme alle regole di un DTD o uno schema XML. Tutti i documenti validi sono necessariamente ben formati, viceversa, non tutti i documenti ben formati sono anche validi.

2.2.1 Struttura del Questionario:

Proprio per la sua struttura gerarchica, il modo più semplice per rappresentare un file XML è l'utilizzo di un document tree, ovvero un albero che ne descriva la struttura.

L'elemento più in alto rappresenterà la radice dell'albero e, per ogni livello a seguire, ci saranno tutti gli elementi figli con i rispettivi attributi. Gli elementi sono rappresentati da forme ovali, collegate al padre (fatta eccezione per l'elemento QUESTIONARIO che è la radice) e agli eventuali figli con linee. Gli attributi sono identificati da linee che alla fine riportano il nome dell'attributo stesso.

Qui di seguito viene riportato il document tree per la rappresentazione del questionario, che descrive appunto, la struttura del nostro file XML.



Non esistono regole assolute o un metodo specifico per l'organizzazione logica del documento, è proprio per questo motivo che è necessaria un'analisi approfondita della realtà da rappresentare ed una conseguente scelta progettuale. Una volta decisa la struttura logica, si procede con la sua traduzione in struttura fisica caratterizzata dalla presenza di Tag.

Una traduzione possibile del Document tree in file XML è la seguente:

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE QUESTIONARIO SYSTEM "DTD1.DTD">
<QUESTIONARIO Titolo="Titolo del questionario">
  <pagina Numero="1">
    <sezione IDsezione="Codice"
      Introduzione="Eventuale introduzione della sezione"
      Titolo="Titolo della sezione"
      Spiegazioni="Spiegazioni della sezione">
      <domanda IDdomanda="Codice identificativo della domanda"
        Allineamento="affiancato/sottostante"
        Titolo_domanda="Titolo della domanda"
        Istruzioni_domanda="Istruzioni della domanda"
        Note="Eventuali note"
        Nhelp="Codice dell'help da visualizzare">
        <risp_singola Forma="text/memo/multiresponse/vuoto"
          Obbligatoria="si/no"
          Visibile="si/no"
          Lettura="si/no"
          Var="Variabile dove salvare la risposta"
          Tipo="Char/Float/BigInt/Int/SmallInt/DateTime/SmallDateTime/
          Numeric/Decimal/Money/SmallMoney/NVarChar/Real/Text/Bit"
          Dimensione="Dimensione del campo da creare nel database"
          Decimali="Numero di cifre dopo la virgola"
```

Var_label="Etichetta della variabile"

Controllo="Codice del controllo"

Default="Testo di default"

Random="si/no"

Aspetto="radio/check/dropdown"

Direzione="orizzontale/verticale"

Maxsel="numero massimo di selezioni"

Modalita_risp="Codice della modalita di risposta"

Ncolonne="Numero di colonne sulle quali dividere le opzioni">

</risp_singola>

<blocco_risposte IDblocco="Codice identificativo del blocco"

Nhelp="Codice dell'help da visualizzare"

Ordinamento="orizzontale/verticale"

Etichette="si/no"

Titolo_blocco="Titolo del blocco"

Span_titolo="Span del titolo"

Istruzioni_blocco="Istruzioni del blocco"

Span_istruzioni="Span delle istruzioni">

<risposta Forma="text/memo/multiresponse/vuoto"

Obbligatoria="si/no"

Visibile="si/no"

Lettura="si/no"

Var="Variabile dove salvare la risposta"

Tipo="Char/Float/BigInt/Int/SmallInt/DateTime/SmallDateTime/

Numeric/Decimal/Money/SmallMoney/NVarChar/Real/Text/Bit"

Dimensione="Dimensione del campo da creare nel database"

Decimali="Numero di cifre dopo la virgola"

Var_label="Etichetta della variabile"

Controllo="Codice del controllo"

Default="Testo di default"

Etichetta="Testo dell'etichetta"

Random="si/no"

Aspetto="radio/check/dropdown"

Direzione="orizzontale/verticale"

Maxsel="numero massimo di selezioni"

Spec_testo="testo di default nel campo specificare"

Spec_tipo="text/memo"

Spec_nomevar="variabile dove salvare il campo specificare"

Spec_control="codice del controllo"

Modalita_risp="Codice della modalita di risposta"

Ncolonne="Numero di colonne sulle quali dividere le opzioni"

Nhelp="Codice dell'help da visualizzare">

</risposta>

</blocco_risposte>

</domanda>

</sezione>

</pagina>

<modalita_risp IDmodalita="Codice della modalit  di risposta">

<opzione Numero="Numero dell'opzione"

Valore="Valore da salvare"

Selezionato="si | no"

Valore_etichetta="Valore dell'etichetta per l'elaborazione"

Testo="Testo dell'opzione"

Spec_testo="Testo dello specificare"

Spec_tipo="text/memo"

Spec_nomevar="Nome della variabile dove salvare la risposta"

Spec_control="Controllo da applicare allo specificare">

</opzione>

