

EXCEL4PS&SP: Un progetto per il supporto al problem solving ed alla programmazione

Nicola Armenise
Istituto Tecnico Commerciale Statale "A. Olivetti" - Lecce
{nicola.armenise@istruzione.it}

Il contributo descrive un progetto software basato su un modello di utilizzo dei fogli elettronici in contesti di "Problem Solving" e "Software Programming", indirizzato sia ad un uso scolastico, in particolare negli anni iniziali dell'istruzione secondaria superiore e in quelli terminali della scuola dell'obbligo, che a percorsi autodidattici. Il grado di automazione raggiunto dal prototipo realizzato consente, con uno sforzo minimo e concentrato sugli aspetti più concettuali e pregnanti, di affrontare la soluzione di un problema ottenendone rapidamente un modello risolutivo Excel e, all'occorrenza, un programma Pascal, generato anch'esso automaticamente. Articolato su diversi fogli all'interno di un unico documento, il modello favorisce la distinzione delle diverse figure / ruoli in gioco negli ambienti di calcolo automatico, risultando quindi un ottimo prelude nei corsi di introduzione alla programmazione.

1. Introduzione

Il prodotto software **EXCEL4PS&SP** (**EXCEL for Problem Solving & "Software Programming**) descritto in questo articolo rappresenta un avanzamento significativo rispetto alla metodologia di base già proposta e descritta in [1] per l'utilizzo dei fogli elettronici in un contesto di risoluzione problemi. Da oltre un decennio il modello viene utilizzato regolarmente dall'autore nella fase iniziale dei corsi di programmazione (triennio ITC ad indirizzo Mercurio), implementato in Excel dagli studenti per le fasi di sviluppo iniziale (individuazione e classificazione dei dati del problema, progettazione dell'interfaccia utente, ...) ed integrato successivamente dalle fasi di realizzazione dell'algoritmo risolutivo in pseudo-codice e della sua codifica in linguaggio Pascal. La validità del suo impianto ha trovato conferma, oltre che nel notevole interesse mostrato dagli alunni, nella recente proposizione, da parte dell'AICA, del progetto di certificazione "**ECDL for Problem Solving**" (**ECDL4PS**), basato sull'uso del foglio elettronico quale strumento da privilegiare nella risoluzione di problemi. Entrambi i progetti ne condividono inoltre diverse caratteristiche e modalità d'uso, troppo spesso trascurate sino ad oggi, perlomeno nella formazione.

Nell'attuale prima versione il software raggiunge l'importante obiettivo pratico di automatizzare quasi completamente la costruzione del modello Excel che, per quanto non complessa dal punto di vista concettuale, risulta piuttosto laboriosa e ripetitiva. Come ulteriore traguardo degno di nota se ne evidenzia la capacità di sintetizzare un programma Pascal, pronto per essere sottomesso ad un compilatore. Non viene garantita la mancanza di errori lessicali, in particolare per quanto riguarda la scelta degli identificatori per i dati fatta dallo studente: più che di un problema tecnico, peraltro facilmente aggirabile imponendo come dominio un sottoinsieme lessicale condiviso da entrambi i linguaggi, si tratta di una scelta progettuale che non vuole sacrificare la valenza trasversale di uno strumento che può trovare applicazione anche al di fuori della programmazione del computer. Più in generale, va considerato che il vincolo primario sarebbe in ogni caso costituito dalle regole sui nomi imposte da Excel che, almeno in linea teorica, potrebbero contraddire quelle del/i linguaggio/i da sintetizzare. Il novizio programmatore potrà naturalmente prestare attenzione nello scegliere identificatori che possano evitargli la necessità di interventi a posteriori sul codice generato dal software.

Dal punto di vista metodologico - didattico, è opportuno comunque che l'uso di EXCEL4PS&SP per i percorsi di introduzione alla programmazione abbia come prerequisito l'assimilazione del modello logico-concettuale sottostante, implementabile manualmente senza

l'ausilio del software. In caso contrario, verrebbero compromessi gli obiettivi formativi che lo hanno ispirato e che sono ritenuti irrinunciabili, come illustrato nell'articolo citato in precedenza.

Nei paragrafi successivi verranno richiamati dapprima alcuni concetti sulla metodologia alla base di EXCEL4PS&SP, quindi verrà presentato il prodotto software e, in ultimo, si delineeranno alcuni possibili sviluppi per l'immediato futuro.