```
</modalita_risp>  
<controllo IDcontrollo="Codice del controllo">  
  <car_alfabetici Maiuscole="si/no"  
    Minuscole="si/no">  
  </car_alfabetici>  
  <car_numerici Numeri="interi/reali"  
    Segno="positivo/negativo/entrambi"  
    Ndecimali="Numero di cifre dopo la virgola"  
    Max="Estremo superiore"  
    Min="Estremo inferiore">  
  </car_numerici>  
  <mail Presente="si/no">  
  </mail>  
  <data Formato="gg/mm/aaaa/mm/gg/aaaa">  
  </data>  
  <ora Formato="12/24">  
  </ora>  
  <codice_fiscale Presente="si/no">  
  </codice_fiscale>  
  <car_alfanumerici Presente="si/no">  
  </car_alfanumerici>  
  <formato_personalizzato Struttura="Struttura personalizzata del controllo">  
  </formato_personalizzato>  
</controllo>  
</QUESTIONARIO>
```

Analizziamo ora la struttura che sta alla base del nostro documento XML.

La prima linea del file rappresenta la dichiarazione XML, che indica la versione (1.0) e la codifica (Unicode *utf-8*) utilizzata nel documento.

Il questionario, con il proprio titolo, è composto di una o più pagine numerate nelle quali saranno contenute le sezioni, dai controlli, se previsti, e le eventuali modalità di risposta. Il Parser in questo modo, ad ogni tag <pagina> andrà a creare una nuova pagina web.

Una sezione è identificata da un proprio ID ed ha un attributo introduzione nel quale verrà inserito il testo da visualizzare come introduzione, un titolo e delle spiegazioni.

All'interno della sezione possiamo trovare le domande che saranno caratterizzate da un codice identificativo ed avranno come attributi l'allineamento che potrà essere affiancato (nel caso in cui la risposta debba essere visualizzata a fianco della domanda) oppure sottostante (se la domanda dovrà essere collocata sopra e la risposta sotto), il titolo della domanda e le rispettive istruzioni, le note che verranno visualizzate sotto la risposta nel caso in cui l'allineamento sia sottostante e sotto le istruzioni nel caso di un allineamento 'affiancato', ed infine il codice identificativo dell'help, visualizzato come un '?' prima del codice della domanda, che non sarà altro che un collegamento ipertestuale che aprirà una nuova finestra con il testo delle spiegazioni della domanda.

A sua volta, ciascuna domanda può contenere o una risposta singola, o una serie di blocchi risposte.

Una risposta singola è caratterizzata da una serie di attributi: la forma, che potrà essere di tipo testo, memo, vuota o multiresponse, l'obbligatorietà della risposta (che verrà gestita con un controllo), la visibilità (necessaria nel caso di domande nascoste), un attributo che indica se la risposta è di sola lettura, il nome della variabile dove salvare la risposta, il tipo di dato da salvare nel database, la dimensione del campo creato e il numero di cifre decimali da impostare sempre nel database, una etichetta della variabile che servirà in fase di esportazione dei dati e di analisi degli stessi, il codice del controllo da applicare alla risposta, l'eventuale testo di default da visualizzare (che potrà essere un testo, un codice della modalità, un calcolo fatto con l'uso di altre risposte, una parola chiave - ad esempio *all* per selezionare tutte le opzioni oppure un'espressione - come *today* per le date -), ci saranno poi, nel caso di domande strutturate, degli attributi che

indicheranno il codice della modalità di risposta (che conterrà le varie opzioni della risposta) se è necessario randomizzare le opzioni di risposta, la direzione con cui verranno visualizzate le opzioni (in orizzontale o in verticale) e l'aspetto che assumeranno in visualizzazione ('RadioButton', 'CheckBox', o 'DropDownList' ovvero menù a tendina), inoltre si potrà specificare il numero massimo di selezioni (anche questo gestito da un controllo) e il numero di colonne su cui visualizzare l'elenco delle opzioni (nel caso di un numero molto elevato di alternative, potrebbe essere utile avere un impatto visivo meno ingombrante e quindi non visualizzarle più una sotto l'altra, ma dividerle su più colonne).

I blocchi risposte, invece, sono identificati da un codice del blocco, ed avranno come attributi il numero identificativo dell'help da visualizzare (in modo analogo a quello della domanda), l'ordinamento del blocco (orizzontale o verticale), il titolo del blocco e le eventuali istruzioni, e per entrambe, la possibilità di uno span (ovvero un numero che indicherà quanti blocchi faranno parte di uno stesso gruppo e che avranno lo stesso titolo, le stesse istruzioni o entrambe) ed infine un attributo che indica la presenza delle etichette all'interno del blocco.

Un blocco risposte può essere formato da più risposte composte in modo analogo alle risposte singole, anche in questo caso avremo come attributi: Forma, Obbligatoria, Visibile, Lettura, Var, Tipo, Dimensione, Decimali, Var_label, Controllo, Default, Random, Aspetto, Direzione, Maxsel, Modalità_risp, Ncolonne e in aggiunta avremo il numero con il codice dell'help, il valore dell'etichetta da visualizzare e una serie di attributi per la gestione del campo "specificare", che saranno Spec_testo, Spec_tipo, Spec_nomevar e Spec_control.

All'interno del questionario, come detto precedentemente, ci potranno essere le modalità di risposta, utilizzate nel caso di domande strutturate, identificate da un proprio codice (quelle a cui si riferisce l'elemento modalita_risp contenuto nel blocco risposte e nelle domande singole), che avranno una o più opzioni, tutte con un proprio numero, il valore da salvare nel database e un attributo che indica se dovrà essere selezionata o meno. Ogni opzione avrà un testo, un'etichetta che sarà chiamata valore_etichetta e che servirà in fase di esportazione dati ed un campo specificare. Questo elemento è stato inserito in fase di analisi, vista l'elevata frequenza con cui si presentavano domande composte di tipo strutturato, con alcune opzioni dove era

necessario specificare la risposta (basti pensare, per esempio, ad un elenco dove è presente un'opzione "altro" e si sia interessati a sapere esattamente cosa questo rappresenti, dicendo di specificare la propria scelta in un campo testuale visualizzato di fianco al testo dell'opzione). L'elemento specificare potrà essere di tipo text o memo (generalmente si userà testo per valori e memo per commenti) in base alla lunghezza che si desidera abbia il campo di specificazione. Si compone di un eventuale testo di default, del nome della variabile dove salvare il contenuto e di un codice che si riferisce ai controlli da effettuare.

Passiamo infine alla struttura del controllo identificato da un IDcontrollo e composto di otto elementi: car_alfabetici, car_numerici, car_alfanumerici, ora, data, codice_fiscale, mail e formato_personalizzato.

car_alfabetici, controlla che nel campo siano presenti solo caratteri alfabetici, con la presenza o meno di lettere minuscole e maiuscole in base a quanto è stato specificato.

car_numerici, verifica che siano presenti solo numeri con eventuale segno positivo o negativo di tipo interi o reali. È inoltre presente un elemento che servirà nel caso in cui il valore inserito nel campo da controllare debba appartenere ad un intervallo di specifica, in questo modo, inserendo l'estremo inferiore e superiore si verificherà che tale vincolo non sia violato.

car_alfanumerici, accerta che siano presenti solo caratteri di tipo alfanumerico e non simboli.

ora, è un controllo sulla struttura del campo, infatti si potranno inserire gli orari secondo il formato 12 ore oppure 24 ore.

data, è anch'esso un controllo sulla struttura del campo ed analogamente a quanto detto per il controllo sull'ora, ci saranno due formati diversi del tipo gg.mm.aaaa e mm.gg.aaaa.

codice_fiscale, controlla che il formato sia compatibile con quello dei codici fiscali, ovvero 16 caratteri, di cui i primi sei siano lettere seguiti da due cifre, poi ancora una lettera, due cifre, una lettera, tre cifre ed infine ancora una lettera.

mail, verifica il formato del campo che contiene un indirizzo e-mail (ad esempio che sia presente una '@', un '.' e che non ci siano caratteri particolari all'interno dell'indirizzo, come ad esempio gli spazi).

formato_personalizzato, accerta che la struttura del valore inserito come risposta corrisponda alle specifiche definite nel campo formato_personalizzato secondo la seguente codifica: i simboli vengono trattati come tali, 'A' corrisponde ad una lettera maiuscola, 'a' ad una lettera minuscola e '9' ad un numero (ad esempio per controllare che un campo sia formato di due lettere maiuscole, una minuscola, un punto e 3 numeri si dovrà inserire la seguente codifica 'AAa.999').

Ad un primo impatto la struttura può sembrare un po' appesantita da tutti questi elementi, ma è stato deciso, dopo una lunga analisi, di generalizzarla il più possibile per renderla maggiormente flessibile ed adattabile ad un numero maggiore di questionari.

Inoltre si è deciso di dividere le risposte singole da quelle in forma tabellare (e quindi appartenenti a dei blocchi risposte) nonostante avessero attributi del tutto analoghi (fatta eccezione per il codice dell'help, il testo dell'etichetta e gli attributi per il campo "specificare"), per rendere meno onerosa, all'utente, la creazione della struttura del questionario in formato XML.

2.2.2 Il DTD (Document Type Declaration):

È possibile, ma non obbligatorio, creare un modello DTD (Document Type Definition), un documento di testo personalizzato e orientato ai documenti elettronici, che descrive la struttura e il contenuto di una classe di documenti.

Grazie a questo documento, viene definito l'insieme di elementi e attributi personalizzati che si potranno poi andare ad usare. C'è già un DTD, sempre presente e creato automaticamente di default, nel quale vengono elencate le regole della specifica del linguaggio.

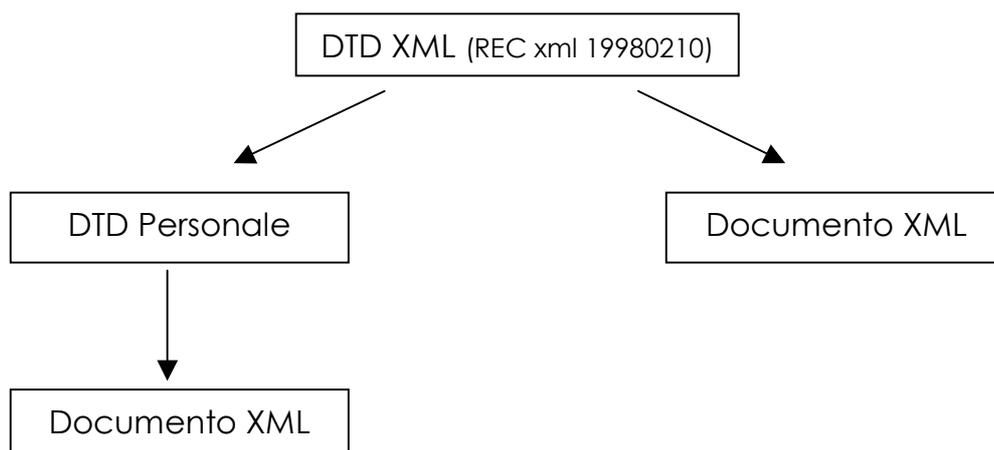


Fig. 2: Utilizzo di un DTD personale

Il DTD è un documento che descrive i Tag che si possono utilizzare in un file XML, ne definisce la loro struttura e l'eventuale lista di attributi presenti per ciascun elemento.

Definiamo ora la sintassi di un DTD:

- Dichiarazione del DTD : `<!DOCTYPE>`

Il DTD nel nostro caso, viene salvato su un file diverso dal documento XML e quindi dovrà essere presente, nel file XML, una dichiarazione che faccia riferimento al file DTD utilizzando il comando: `<!DOCTYPE questionario SYSTEM "questionario.dtd">`

- Dichiarazione di elemento : `<!ELEMENT>`

`<!ELEMENT pagina+>` indica che l'elemento pagina si può presentare una o più volte.

In generale, il simbolo che segue il nome dell'elemento ne rappresenta il numero di occorrenze:

- o '+' indica che l'elemento sarà presente almeno una volta
- o '*' indica che l'elemento può essere presente zero o più volte
- o ed infine '?' indica un elemento dicotomico, ovvero presente zero o al massimo una sola volta.

Se un Tag non contiene elementi figli, allora la dichiarazione sarà `<!ELEMENT elemento EMPTY>` nel caso in cui sia vuoto e `<!ELEMENT elemento (#PCDATA)>` se racchiude del testo.

Con la dichiarazione `<!ELEMENT elemento ANY>` l'elemento potrà essere vuoto oppure contenere altri elementi. Questa dichiarazione è molto comoda per elementi il cui contenuto non è definito a priori.

- Dichiarazione di una lista di attributi : `<!ATTLIST>`

`<!ATTLIST risposta`

`IdRisposta (#PCDATA) #REQUIRED`

`<!-- indica che l'attributo "IdRisposta" è obbligatorio -->`

`Testo (#PCDATA) # IMPLIED`

`<!-- indica che l'attributo "Testo" è opzionale -->`

`Selezionato (si|no) "no"`

`<!-- indica che l'attributo "Selezionato" può assumere valore si o no`

`e di default assume il valore "no" -->`

`>`

Il nostro DTD avrà quindi una struttura del tipo:

Un 'questionario', con un proprio titolo, avrà una o più 'pagine' e zero o più 'modalità di risposta' e 'controlli'.

`<!ELEMENT QUESTIONARIO (pagina+,modalita_risp*,controllo*)>`

`<!ATTLIST QUESTIONARIO Titolo CDATA #REQUIRED>`

Ogni 'pagina' identificata da un proprio numero, si comporrà di zero o più 'sezioni'.

`<!ELEMENT pagina (sezione*)>`

`<!ATTLIST pagina Numero CDATA #REQUIRED>`

Una ‘sezione’ sarà identificata da un ID ed avrà una propria introduzione, un proprio titolo, entrambi non obbligatori, e delle eventuali spiegazioni. Inoltre avrà zero o più ‘domande’.

```
<!ELEMENT sezione (domanda*)>
<!ATTLIST sezione
  IDsezione CDATA #REQUIRED
  Introduzione CDATA #IMPLIED
  Titolo CDATA #IMPLIED
  Spiegazioni CDATA #IMPLIED >
```

A sua volta la ‘domanda’, con un suo identificativo si comporrà di ‘risposte singole’ o ‘blocchi risposte’ (nel caso della risposta singola, il tag potrà essere presente al massimo una volta, mentre nel caso dei blocchi risposte, le occorrenze potranno essere zero o più). Gli attributi della domanda saranno l’allineamento che potrà essere affiancato o sottostante (di default sarà affiancato), il titolo della domanda, le istruzioni, le note e il codice identificativo dell’help a cui riferirsi, tutti facoltativi.

```
<!ELEMENT domanda (risp_singola?,blocco_risposte*)>
<!ATTLIST domanda
  IDdomanda CDATA #REQUIRED
  Allineamento (affiancato|sottostante) "affiancato"
  Titolo_domanda CDATA #IMPLIED
  Istruzioni_domanda CDATA #IMPLIED
  Note CDATA #IMPLIED
  Nhelp CDATA #IMPLIED >
```

Per la ‘risposta_singola’, che non conterrà altri elementi, avremo come attributi: Forma (che potrà essere di tipo text, memo, multiresponse o vuoto; di default sarà text), Obbligatoria (che assumerà valore si o no con quest’ultimo di default), Visibile (anch’essa di tipo si/no, ma in questo caso avremo come valore predefinito si), Lettura (che indica se la variabile sarà visualizzata in forma di sola lettura e che sarà di tipo si/no con quest’ultimo valore di default), Var che sarà opzionale, Tipo che assumerà i vari valori dei tipi di dato che possiamo trovare nel database e di default sarà Int, Dimensione, Decimali, Var_label, Controllo, Default, Maxsel, Modalita_risp,

Ncolonne, tutti non obbligatori Random (che assumerà valori si/no) , Aspetto (che potrà essere radio, check o dropdown) e Direzione (che potrà essere orizzontale o verticale con il primo valore di default).

```

<!ELEMENT risp_singola EMPTY>
<!ATTLIST risp_singola
  Forma (text/memo/multiresponse/vuoto) "text"
  Obbligatoria (si/no) "no"
  Visibile (si/no) "si"
  Lettura (si/no) "no"
  Var CDATA #IMPLIED
  Tipo (Bit/Char/Float/BigInt/Int/SmallInt/DateTime/SmallDateTime/Numeric/
    Decimal/Money/SmallMoney/NVarChar/Real/Text) "Int"
  Dimensione CDATA #IMPLIED
  Decimali CDATA #IMPLIED
  Var_label CDATA #IMPLIED
  Controllo CDATA #IMPLIED
  Default CDATA #IMPLIED
  Random (si/no) #IMPLIED
  Aspetto (radio/check/dropdown) #IMPLIED
  Direzione (orizzontale/verticale) "orizzontale"
  Maxsel CDATA #IMPLIED
  Modalita_risp CDATA #IMPLIED
  Ncolonne CDATA #IMPLIED
>

```

Il ‘blocco risposte’, che conterrà zero o più ‘risposte’, sarà identificato da un IDblocco ed avrà il codice dell’help, il titolo del blocco, le sue istruzioni, gli span, sia per il titolo che per le istruzioni, tutti facoltativi, la presenza delle etichette (che assumerà valore si/no con quest’ultimo di default) ed infine l’ordinamento che potrà essere orizzontale o verticale con il primo usato come valore predefinito.

```

<!ELEMENT blocco_risposte (risposta*)>

```

```

<!--ATTLIST blocco_risposte
  IDblocco CDATA #REQUIRED
  Nhelp CDATA #IMPLIED
  Ordinamento (orizzontale/verticale) "orizzontale"
  Etichette (si/no) "no"
  Titolo_blocco CDATA #IMPLIED
  Span_titolo CDATA #IMPLIED
  Istruzioni_blocco CDATA #IMPLIED
  Span_istruzioni CDATA #IMPLIED
-->

```

L'elemento 'risposta', che sarà vuoto, avrà degli attributi analoghi alle risposte singole. Essi saranno: Forma, Obbligatoria, Visibile, Lettura, Var, Tipo, Dimensione, Decimali, Var_label, Controllo, Default, Random, Aspetto, Direzione, Maxsel, Modalita_risp e Ncolonne. Differentemente dalla risposta singola, la risposta avrà anche un attributo con il codice dell'help e il valore dell'etichetta da visualizzare, entrambi non obbligatori e gli attributi per il campo "specificare": Spec_testo, Spec_tipo, Spec_nomevar, Spec_control.

```

<!--ELEMENT risposta EMPTY-->
<!--ATTLIST risposta
  Forma (text/memo/multiresponse/vuoto) "text"
  Obbligatoria (si/no) "no"
  Visibile (si/no) "si"
  Lettura (si/no) "no"
  Var CDATA #IMPLIED
  Tipo (Bit/Char/Float/BigInt/Int/SmallInt/DateTime/SmallDateTime/Numeric/
    Decimal/Money/SmallMoney/NVarChar/Real/Text) "Int"
  Dimensione CDATA #IMPLIED
  Decimali CDATA #IMPLIED
  Var_label CDATA #IMPLIED
  Controllo CDATA #IMPLIED
-->

```

Default CDATA #IMPLIED

Etichetta CDATA #IMPLIED

Random (si/no) #IMPLIED

Aspetto (radio/check/dropdown) #IMPLIED

Direzione (orizzontale/verticale) "orizzontale"

Maxsel CDATA #IMPLIED

Modalita_risp CDATA #IMPLIED

Ncolonne CDATA #IMPLIED

Nhelp CDATA #IMPLIED

Spec_testo CDATA #IMPLIED

Spec_tipo CDATA #IMPLIED

Spec_nomevar CDATA #IMPLIED

Spec_control CDATA #IMPLIED

>

Ogni ‘modalità di risposta’, identificata da un proprio ID, avrà una o più ‘opzioni’.

<!ELEMENT modalita_risp (opzione+)>

<!ATTLIST modalita_risp IDmodalita CDATA #REQUIRED>

Ogni ‘opzione’, con un proprio numero e valore, entrambi obbligatori, un attributo selezionato impostato di default a ‘no’, un valore dell’etichetta e un testo, entrambi non obbligatori, è composta anche di Spec_testo, Spec_tipo (che potrà essere di tipo text o memo), Spec_nomevar, e Spec_control, tutti facoltativi e che saranno presenti nel caso di opzioni da specificare.

<!ELEMENT opzione EMPTY>

<!ATTLIST opzione

Numero CDATA #REQUIRED

Valore CDATA #REQUIRED

Selezionato (si/no) "no"

Valore_etichetta CDATA #IMPLIED

Testo CDATA #IMPLIED

Spec_testo CDATA #IMPLIED

Spec_tipo (text/memo) #IMPLIED

Spec_nomevar CDATA #IMPLIED

Spec_control CDATA #IMPLIED

>

L'elemento 'controllo', identificato da un ID, si compone di otto elementi non obbligatori, che possono comparire al massimo una sola volta.

```
<!ELEMENT controllo(car_alfabetici?,car_numerici?,mail?,data?,ora?,
    cod_fiscale?,car_alfanumerici?,formato_personalizzato?)>
```

L'elemento 'car_alfabetici' ha due attributi Maiuscole e Minuscole che assumeranno come valori si o no, con il primo usato come valore predefinito.

```
<!ELEMENT car_alfabetici EMPTY>
```

```
<!ATTLIST car_alfabetici
```

```
    Maiuscole (si/no) "si"
```

```
    Minuscole (si/no) "si"
```

>

L'elemento 'car_numerici' ha come attributi: Numeri impostato di default ad 'interi' e Segno impostato ad 'entrambi', il numero di cifre decimali ed infine, il massimo e il minimo di un eventuale range.