2. La metodologia alla base del progetto Excel4PS&SP

Come già detto, Excel4PS&SP implementa il metodo proposto in [1], la cui peculiarità va ricercata nel ricorso a fogli di lavoro multipli all'interno di un unico documento Excel relativo al problema da risolvere, operato secondo criteri tendenti a favorire la distinzione delle diverse figure / ruoli in gioco, e specificatamente:

1. **Esecutore** (uomo o computer);
2. **Utente** (uomo);
3. **Programmatore** (uomo).

Per delineare un quadro concettuale di riferimento che possa orientare lo studente, un progetto per la risoluzione di problema sarà articolato di conseguenza sui seguenti fogli:

- a) foglio "Costanti" (Const), per contenere i dati immutabili in relazione alla situazione problematica da affrontare, o che perlomeno si stima rimangano stabili per periodi sufficientemente lunghi;
- b) foglio "Input", per permettere l'inserimento dei dati in ingresso da parte dell'utente durante la fase di esecuzione;
- c) foglio "Variabili" (Var), per contenere tutti i dati, ad eccezione di quelli costanti, necessari per l'elaborazione (dati in ingresso/uscita e di lavoro): in esso verranno dunque memorizzati anche i dati in ingresso prelevati dal foglio Input e quelli in uscita, destinati al foglio Output;
- d) foglio "Elaborazione" o "Programma", in cui dovranno essere incluse tutte le istruzioni necessarie per elaborare i dati (costanti e variabili, prelevati dai relativi fogli) al fine di ottenere i risultati richiesti;
- e) foglio "Output", per visualizzare quei dati che costituiscono i risultati attesi dall'utente, prelevati dal foglio Variabili. E' naturalmente possibile fornire in uscita anche un dato ricevuto in ingresso, ma prelevandolo comunque dall'apposito contenitore sul foglio delle variabili.

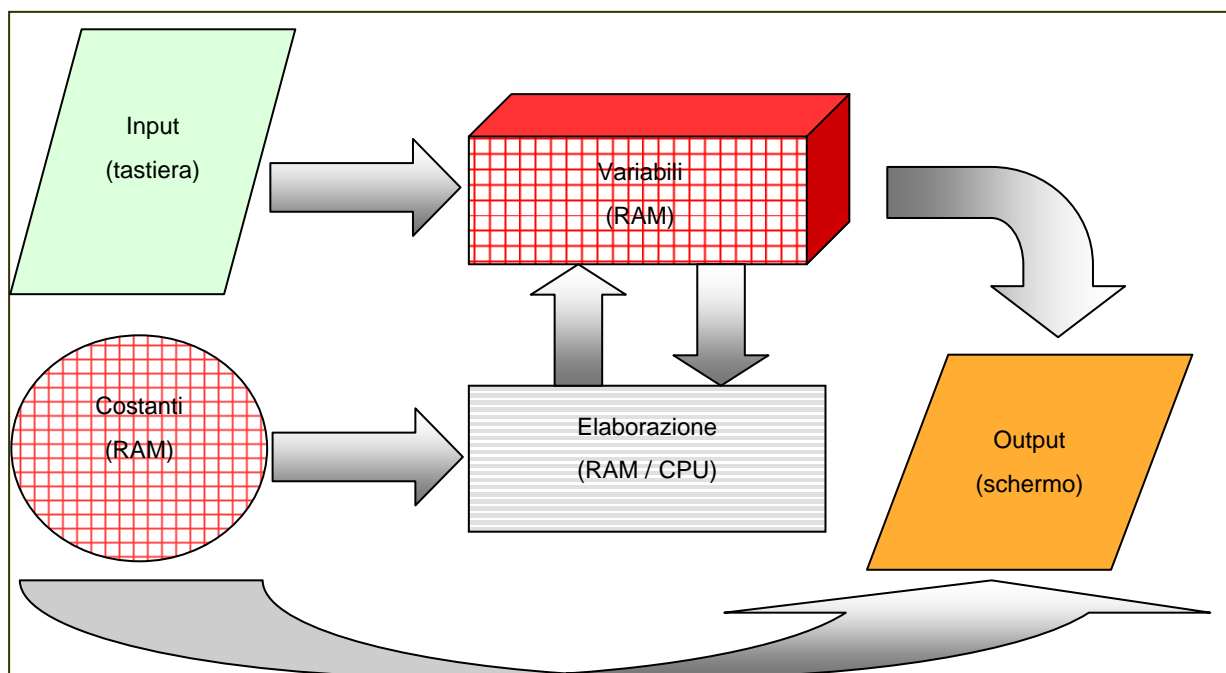


Figura 1 Diagramma di flusso dei dati tra i fogli Excel

In Figura 1 vengono schematizzati i flussi di dati tra i vari fogli, da cui emerge in particolare che i fogli Input / Output hanno la sola funzione di permettere l'inserimento dati e la visualizzazione dei risultati, emulando dunque la tastiera e lo schermo di un PC. Proseguendo nell'analogia, i fogli Costanti e Variabili, contenendo soltanto i dati utilizzati nell'algoritmo risolutivo, possono essere assimilati funzionalmente alla RAM (zona dati). Il foglio Elaborazione, contenendo le istruzioni ed i risultati prodotti, emula RAM (zona programma) e CPU (registri).

Utilizzando il PC si dovrà dunque creare un nuovo documento col software a disposizione (Excel o altro spreadsheet), ed in esso predisporre cinque nuovi fogli, nell'ordine e con i nomi di cui sopra (le alternative tra parentesi fanno riferimento alle intestazioni delle relative sotto-sezioni dichiarative di un programma Pascal). Durante la fase di progettazione / test, che potrà trovare nel software un valido strumento di documentazione, lo studente sarà chiamato a riflettere sul testo della situazione problematica al fine dell'individuazione e classificazione preliminare dei dati in gioco (costanti, input, output), da collocare sugli appositi fogli di lavoro.

Nel foglio Variabili, verranno progressivamente riportati gli stessi dati di I/O, cui verranno aggiunti, man mano che si procede nella progettazione, tutti i dati di lavoro necessari.

Nel foglio Programma devono essere elencati in sequenza i passi di elaborazione sotto forma di formule che permettono di risolvere il problema, contenuti in apposite celle-istruzioni. Ogni cella-istruzione su questo foglio dovrà permettere di assegnare un contenuto ad una variabile. Ora, avendo sistemato tutte le variabili nell'apposito foglio, ne scaturisce che il contenuto relativo ad un dato di lavoro potrà e dovrà essere prelevato dalla cella-istruzione che ne produce il valore.

La fase di collaudo dovrebbe essere condotta dallo studente inserendo i dati in ingresso nelle apposite celle-dato sul foglio di Input e, quindi, osservando ed interpretando i risultati su quello di Output. Se nell'implementazione del progetto è stata rispettata la metodologia illustrata nelle pagine precedenti non sarà necessario, né opportuno, andare ad apportare modifiche sugli altri fogli, se non per eventuali correzioni agli errori (debug) evidenziati dal test. Terminato il collaudo, si procederà nascondendo tutti i fogli che non riguardano l'utente, lasciando visibili solo quelli d'interfaccia (Input e Output). In tal modo si rafforzerà ulteriormente nello studente la distinzione dei ruoli, così come il quadro concettuale relativo ad un sistema di elaborazione.

3. Analisi dei dati del problema

La progettazione dei fogli secondo il modello esposto non presenta particolari problemi, una volta compresa la logica alla base dei flussi di dati in transito tra di essi (implementata ovviamente attraverso un uso sapiente delle formule e dei nomi da assegnare alle celle) e, soprattutto, che sia stato messo a fuoco il problema, in particolare per quanto riguarda i dati. A tal scopo è quanto mai opportuno effettuare un'analisi propedeutica da documentare in modo appropriato, che potrà permettere in aggiunta un elevato grado di automatismo nell'ottenimento del progetto Excel finale.

Andiamo a descrivere questo processo di analisi iniziale dei dati del problema, scomponendolo in 2 sotto-fasi ed analizzando i rispettivi formalismi di documentazione sottostanti.

fase a) Diagramma dei dati in Input/Output

L'importanza di questa fase, collegata in modo stretto alle richieste poste dall'utente committente (o dal problema assegnato, in un ambiente didattico), è naturalmente estrema, al punto da poter pregiudicare la stessa accettazione del software realizzato. Uno degli errori tipici commessi dai principianti è quello di considerare "costanti" tutti i numeri e i nomi presenti nella traccia o definizione del problema da risolvere: ciò porta ovviamente a progettare algoritmi poco flessibili, utilizzabili solo in situazioni particolari, a meno di non essere disposti (programmatori e/o utenti) a cambiare ripetutamente i valori assunti. Questo aspetto legato al trattamento delle pseudo - costanti di un problema dovrebbe esser tenuto in debito conto nella progettazione di prove di valutazione per un sistema / componente di e-learning, insistendo opportunamente sui criteri-guida da adottare per risolvere il dualismo presente, spesso non a caso, tra dato costante e dato in ingresso.

Il formalismo utilizzato nel contesto didattico in esame è un semplice diagramma di flusso dei dati in entrata/uscita, limitato ai dispositivi standard (Input da tastiera, Output su

schermo), che dunque non sono graficamente raffigurati. Come schematizzato in Figura 2, il rettangolo centrale corrisponde al problema/programma, di cui ne contiene un titolo sintetico, nel/dal quale entrano/escono frecce etichettate, una per ogni singolo dato in I/O.