```
<!ELEMENT car_numerici EMPTY>
```

```
<!ATTLIST car_numerici
```

```
    Numeri (interi/reali) "interi"
```

```
    Segno (positivo/negativo/entrambi) "entrambi"
```

```
    Ndecimali CDATA #IMPLIED
```

```
    Max CDATA #IMPLIED
```

```
    Min CDATA #IMPLIED
```

>

L'elemento data è vuoto ma ha un attributo nel quale viene definito il formato che di default è 'gg.mm.aaaa'.

```
<!ELEMENT data EMPTY>
```

```
<!ATTLIST data Formato (gg.mm.aaaa/mm.gg.aaaa) "gg.mm.aaaa">
```

Anche ora è vuoto ma ha un attributo nel quale viene definito il formato che di default è '24'.

```
<!ELEMENT ora EMPTY>
```

```
<!ATTLIST ora Formato (12/24) "24">
```

L'elemento 'codice_fiscale' è caratterizzato dal suo attributo che ne indica la presenza.

```
<!ELEMENT codice_fiscale EMPTY>
```

```
<!ATTLIST codice_fiscale Presente (si/no) "no">
```

Anche l'elemento 'car_alfanumerici', come il precedente, è caratterizzato da un attributo che ne indica la presenza.

```
<!ELEMENT car_alfanumerici EMPTY>
```

```
<!ATTLIST car_alfanumerici Presente (si/no) "no">
```

L'elemento 'formato_personalizzato', infine, avrà un attributo Struttura nel quale sarà specificata la forma del controllo.

```
<!ELEMENT formato_personalizzato EMPTY>
```

```
<!ATTLIST formato_personalizzato Struttura CDATA #IMPLIED>
```

2.2.3 XML Schema:

Purtroppo nel DTD non è concesso definire il tipo di dati degli attributi, né il numero massimo o minimo delle occorrenze di un elemento. Inoltre la sintassi con cui viene scritto non è la stessa di un file XML. Per questi motivi, si introduce un nuovo metodo che prende il nome di XML Schema.

Come per il DTD, XML Schema descrive la grammatica di un file XML, definendone la struttura e il contenuto. Le principali limitazioni del DTD vengono superate da questo nuovo approccio, infatti la sintassi utilizzata è la stessa dell' XML e questo garantisce una grande flessibilità. Per quanto riguarda la definizione del tipo di dati degli attributi, è possibile definirli secondo tipi predefiniti o personalizzati.

Nonostante questi miglioramenti, è stato deciso ugualmente di utilizzare un documento DTD, maggiormente impiegato in rete, in quanto fa parte dello standard XML 1.0, mentre XML Schema è un nuovo standard in fase di approvazione.

2.3 Il Parser:

L'XML, come detto precedentemente, è un metalinguaggio che, se viene utilizzato da solo, rappresenta un semplice contenitore di dati che nella realtà non riesce a trovare riscontro pratico. Per questo motivo, è necessario un supporto da altri programmi, come ad esempio Parser o XML processors. Ciascun questionario è contenuto in un file XML costruito secondo un opportuno schema DTD che ne descrive la struttura. Questo file verrà letto con un Parser, ovvero un software che permette di creare la struttura HTML riconoscendo e trasformando i Tag XML.

Le operazioni che svolgerà il Parser saranno quindi la lettura dei Tag XML e la verifica della loro correttezza, in base a quanto scritto nel file DTD dove viene definita la loro sintassi. In altre parole, il Parser andrà a controllare che il documento XML sia ben formato, cioè che i Tag siano corretti e non contengano errori (come ad esempio la chiusura di un Tag senza che sia stato precedentemente aperto) e valido, vale a dire coerente con la definizione data nel DTD.

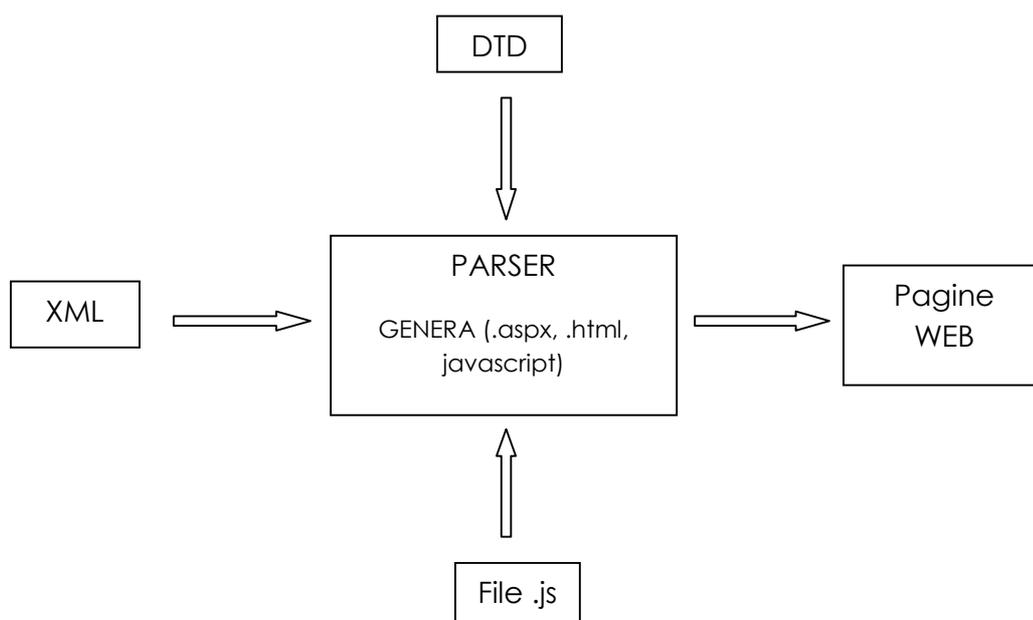


Fig. 3: Elementi che interagiscono con il Parser

Sono ancora in corso massicce operazioni di standardizzazione, ad opera del comitato W3C, riguardanti la manipolazione dell'XML mediante utilizzo delle API (Application Programming Interface). In particolare, questa standardizzazione, ha prodotto le librerie DOM e SAX.

DOM (Document Object Model) è una libreria ad alto livello che permette di trasformare un file XML in un albero ad oggetti con la stessa struttura del documento, che potrà essere modificato o riconvertito in un nuovo file XML, mentre SAX (Simple Api for Xml parsing) è una libreria di basso livello che permette di analizzare sequenzialmente il documento XML generando eventi.

Per il nostro progetto ricorreremo all'utilizzo della libreria DOM che, sfruttando l'annidamento dei tag e la struttura gerarchica dei documenti XML, permette di vedere il documento come un albero e offre la possibilità di navigare tra i nodi che rappresentano i vari elementi.

2.4 Metodi e tecnologie:

Per la gestione dei dati all'interno di database, si è deciso di utilizzare un database centrale dove saranno contenute le informazioni generali di tutte le indagini e tanti database satelliti, uno per ogni indagine, che conterranno le informazioni specifiche di ogni singola indagine.

Tutte le applicazioni sono integrate attraverso moduli adattabili tra di loro e allo stesso tempo espandibili. Avremo quindi tre macro-blocchi che saranno:

- modulo di creazione dell'indagine, del questionario e del database,
- modulo di rilevazione e monitoraggio dell'indagine,
- modulo di esportazione dei dati e datawarehouse.

L'architettura è basata su ASP.Net 2.0⁹ e SQL Server Express 2005 su Windows 2003 Server.

⁹ Il Framework .NET è una tecnologia composta da una struttura di classi che possono essere utilizzate con diversi linguaggi di programmazione *object-oriented* (come ad

Si tratterà quindi di integrare la tecnologia Dot.Net con il metalinguaggio XML, utilizzato come contenitore dei dati per la struttura dei questionari e SQL Server, per il salvataggio dei dati delle risposte.

Aspetti tecnici da tener conto in fase d'analisi, prima dell'implementazione, sono l'utilizzo della tecnica del code-behind per la separazione del codice dalla pagina web, la localizzazione e l'internazionalizzazione, suggerite dal comitato W3C¹⁰, per permettere l'accessibilità dell'intero sistema anche in altre lingue e dell'impiego della divisione dei tre livelli: data layer, business logic e GUI (graphical user interface).

esempio C#, VB.NET, Visual C++ e J#). Il codice compilato, in .NET, prende il nome MSIL (Microsoft Intermediate Language), che verrà poi eseguito tramite una macchina virtuale detta CLR (Common Language Runtime).

¹⁰ Il W3C è stato creato per portare il Web al massimo del suo potenziale, sviluppando protocolli comuni che permettano la sua evoluzione e ne assicurino l'interoperabilità.

Capitolo 3: Descrizione del sistema creato, potenzialità e limiti

Il sistema che si propone, crea in maniera automatica un questionario online (su una o più pagine) a partire da un file XML nel quale è contenuta la sua struttura. In questo capitolo entreremo nello specifico, analizzando come è strutturato il sistema e quali sono le funzioni che lo compongono. Tra queste ci saranno le funzioni che leggono il file XML, salvano i dati (per poi costruire le pagine web), creano il database, salvano le risposte inserite dai vari utenti (dopo aver effettuato la registrazione) e gestiscono la grafica dei questionari.

3.0 Potenzialità del sistema:

Grazie alla sua struttura, questo sistema permette di gestire un gran numero di questionari differenti, sia che essi contengano domande singole che domande sotto forma di tabella.

Per la realizzazione delle tabelle abbiamo creato una struttura composta di un numero variabile di elementi chiamati “blocchi_risposte” (idealmente il numero è illimitato, dobbiamo però, tener presente che dovranno essere visualizzati sullo schermo di un computer e quindi una dimensione di 5 “blocchi_risposte” sembra essere il massimo accettabile per una buona visualizzazione). Ogni “blocco_risposta” può contenere più “risposte”, che devono essere dello stesso tipo (o eventualmente vuote). Nel caso di domande multiresponse le risposte appartenenti ad uno stesso blocco, dovranno avere la stessa modalità di risposta.

Si possono decidere a priori il numero di pagine che dovrà avere il questionario Web. Ogni pagina avrà un *Header* (la prima ne conterrà uno differente dalle successive), una serie di *User Controls*¹¹, chiamati anche controlli, tanti quanti sono le sezioni, nei quali

¹¹ Uno User Control è una porzione di pagina ASP.NET, incapsulata in un componente che può essere inserito come parte di una o più pagine delle nostre applicazioni Web.

saranno presenti le domande del questionario, un bottone per la navigazione e un *Footer* (uguale per tutte le pagine).

Una volta effettuato il login, verrà salvato lo *UserID* dell'utente, che nel nostro caso è il rispondente, in una variabile di sessione. Le risposte date, verranno salvate nel database (in corrispondenza dello *UserID*) ad ogni cambio di pagina, così, nel caso in cui il rispondente si stancasse di rispondere al questionario (prima di arrivare all'ultima pagina) e volesse continuarlo in seguito, al prossimo accesso le risposte che aveva fornito (prima di aver cliccato il bottone *Next*) si visualizzeranno nuovamente nella pagina Web.

3.1 Lettura del file XML e validazione tramite DTD:

Per leggere il file XML, viene creata una variabile di tipo *XmlReader*¹² che supporta la lettura dei dati XML da un flusso o da un file (nel nostro caso useremo quest'ultima) e una variabile di tipo *XmlReaderSettings*¹³ che permetterà di abilitare l'utilizzo del DTD e validare il file XML tramite DTD (verranno inoltre gestite le eccezioni che possono essere generate dalla validazione).

La variabile di tipo *XmlReader* verrà utilizzata nel salvataggio dei dati nelle classi perché in essa sono contenuti i dati XML.

3.1.1 Creazione delle classi:

Per avere a disposizione i dati XML in modo più fruibile è stato deciso di salvarli in apposite classi, che generalmente ne rispecchiano la struttura. In particolare ad ogni elemento corrisponderà una classe e gli attributi verranno salvati in appositi *Hashtable*¹⁴

¹² La classe *XmlReader*, implementata nella libreria *System.XML*, è una classe astratta che fornisce un accesso ai dati XML, di sola lettura.

¹³ Specifica un insieme di funzioni da supportare nell'oggetto *XmlReader* creato dal metodo *Create*.

¹⁴ Una *Hashtable* rappresenta un insieme di coppie chiave/valore.

(alla coppia chiave/valore, corrisponderanno rispettivamente il nome dell'attributo e il suo valore).

Per mantenere la struttura gerarchica che è propria dei file XML, in ogni elemento padre, sarà contenuto un *Arraylist* (uno per ogni elemento figlio di tipo diverso) che conterrà tanti elementi, quanti sono i figli di quel tipo.

Analizziamo ora le classi nello specifico:

Per la classe QUESTIONARIO, è presente un *Hashtable* che conterrà gli attributi di questo elemento (nel nostro caso, l'elemento questionario ha un solo attributo, ma abbiamo deciso di rendere queste classi il più generalizzate possibili, così se volessimo in futuro aggiungere altri elementi, potremmo farlo senza troppi problemi).

La proprietà attributi, nel caso "sia Get" ritorna il valore dell' *Hashtable*, mentre nel caso di Set, imposterà il suo valore a seconda del dato assegnato. In questo ultimo caso, verrà passato un parametro (che conterrà il valore da assegnare alla proprietà) alla routine Set durante le assegnazioni di proprietà. Il valore viene memorizzato in una variabile locale privata e restituito ogni volta che viene chiamata la routine Get.

La classe contiene inoltre i riferimenti ai suoi elementi figlio (Pagine, ModRisposte e Controlli). Sono tutti degli *Hashtable*, che conterranno rispettivamente elementi di tipo Pagina, ModRisposta e Controllo, definiti nelle loro specifiche classi.