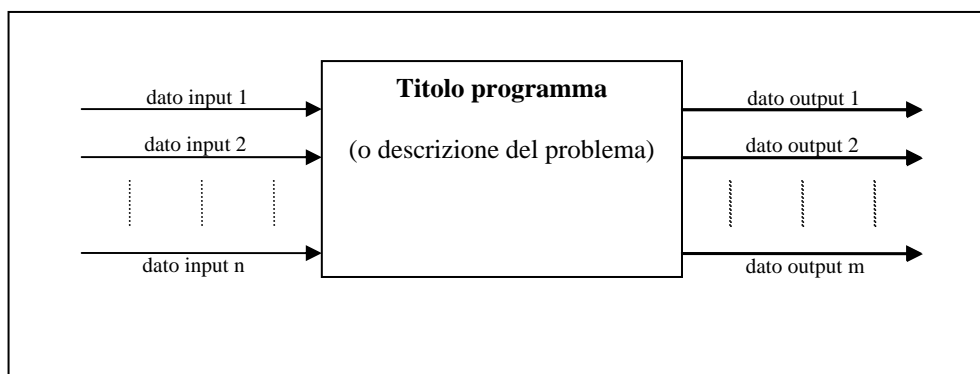


Figura 2 Diagramma di flusso dei dati in Input/Output

fase b) Tabella dei dati

Il cuore del lavoro di progettazione alla base di questo metodo di introduzione alla programmazione è costituito dal prodotto di questa 2a sottofase di analisi dei dati del problema/programma, che può anche giovare in qualche modo dell'uso di un DBMS come Microsoft Access™.

Anche per questa attività è in agguato un errore (metodologico) abbastanza tipico da parte di molti neofiti, per la verità indotto spesso anche da alcuni testi scolastici in uso. Si tratta del tentativo di completare la tabella con tutte le variabili da utilizzare nell'algoritmo, comprese quelle di lavoro, cosa del tutto innaturale in questa fase iniziale, prima ancora di aver cominciato a lavorare sul procedimento risolutivo! E' opportuno invece procedere ad un completamento graduale della tabella dati, inserendovi subito i dati desumibili dal diagramma di I/O ed eventuali costanti del problema, ed aggiungendo solo successivamente le variabili di lavoro man mano che vengono introdotte. Sarebbe bene che nei libri di testo si evidenziasse chiaramente che le soluzioni ai problemi svolti non riflettono necessariamente un ordine cronologico delle attività, ma il prodotto finale di un lavoro di progettazione sovente sommerso. Da questo punto di vista, i numerosi listati che riempiono le pagine di molti testi sulla programmazione rischiano di avere effetti tutt'altro che positivi su uno studente che non sia già addentrato in materia, e non solo sotto il profilo economico!

4. Prototipo Excel e implementazione del programma

In Figura 3 viene rappresentato graficamente il percorso a tappe che permette di raggiungere l'obiettivo (strato più esterno), ossia un programma codificato in un linguaggio di programmazione (Pascal e, come prossima scelta, MicroSoft SmallBasic™), a partire dall'analisi iniziale dei dati in I/O (strato più interno). Gli strati intermedi prevedono la costruzione incrementale della tabella dati e del prototipo implementato attraverso il foglio elettronico. Via via che ci si avvicina all'obiettivo le fasi diventano meno critiche, le decisioni da prendere meno importanti: gli errori iniziali, come ben noto, hanno le conseguenze più gravi. Gran parte del lavoro successivo all'analisi iniziale può essere automatizzata, producendo i fogli const, Input, var (parzialmente) e Output del prototipo, le sezioni iniziali dichiarative del programma e, limitatamente all'I/O, anche quelle esecutive: nell'ottica del monitoraggio automatico dello studente in un contesto di e-learning, questo fattore è garanzia di migliori possibilità d'intervento.

Nel foglio "elaborazione" va inserita la sequenza di passi per ottenere i risultati attesi, sotto forma di formule che, sebbene in un foglio di calcolo possano essere disposte secondo un ordine non predefinito, andranno qui sistemate in base alla logica sequenziale del procedimento risolutivo. Ciò renderà agevole la successiva fase di produzione del programma, sia essa

condotta dallo studente, il quale dovrà semplicemente tradurre ogni formula in un comando di assegnazione, sia che venga effettuata automaticamente.

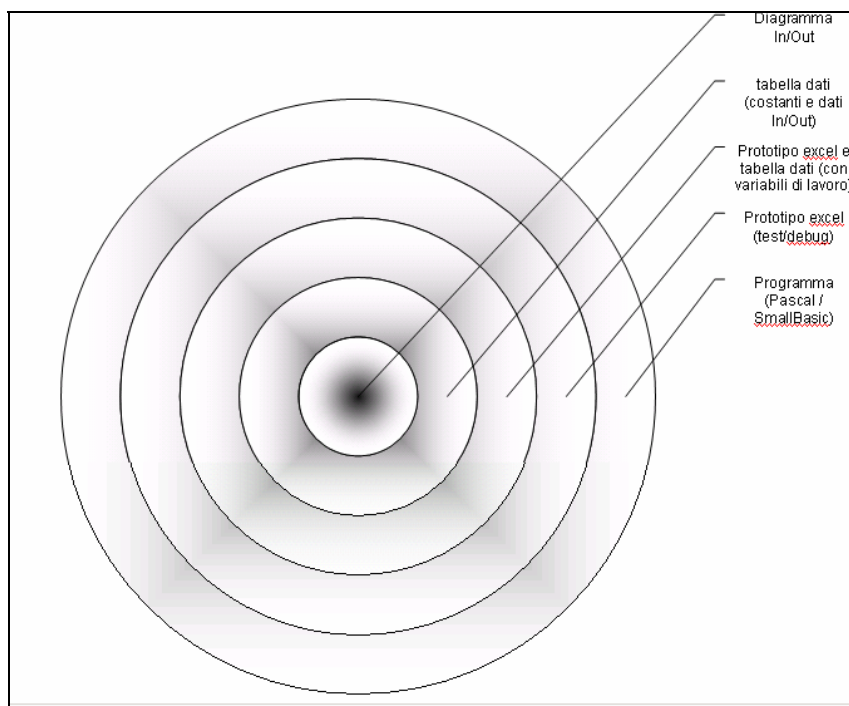


Figura 3 Dal problema al programma, passando per il prototipo Excel

Esperienze del metodo in classi reali hanno confermato la semplicità del compito di traduzione del prototipo Excel in Pascal da parte degli studenti, perlomeno di quelli motivati verso la disciplina informatica.

5. Interfaccia e funzionalità di Excel4PS&SP

L'applicazione è strutturata su diversi fogli, alcuni dei quali propri del modello risolutivo su cui poggia Excel4PS&SP e già descritti in precedenza, ed altri introdotti a scopi documentativi ma, soprattutto, per implementarne gli automatismi che la caratterizzano. In dipendenza della figura / ruolo che di volta in volta si intende assumere, o anche della funzionalità desiderata, solo un sottoinsieme dei fogli saranno d'interesse: è stato incorporato per questo un meccanismo di raggruppamento degli stessi in "viste", selezionabili dal pannello principale dell'applicazione (Figura 4), che permette anche l'inserimento della traccia del problema da risolvere. Sarà possibile la selezione di una delle viste seguenti:

1. **Problem Solving**, il punto di vista di chi è interessato alla progettazione di procedimenti per la risoluzione di problemi, ma non agli aspetti legati alla loro programmazione;
2. **Utente**, da selezionare quando interessa assumere il ruolo di un utente finale, interessato alla effettiva risoluzione di un problema ma non al procedimento sottostante;
3. **Programmatore**, la vista preferenziale nel caso in cui il fuoco sia sulla progettazione di algoritmi risolutivi, da codificare eventualmente in un linguaggio di programmazione;
4. **Test**, una vista composita (programmatore + esecutore + utente) idonea per la fase di collaudo di un algoritmo codificato in precedenza (tramite la vista "programmatore"), grazie alla presenza di un apposito foglio per la tabella di traccia dell'esecuzione;
5. **Programmatore Pascal**, grazie alla quale è possibile richiedere la generazione automatica del codice in linguaggio Pascal;
6. **ECDL4PS** (in corso di implementazione), utile per far convergere il progetto Excel4PS&SP verso il percorso di certificazione per il Problem Solving proposto da AICA;

7. **Authoring**, per una veduta d'insieme di tutti i fogli presenti nell'applicazione (attivata in Figura 4), riservata all'uso da parte di figure abilitate alla gestione, sviluppo e manutenzione del software.