```
Public Class QUESTIONARIO
    Dim _attributo As Hashtable

    'attributi di QUESTIONARIO
    Public Property attributi() As Hashtable
        Get
            Return _attributo
        End Get
        Set(ByVal value As Hashtable)
            _attributo = value
        End Set
    End Property

    Public Pagine As New ArrayList()
    Public ModRisposte As New ArrayList()
    Public Controlli As New ArrayList()
End Class
```

Analogamente, per la classe Pagina, saranno salvati gli attributi dell'elemento "Pagina" e un ArrayList di "Sezioni".

```
Public Class Pagina
    Public _attributo As Hashtable

    'attributi di Pagina
    Public Property attributi() As Hashtable
        Get
            Return _attributo
        End Get
        Set(ByVal value As Hashtable)
            _attributo = value
        End Set
    End Property

    Public Sezioni As New ArrayList()
End Class
```

Anche ModRisposta avrà i suoi attributi, salvati in un Hashtable, e Opzioni, un ArrayList di elementi di tipo Opzione.

```
Public Class ModRisposta
    Public _attributo As Hashtable

    'attributi di ModRisposta
    Public Property attributi() As Hashtable
        Get
            Return _attributo
        End Get
        Set(ByVal value As Hashtable)
            _attributo = value
        End Set
    End Property

    Public Opzioni As New ArrayList
End Class
```

Per la classe Opzione, non essendo un elemento padre (già nella definizione data dal DTD si può notare che questo elemento sarà vuoto), sarà contenuto solo l'Hashtable degli attributi.

```
Public Class Opzione
    Public _attributo As Hashtable
```

```
'attributi di Opzione
Public Property attributi() As Hashtable
    Get
        Return _attributo
    End Get
    Set(ByVal value As Hashtable)
        _attributo = value
    End Set
End Property
End Class
```

La classe Sezione conterrà anch'essa i suoi attributi e il riferimento ai suoi elementi figli "Domande".

```
Public Class Sezione

    Public _attributo As Hashtable

    'attributi di Sezione
    Public Property attributi() As Hashtable
        Get
            Return _attributo
        End Get
        Set(ByVal value As Hashtable)
            _attributo = value
        End Set
    End Property

    Public Domande As New ArrayList
End Class
```

La classe Domanda oltre ai suoi attributi avrà il riferimento ai suoi due elementi figli: Risp_singole e Blocchi_risposte.

```
Public Class Domanda
    Public _attributo As Hashtable

    'attributi di Domanda
    Public Property attributi() As Hashtable
        Get
            Return _attributo
        End Get
        Set(ByVal value As Hashtable)
            _attributo = value
        End Set
    End Property
End Class
```

```
End Property
```

```
Public Risp_singole As New ArrayList  
Public Blocchi_risposte As New ArrayList  
End Class
```

La classe Risp_singola, essendo un elemento vuoto, conterrà solo i suoi attributi.

```
Public Class Risp_singola  
Public _attributo As Hashtable  
  
'attributi di Risp_singola  
Public Property attributi() As Hashtable  
Get  
Return _attributo  
End Get  
Set(ByVal value As Hashtable)  
_attributo = value  
End Set  
End Property  
End Class
```

Blocco_risposte, invece, essendo un elemento padre (di Risposta), avrà oltre agli attributi, un ArrayList di elementi di tipo Risposta.

```
Public Class Blocco_risposte  
  
Public _attributo As Hashtable  
  
'attributi di Blocco_risposte  
Public Property attributi() As Hashtable  
Get  
Return _attributo  
End Get  
Set(ByVal value As Hashtable)  
_attributo = value  
End Set  
End Property  
Public Risposte As New ArrayList  
End Class
```

La classe Risposta conterrà solo i suoi elementi.

```
Public Class Risposta  
Public _attributo As Hashtable  
  
'attributi di Risposta  
Public Property attributi() As Hashtable
```

```
Get
    Return _attributo
End Get
Set(ByVal value As Hashtable)
    _attributo = value
End Set
End Property
End Class
```

Infine, per gli elementi di tipo Controllo, si è deciso di procedere in modo leggermente differente. Invece di usare un Hashtable per salvare gli attributi, è stata usata una stringa per salvare l'unico attributo presente, mentre, i suoi figli sono stati accorpati in un unico ArrayList chiamato ElementiControlli.