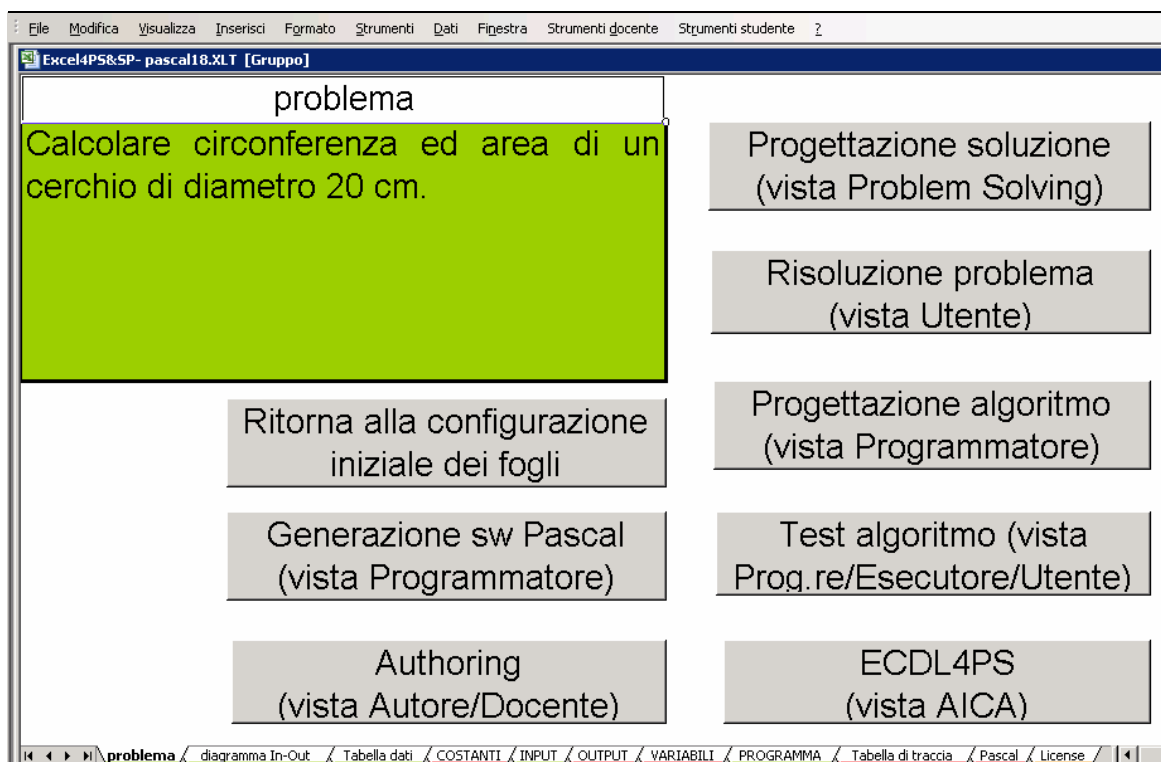


Figura 4 Interfaccia di Excel4PS&SP: le viste

La struttura portante dell'intera applicazione è la tabella progettata nella fase b) dell'analisi iniziale dei dati del problema (Figura 5), incapsulata in un apposito foglio. La sua compilazione, che avviene riga per riga ma può essere sospesa e ripresa a proprio piacimento, viene monitorata dal software, che effettua anche alcuni controlli sul tipo di dato (ultima colonna), utili al fine della eventuale generazione successiva del programma Pascal, oltreché per la congruenza della documentazione. La modifica delle informazioni immesse è possibile in qualsiasi momento sulle aree della tabella opportunamente evidenziate con sfondo verde, inibite sulle altre (in colore rosso). La registrazione effettiva di un dato nel sistema avviene a seguito dell'attivazione di uno dei pulsanti disposti sotto la tabella, che abilitano nell'ordine le funzioni di inserimento di costanti, dati in Input, dati in Output e dati di lavoro.

The screenshot shows the 'Tabella dati' sheet in the Excel4PS&SP application. The table has columns for 'id', 'description', 'var', 'const', 'input', 'output', 'work', 'alfanumerico', 'intero', 'reale', 'value', and 'type'. The 'value' column contains the value 3,14 for the 'pi greco' row. Below the table, there are four buttons: "inserisci costante", "inserisci dato Input", "inserisci dato Output", and "inserisci dato lavoro".

id	description	var	const	input	output	work	alfanumerico	intero	reale	value	type
raggio	misura in cm del raggio	x						x	x		tipo non definito più di un tipo definito
area	area del cerchio in cm	x			x				x		real
circonferenza	misura della circonferenza in cm	x			x				x		real
diametro	diametro della circonferenza	x		x					x		real
pi greco	valore di pi greco		x						x	3,14	real