```
Public Class Controllo
    Dim IDcontrollo As String

    'attributo di Controllo
    Public Property attributi() As String
        Get
            Return IDcontrollo
        End Get
        Set(ByVal value As String)
            IDcontrollo = value
        End Set
    End Property
    Public ElementiControllo As New ArrayList
End Class
```

Nella classe ElementoControllo vengono salvati il nome del controllo (car_numerici, data, ...), il codice identificativo del controllo che lo contiene e gli attributi dell'ElementoControllo.

Le proprietà presenti, fanno in modo di poter 'settare' un valore, o di ottenerlo, per le variabili dichiarate.

```
Public Class ElementoControllo
    Dim IDcontr As String
    Dim NomeContr As String
    Dim _attributo As Hashtable

    'attributi di ElementoControllo
    Public Property IDcontrollo() As String
        Get
            Return IDcontr
        End Get
    End Property
End Class
```

```

Set(ByVal value As String)
    IDcontr = value
End Set
End Property

```

```

Public Property attributi() As Hashtable
    Get
        Return _attributo
    End Get
    Set(ByVal value As Hashtable)
        _attributo = value
    End Set
End Property

```

```

Public Property nome() As String
    Get
        Return NomeContr
    End Get
    Set(ByVal value As String)
        NomeContr = value
    End Set
End Property
End Class

```

3.1.2 Salvataggio dei dati letti dal file XML:

Per il salvataggio dei dati letti dal file XML, sono state utilizzate delle classi (spiegate precedente) che rispecchiano la struttura del documento XML.

È stata creata a tal scopo, un'ulteriore classe, "Lettura.vb", nella quale verranno individuati i vari elementi, con i relativi attributi, e grazie ad una serie di cicli annidati, salvati negli appositi oggetti (che saranno di tipo Questionario, Sezione, ...).

Grazie alla proprietà *XmlNodeType*¹⁵, per ogni nodo all'interno della variabile di lettura, controllo se è di tipo *Element*, se rappresenta cioè un elemento, e lo confronto con gli elementi padre del file XML (a ciascuno dei quali corrisponde una classe).

¹⁵ La proprietà *XmlNodeType*, implementata nella libreria System.XML, ottiene il tipo del nodo corrente.

Per ogni elemento, in base al nome, viene creato un oggetto di quel dato tipo (ad esempio, “pagina”, “sezione”,...), vengono salvate le informazioni riguardanti tutti i suoi attributi e, nel caso in cui sia anche un elemento figlio, verrà salvato a sua volta all'interno dell'ArrayList contenuto nel padre.

Queste operazioni, ci permetteranno in seguito, di poter disporre liberamente dei dati che abbiamo salvato e di poter utilizzare gli oggetti creati.

3.2 Creazione delle pagine Web:

Quando viene richiamata la pagina che crea il Questionario, vengono create le sue pagine con estensione “.aspx”, ognuna delle quali contiene dei controlli “.ascx”, in particolare, uno per l'Header (quello della prima pagina, sarà differente da quello delle pagine successive, in quanto nel primo caso, saranno contenute maggiori informazioni, che potranno essere omesse nelle pagine seguenti), uno per il Footer e dei controlli nei quali saranno visualizzate le domande del questionario (il numero di questi controlli sarà pari al numero di sezioni contenute in quella pagina). Inoltre saranno presenti dei bottoni per la navigazione tra le pagine del questionario. Alla fine del questionario, verrà visualizzata una nuova pagina con i ringraziamenti per averlo completato.

3.2.1 Creazione del contenuto dei controlli:

Ad ogni sezione, corrisponde un controllo, all'interno del quale sono contenute le domande del questionario.

Per visualizzare le domande, è stata utilizzata una procedura *creaDom()* nella quale viene creata una tabella che contiene il titolo, le istruzioni, le note e la risposta di ciascuna domanda. Prima del titolo, verrà visualizzata l'immagine con il collegamento all'help della domanda, se è previsto. Se il valore dell'attributo “Allineamento” è affiancato, allora la struttura sarà del Tipo A, mentre se è sottostante, la struttura sarà del Tipo B in Fig. 4.

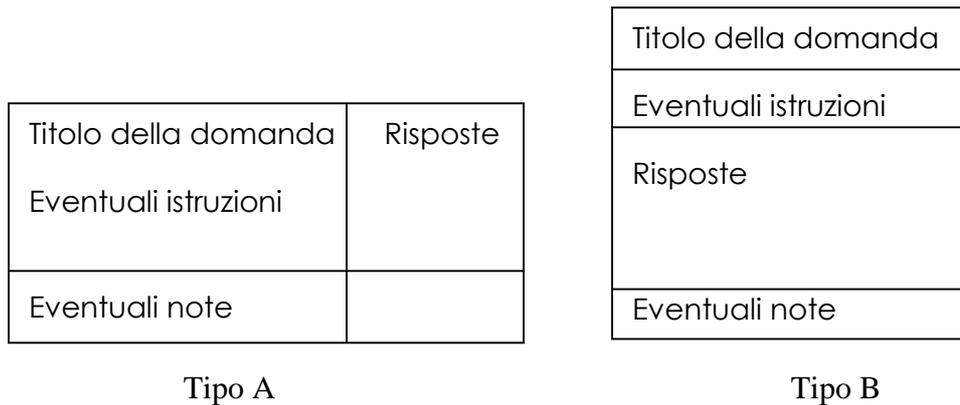


Fig. 4: Tipo A: affiancato, Tipo B: sottostante

All'interno di *creaDom()* vengono richiamate anche altre procedure: per ogni risposta singola, in base al valore dell'attributo "Forma", verranno richiamate *creaVuoto()*, *creaTextBox()*, *creaMulti()*, mentre per i "blocchi_risposte", verrà richiamata la procedura *convalida()*.

creaVuoto(): visualizza una risposta vuota.

creaTextBox(): visualizza una risposta di tipo text o memo. Viene creato un *TextBox* con l'attributo *TextMode* uguale a *MultiLine* se la risposta è di tipo memo e uguale a *SingleLine* se la risposta è di tipo text. Vengono inoltre impostati i valori contenuti negli altri attributi, in particolare: *ID*, al quale viene assegnato il valore dell'attributo "Var", *Visible*, che viene impostato in base all'attributo "Visibile", *ReadOnly*, al quale viene attribuito il valore contenuto in "Lettura", e *Text* al quale viene assegnato il valore contenuto nell'attributo "Default".

creaMulti(): visualizza una risposta di tipo multiresponse. Crea un *RadioButtonList*, un *CheckBoxList* o un *DropDownList* in base al valore dell'attributo "Aspetto". Per i primi due, se l'attributo "Direzione" è orizzontale allora avranno un attributo *RepeatDirection* impostato a *Horizontal*, altrimenti sarà impostato a *Vertical* e, nel caso in cui sia stato inserito il numero di colonne su cui dividere le opzioni, avranno anche *RepeatColumns* uguale al valore dell'attributo "Ncolonne". Per tutti e tre, saranno inserite le opzioni contenute nella modalità di risposta corrispondente e che è indicata nell'attributo ("Modalita_risp"). In questo modo, diverse domande multiresponse, con opzioni uguali,

possono richiamare la stessa modalità di risposta e quindi basterà scrivere le diverse opzioni una sola volta, invece di ripeterle ad ogni domanda.

Inoltre, per i *CheckBoxList*, se presente, verrà salvato il valore del numero massimo di selezioni.

convalida(): controlla che tutti i blocchi di una domanda siano tra loro compatibili, ovvero, che abbiano lo stesso numero di risposte, se non lo sono, sarà necessario dividerli in più blocchi risposta. Se la convalida, invece, ritorna esito positivo, verrà richiamata la procedura *creaTabella()*.

creaTabella(): serve per le domande di tipo tabella. Come spiegato all'inizio del capitolo, una tabella è organizzata in uno o più "blocchi_risposta", ad ognuno dei quali possono appartenere più elementi di tipo "risposta". Ogni "blocco_risposta", può contenere delle etichette, che, se presenti, verranno visualizzate in una colonna posta sulla sinistra, prima delle risposte.

Per ogni "blocco_risposte" se le etichette sono presenti, verrà fatta un'ulteriore colonna (nella quale verranno visualizzate), altrimenti sarà presente solo la colonna delle risposte. Il titolo di ciascun blocco sarà visualizzato sulla colonna delle risposte, così come le istruzioni del blocco. Sarà però possibile, impostando l'attributo "Span", far visualizzare lo stesso titolo o le stesse istruzioni, su più blocchi contigui.

Sotto alle istruzioni, saranno visualizzati i testi delle opzioni nel caso in cui il blocco sia composto di risposte multiresponse (fatta eccezione nel caso di *DropDownList*, perché le opzioni saranno inserite all'interno del menù a tendina). Viene data, inoltre, la possibilità di mandare a capo il testo di un'opzione troppo lunga, inserendo il carattere '\$' che verrà interpretato come
.

Si visualizzeranno a questo punto, tutte le risposte, una per riga. Nel caso in cui si voglia inserire un elemento "specificare" in una risposta della tabella, verrà inserito, nell'etichetta corrispondente, un *TextBox* che consentirà di salvare le informazioni di tale elemento.

Tutti i blocchi saranno disposti uno affianco all'altro, nel caso in cui l' "Ordinamento" sia orizzontale, mentre uno sotto l'altro nel caso di un orientamento verticale¹⁶.

Gli "help" per quanto riguarda le tabelle, potranno essere inseriti nel blocco, in ciascuna sua risposta o in entrambi.

The diagram illustrates the structure of a question in a tabular form. It includes labels for 'Domanda', 'Codice domanda', 'Titolo domanda', 'Blocco_risposte 1', 'Blocco_risposte 2', 'Blocco_risposte 3', 'Istruzioni_blocco', 'Etichette Blocca 1', 'Testo delle opzioni', and 'Risposta vuota'. The main table has columns for 'Consuntivo 2° semestre 2006' and 'Previsioni 1° semestre 2007', with sub-columns for 'Tendenza' and 'Variazione effettiva'.

	Consuntivo 2° semestre 2006			Variazione effettiva	Previsioni 1° semestre 2007			
	Tendenza				Tendenza			
	aumento	stabile	diminuz.		aumento	stabile	diminuz.	non so
Ordinativi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>%	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
di cui: Italia	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extra UE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Produzione	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>%	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fatturato	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>%	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
di cui: Italia	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extra UE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Occupazione	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>%	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ore di lavoro straordinario	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>

Fig. 5: Struttura e relativi elementi di una domanda in forma tabellare.

3.3 Creazione del database:

Per la creazione del database dove salvare i dati delle risposte inserite dai vari utenti, è stato deciso di creare un'unica tabella¹⁷ che può essere meno efficiente (in quanto

¹⁶ L' orientamento verticale, in questa fase, non è ancora stato implementato.

¹⁷ Un'altra soluzione sarebbe stata quella di costruire una tabella (di supporto) a parte per le risposte delle domande dinamiche, inserendo un nuovo record ad ogni riga, e mantenerne un riferimento nella tabella di base.

necessita di maggior tempo per ottenere i dati e reperire le informazioni) ma nel nostro caso è più utile in quanto dobbiamo esportarla in SPSS (se avessimo creato più tabelle, in seguito avremmo dovuto comunque accorparle).

La tabella viene realizzata tramite una funzione che controlla se nel database è già presente una tabella “Questionario” e in tal caso, chiede conferma per l’eliminazione della tabella presente e la creazione di quella nuova.

I campi del database vengono creati automaticamente in base al tipo di risposte previste per le domande del questionario, mentre la chiave primaria sarà sempre lo *UserID* del rispondente.

Per le risposte di tipo “text”, “memo”, “radio” e “dropdown” viene creato un campo, con lo stesso nome della variabile nella quale è contenuta la risposta, di tipo *NVarChar*, nel quale verrà salvato il valore della risposta: nei primi due casi, conterrà il valore del *TextBox*, mentre negli ultimi due verrà salvato il valore dell’opzione che è stata selezionata.

Per le risposte di tipo “check”, ci sono due modi di procedere, che dipendono dalla presenza dell’attributo “Maxsel”, infatti per questo tipo di risposte, si possono selezionare un numero di alternative differenti che vanno da 0 al numero di opzioni presenti e, nel caso in cui sia presente “Maxsel”, da 0 al valore indicato da tale attributo. Se l’attributo non è presente, verranno creati tanti campi di tipo *Bit* (true/false), quanti sono il numero delle opzioni, altrimenti il numero dei campi (che in questo caso saranno di tipo *NVarChar*) sarà uguale al numero massimo di opzioni selezionabili. I nomi dei campi saranno del tipo “nomeVariabile_i” dove *i* rappresenta, nel primo caso, tutti i numeri da 1 al numero di opzioni e nel secondo caso, i numeri da 1 al numero massimo di selezioni.

Nel caso di risposte che prevedano un elemento “Specificare”, verrà creato un campo dove salvarne i dati, con nome: nomeVarRisposta_nomeVarSpecificare.

Tutti i campi, fatta eccezione per l’identificativo dell’utente, ammettono valori nulli, perché nella gestione dell’evento *on_Click* del bottone di navigazione della pagina, viene richiamata una funzione che controlla se l’utente è già presente nel database.

Nel caso in cui, questa funzione dia esito negativo (ovvero l'utente non è ancora presente), per tale utente viene inserito un nuovo record vuoto. Successivamente verrà richiamata un'altra funzione che inserirà nel database tutti i valori dei campi compilati. Se non avessimo previsto, per tutti i campi la possibilità di essere impostati a *null*, sarebbe stata generata un'eccezione.

3.3.1 Salvataggio dei dati nel database:

Innanzitutto, dobbiamo capire chi può entrare e completare il questionario: il rispondente, al quale vengono fornite una username e una password, e il rilevatore, per conto del rispondente, che lo intervisterà e inserirà le sue risposte.

A questo punto, dobbiamo distinguere due casi possibili, ovvero che sia la prima volta che il rispondente (o chi per esso) entra nel questionario o siano le successive *n* volte. Nel primo caso, non essendo ancora presente nel database, dovrò provvedere inserendo la sua username nel campo *UserID* e salvare poi le risposte, nel secondo caso, mi basterà salvare le risposte dell'utente, nel record già presente, in corrispondenza del suo *UserID*.

Come detto precedentemente, nella gestione dell'evento *on_Click* del bottone di navigazione, viene richiamata una funzione che controlla se quell'utente entra per la prima volta nel questionario, se necessario inserisce un record e successivamente salva i dati inseriti nella pagina.

Per salvare i dati contenuti nei vari campi, la funzione *SalvaDomande*, controlla per ciascun elemento, se contiene qualche valore o se le sue opzioni sono selezionate (in base al tipo del controllo a cui ci stiamo riferendo) e ne inserisce il valore nell'apposito campo, creato precedentemente nel database, richiamando a sua volta la funzione *SalvaDati* che, in base allo username, al nome del campo e al valore da inserire, esegue un *Update* nel database.

3.3.2 Inserimento delle risposte già date:

Nel caso in cui il rispondente non stia entrando nel questionario per la prima volta, al momento del caricamento della pagina viene richiamata la funzione *InsertValoriDomande* che inserisce i valori delle risposte, se presenti nel database, che il rispondente aveva dato precedentemente.

Egli sarà così più “ben disposto” a completare le risposte mancanti, rispetto a doverle inserire nuovamente tutte dall’inizio.

Ciò è possibile, grazie alla scelta di salvare man mano le risposte inserite, ad ogni cambio pagina. È chiaro che nel caso in cui il questionario sia composto di un’unica pagina, i due metodi, quello di salvare le risposte alla fine del questionario e quello di salvarle ad ogni pagina, coincidono.

La funzione *InsertValoriDomande*, alla quale bisogna passare lo username e la pagina, controlla, per ciascun elemento o controllo (ad es. *CheckBoxList*, *TextBox*, ...) contenuto nella pagina, se il rispettivo campo del database contiene qualche valore e, in tal caso, lo inserisce nel campo del controllo. In questo modo, il rispondente, all’apertura del questionario e alla visualizzazione delle pagine, troverà le risposte già inserite.

3.4 La grafica:

Ogni elemento ha una sua classe che sarà contenuta nel foglio di stile, richiamato per ogni pagina, così da poter rendere il questionario il più personalizzabile possibile.

Le classi contenute nel foglio di stile saranno:

Per il corpo della pagina e il contenitore:

```
body {margin:0px;padding:0px;font-family: verdana,arial,sans-serif;font-size:
82%;background: #8CBCE0;color:#333;}
div#container{width: 780px;margin: 0 auto;text-align: center;background-
color:#FFF;}
#questionario {margin:10px;padding:0;text-align:left;}
```

Per gli elementi Sezione:

```
.introduzioneSezione { }  
.titoloSezione { }  
.spiegazioniSezione { }  
.headerSezione{ }  
.corpoSezione{ }
```

Per gli elementi di navigazione:

```
.navigazionePagine{ }  
.bottoneNavigazione { }
```

Per le Domande:

```
.tabellaDomanda { }  
.tdTitoloDomanda{ }  
.titoloDomanda { }  
.nhelpDomanda{ }  
.istruzioniDomanda{ }  
.noteDomanda{ }
```

Per i Blocchi_risposte (la parte che avranno in comune):

```
.nhelpBlocco{ }  
.nhelpEtichetta{ }  
.etichettaRisposta { }  
.tdOpzioni { }  
.tdTestoOpzioni { }  
.tabellaTestoOpzioni{ }
```

Per ciascuna tabella, in base al numero di blocchi di cui è composta:

```
.tabellaBlocco1 { }  
.titoloBlocco1 { }  
.istruzioniBlocco1 { }  
.tdOpzioniBlocco1 { }  
.tdEtichetteBlocco1 { }
```

```
.tabellaBlocco2 { }  
.titoloBlocco2 { }  
.istruzioniBlocco2 { }  
.tdOpzioniBlocco2 { }  
.tdEtichetteBlocco2 { }
```