Figura 5 Tabella dei dati del problema, con vista Problem Solving abilitata

L'inserimento del procedimento risolutivo avviene mediante l'immissione guidata delle formule all'interno del foglio "programma" (Figura 6), nelle celle anche qui evidenziate dal colore verde dello sfondo. Si noti che il software propone automaticamente le righe che necessitano di istruzioni per il calcolo dei dati descritti nell'apposita colonna di commento, sulla base delle informazioni immesse nella tabella dati. Con riferimento all'esempio in figura, Excel4PS&SP richiede formule per il calcolo dei dati in Output (area e circonferenza) e del raggio, dato di lavoro la cui necessità/opportunità potrà emergere in fase di problem solving quando lo studente, costretto ad inserire l'istruzione per il calcolo dell'area, sarà invogliato a tornare sulla tabella dati per inserire una ulteriore riga per il dato di lavoro mancante. Nella composizione delle formule sarà opportuno ricorrere alla scelta degli identificatori con l'apposita funzione "incolla nomi", eliminando alla radice la possibilità, generalmente deleteria, di far riferimento alle celle con le tradizionali coordinate di riga e colonna: lo studente dovrà essere educato all'uso dei soli dati presenti in tabella (fatta eccezione per i casi di costanti poco significative), e nelle prossime versioni il software forzerà quanto più possibile questo modo di operare nel contesto della redazione del foglio di elaborazione.

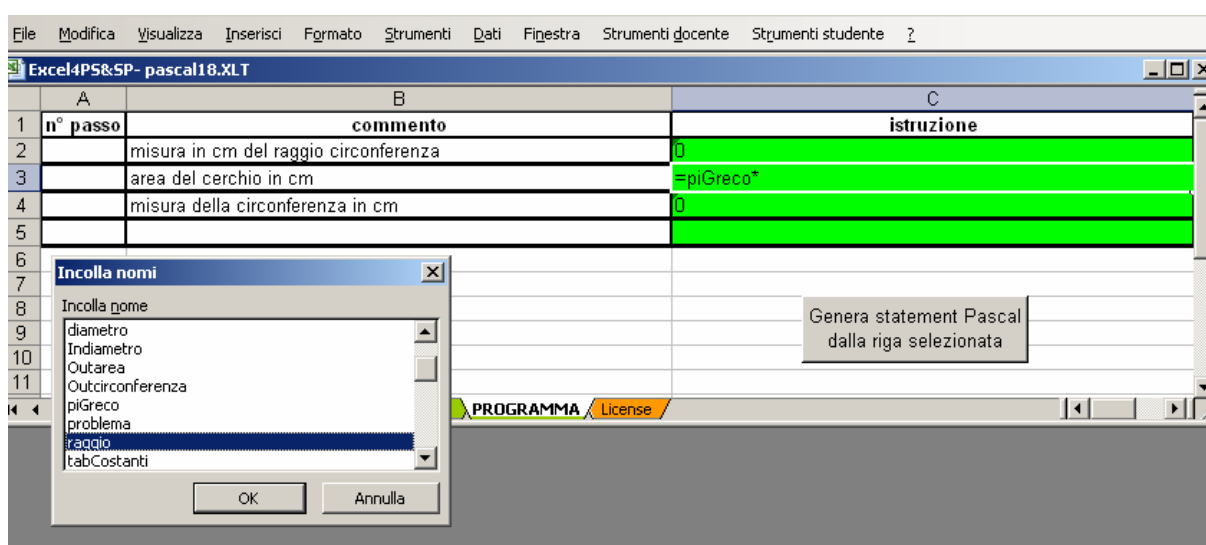


Figura 6 inserimento delle formule nel foglio programma

Terminata la fase di immissione del procedimento risolutivo nel foglio programma, si potrà passare al test dell'algoritmo, attivando l'apposita vista dal pannello principale dell'applicazione, che lascerà visibili i soli fogli dell'interfaccia con l'utente: problema, Input ed Output (Figura 7). Assumendo il ruolo di utente, lo studente dovrà introdurre i dati iniziali sul foglio di Input (nell'esempio, il diametro) vedendo quindi comparire i risultati corrispondenti su quello di Output: il test potrà essere poi ripetuto grazie agli appositi pulsanti disposti in questi fogli di interfaccia.

In ultimo, attivando la vista Pascal, sarà possibile attivare la generazione e visualizzazione del programma Pascal, riportato di seguito per l'esempio del cerchio (vedi Tabella 1).

6. Linee guida per la convergenza di Excel4PS&SP con ECDL4PS

L'implementazione della vista Excel4PS per l'integrazione del modello di Problem Solving proposto da AICA non dovrebbe porre particolari problemi, tenuto conto delle similitudini di base che accomunano i 2 modelli.