```
.tabellaBlocco3 { }  
.titoloBlocco3 { }  
.istruzioniBlocco3 { }  
.tdOpzioniBlocco3 { }
```

```
.tdEtichetteBlocco3 {}  
.tabellaBlocco4 {}  
.titoloBlocco4 {}  
.istruzioniBlocco4 {}  
.tdOpzioniBlocco4 {}  
.tdEtichetteBlocco4 {}
```

Per gli elementi di input ed altri eventuali elementi HTML:

```
.textbox {border: solid 1px #999}
```

3.5 Limiti del sistema:

Attualmente, nel sistema non sono ancora stati implementati i controlli, le domande dinamiche e i filtri. I controlli di validità, per ora, vengono inseriti manualmente, includendo le funzioni *Javascript* che ci interessano.

Nel caso di domande sotto forma di tabella, bisogna fare attenzione ad inserire, nei blocchi che contengono risposte di tipo “multiresponse”, il codice della modalità di risposta, anche per le risposte vuote, altrimenti, il programma segnalerà un errore di incompatibilità tra le risposte dello stesso blocco. Inoltre, per quanto riguarda l’attributo “Span”, bisogna prestare attenzione a non inserirlo per una sequenza di blocchi con la presenza di etichette per quelli successivi al primo (ovvero, nel caso in cui ci fosse uno span sul titolo, o sulle istruzioni, per il primi tre blocchi, il primo può avere le etichette, mentre il secondo e il terzo no, altrimenti la tabella non viene più visualizzata come si vorrebbe, in quanto bisogna aggiungere una colonna in più per ogni blocco con etichette).

Nei menù a tendina (*DropDownList*), non viene ancora fornita la possibilità di inserire un campo “specificare”.

Anche se sono stati creati in fase di progettazione, non sono ancora stati implementati gli attributi “Random” (che visualizza le opzioni, di risposte di tipo “multiresponse”, in modo casuale) delle risposte e la modalità verticale di “Ordinamento” (che permette di visualizzare nelle tabelle, i “blocchi_risposte” uno sotto l’altro) dei “blocchi_risposte”.

Capitolo 4: Conclusioni propositive (sviluppi futuri)

Il sistema proposto è alla sua prima versione e non è ancora completo in tutte le sue parti.

La struttura XML per ora è scritta manualmente, si vuole, però, creare una procedura guidata che fornisca un aiuto per la creazione del file XML, in modo da rendere meno oneroso il compito di realizzare la struttura e, non di meno, ridurre al minimo il numero degli errori che si possono commettere in fase di scrittura del file.

I controlli di validità sono stati realizzati in fase di progettazione, ma non sono ancora stati implementati. Si vorrebbe creare un metodo con il quale tutti i controlli possano essere implementati automaticamente e inseriti nelle pagine del questionario che li richiamano.

I controlli di coerenza e i filtri non sono ancora stati realizzati, questi infatti sono stati previsti e studiati in fase di analisi, ma non è ancora stato trovato il modo migliore per gestirli (una soluzione possibile, è quella di inserire degli appositi elementi nella struttura XML).

Per le domande dinamiche, è stato deciso di trattarle in modo analogo ai filtri, ovvero, in base alla risposta data alla domanda principale di una domanda dinamica, verrà richiamata una nuova pagina con tante domande quante sono le risposte consentite. Verranno create, quindi, tante pagine, quante sono le risposte ammesse e, in base alla risposta data alla domanda principale, verrà visualizzata la pagina con il numero giusto di domande. Ovviamente, queste domande avranno un limite fisico, in quanto se si presentano troppe domande, l'intervistato si può stancare e si corre il rischio di avere un questionario incompleto.

Per quanto riguarda il tipo di dati dei campi nel Database, attualmente, sono stati impostati tutti come stringhe, per comodità d'esportazione, si potrebbero però creare delle funzioni per salvare i vari dati nei tipi inseriti nel database e realizzare un metodo per consentire una corretta esportazione, dei dati contenuti nel database, in SPSS.

Attualmente non è possibile modificare caratteri e stili dei testi delle etichette e delle domande, se non utilizzando i fogli di stile. Si vorrebbe dare la possibilità di inserire la sottolineatura, il grassetto, il corsivo e modificare la dimensione e il colore dei caratteri, in base a degli attributi forniti nella struttura del questionario o con l'utilizzo di caratteri speciali (o loro sequenze).

Inoltre, per i menù a tendina, non è ancora prevista la visualizzazione di un campo "Specificare", per questo motivo, si vorrebbe creare un metodo che visualizzi o abiliti a fianco del *DropDownList*, un *TextBox* (nel quale si potranno salvare le informazioni del campo "Specificare") solo quando viene selezionata l'opzione "altro" (o una sua opzione equivalente).

Bibliografia:

- ISTAT (1989) Manuale di tecniche di indagine 1. Pianificazione della produzione dei dati, Istat, Roma
- ISTAT (1989) Manuale di tecniche di indagine 2. Il questionario: progettazione, redazione e verifica, Istat, Roma
- Andrea Bosco (2003) Come si costruisce un questionario, Le bussole, Roma
- Luigi Fabbris (2003) Dispense su “Il Questionario”, Padova
- Luigi Fabbris (2000) 1. Il questionario elettronico, Cleup, Padova
- Antonio Giusti (2000) 2. Ingegnerizzazione del processo di produzione dei dati statistici, Cleup, Padova
- Marisa Civardi e Luigi Fabbris (2000) 3. Valutazione della didattica con sistemi computer assisted, Cleup, Padova
- Luigi Fabbris (2001) 4. Captor 2000: Qualità della didattica e sistemi computer assisted, Cleup, Padova
- Piergiorgio Corbetta (2003) La ricerca sociale: metodologia e tecniche 1. I paradigmi di riferimento, Il Mulino, Bologna
- Piergiorgio Corbetta (2003) La ricerca sociale: metodologia e tecniche 2. Le tecniche quantitative, Il Mulino, Bologna
- Piergiorgio Corbetta (2003) La ricerca sociale: metodologia e tecniche 3. Le tecniche qualitative, Il Mulino, Bologna
- J.Martin - B.Tomson (2001) Imparare ASP.NET , tecniche nuove, Milano

Sitografia:

- <http://www.html.it>
- <http://www.asp.net>
- <http://www.xml.html.it>
- <http://samples.gotdotnet.com/quickstart>
- <http://msdn2.microsoft.com>
- <http://www.confirm.com>
- <http://www.2ask.net>
- <http://www.askia.com>
- <http://www.blaise.com>
- <http://www.pulsetr.co.uk>
- <http://www.istat.it/strumenti/metodi/lineeguida>
- <http://www.w3.org>
- <http://www.morpheusweb.it>

Ringraziamenti:

È difficile in poche righe ringraziare tutte le persone che mi hanno aiutata, sostenuta ed incoraggiata in questi anni di scuola e di Università, perché se ho raggiunto questo traguardo, lo devo soprattutto a loro.

Vorrei innanzitutto ringraziare il professor Fabbris, per la sua competenza e per la disponibilità e cortesia dimostratami.

Un secondo ringraziamento va sicuramente allo ‘staff Questlab’ senza il quale non avrei potuto applicare le mie conoscenze teoriche ed ancor di più, imparare nuovi metodi e tecnologie. Vorrei ringraziare i “capi” Fabio, Giuseppe e Nicola per la loro disponibilità ed in particolare Giuseppe per avermi insegnato ad usare queste nuove tecnologie ed avermi aiutata quando il progetto sembrava essere troppo complicato. I miei compagni d’ufficio Silvia, sempre pronta a rispondere con gentilezza e preparazione alle mie domande, e Paolo, mago dell’SPSS (nonché colonna sonora dello stage), per avermi rincuorato quando i risultati tardavano ad arrivare. Non ultima Mirca, che con la sua spontaneità ed allegria, ha contribuito a creare un clima sereno ed amichevole.

Ringrazio ora i miei compagni di avventura, di studio nonché di “scorribande” di questi tre anni, in particolare: Sara, Marina, Alessio, Diego, Martina, Peiman, Davide e Federico con i quali ho condiviso “gioie e dolori” dell’Università e non solo.

Tutti i miei amici, primi “supporter” dal tempo delle superiori (ed alcuni anche da prima).

Non posso non ricordare le persone che forse più di tutti avrebbero voluto vedermi raggiungere questo traguardo, ma che purtroppo non ci sono più: i miei nonni Arnaldo, Angela e Pasquale.

Un ringraziamento speciale a tutti coloro che mi hanno sopportato in questi anni, incoraggiandomi ad andare sempre avanti.

Di certo non posso dimenticare la mia famiglia, in particolare nonna Renata e zio Gino, che sono sempre stati una presenza costante e un baluardo sul quale poter fare sempre affidamento.

Vorrei ringraziare infine i miei genitori, che non mi hanno mai fatto mancare il loro affetto e la loro presenza. Raggiungendo questo traguardo, spero di poter ripagare, almeno in parte, tutti i sacrifici (non solo economici) che hanno fatto, per permettermi di continuare a studiare.