In particolare, per ogni domanda di esempio presente nell'apposita sezione riservata del sito AICA dedicata alla certificazione ECDL4PS [2] sono presenti la descrizione dell'esercizio, il foglio di lavoro per lo studente e la soluzione per verifica. L'esercizio viene descritto attraverso una sezione iniziale di illustrazione del contesto, che può essere naturalmente inglobata nel

foglio "problema" di EXCEL4PS&SP, un'altra centrale che ne elenca le richieste, ed una finale per esplicitare i risultati richiesti, sostanzialmente coincidente con il foglio di output. Dall'analisi di alcune delle sezioni descrittive delle richieste si evince soprattutto la loro funzione esplicativa in relazione agli obiettivi finali (Output), spesso accompagnati da obiettivi intermedi necessari per il loro raggiungimento, facilmente associabili a variabili di lavoro e dunque gestibili anch'essi attraverso il completamento della tabella dati (vedi Figura 5).

```

Program cerchio
Const
  piGreco = 3,14 (* valore di pi greco *)
Var
  raggio : più di un tipo definito (* misura in cm del raggio circonferenza *)
  area : real (* area del cerchio in cm *)
  circonferenza : real (* misura della circonferenza in cm *)
  diametro : real (* diametro della circonferenza *)
Begin(* main *)
(* Input *)
  read (diametro)
(* Elaborazione *)
  raggio := diametro / 2
  area := piGreco * raggio * raggio
  circonferenza :=diametro*piGreco
(* Output *)
  write (area)
  write (circonferenza)
End. (* main *)
    
```

Tabella 1 Programma Pascal generato da EXCEL4PS&SP

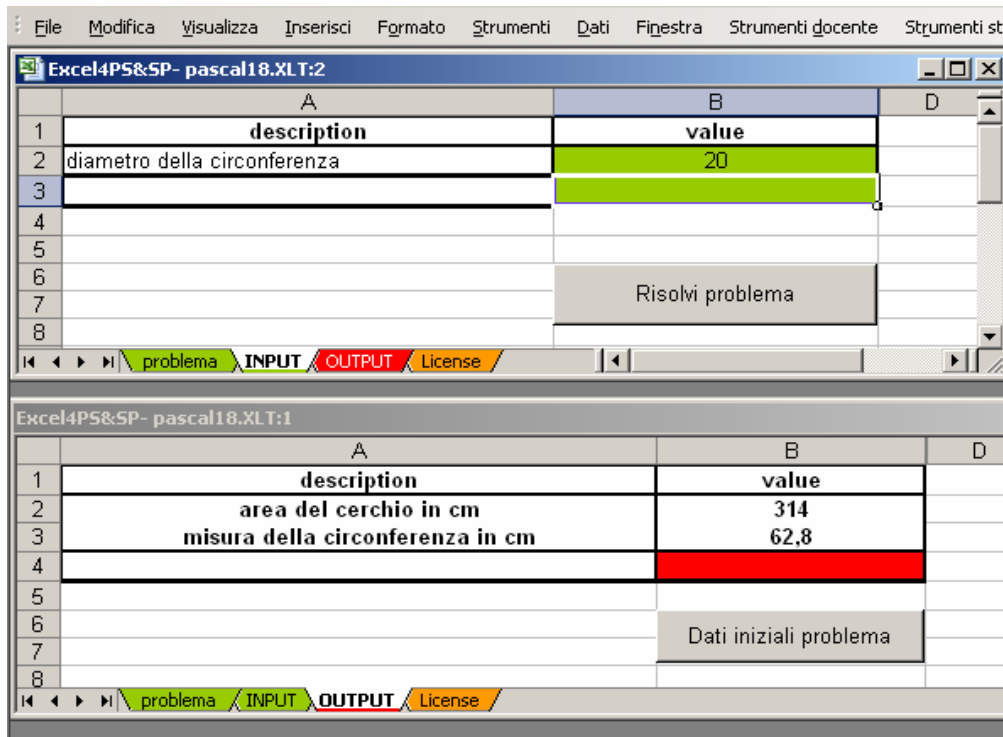


Figura 7 Vista di test dell'algorithm

Sempre nella sezione delle richieste vengono spesso forniti dei chiarimenti sulle modalità di calcolo degli obiettivi (intermedi e finali), che possono confluire in modo assolutamente convergente nel foglio Programma, codificati in formule.

Il foglio di lavoro elettronico proposto da ECDL4PS, pur utilizzabile liberamente dallo studente, deve contenere alcune parti standard non modificabili per l'inserimento di dati, formule e risultati, corrispondenti ai fogli Costanti, Input, Programma e Output di ECDL4PS&SP. A differenza di quest'ultimo, tuttavia, non viene imposta la loro ripartizione su più fogli: al contrario, negli esempi viene proposto l'uso di diverse aree su un unico foglio, sebbene sia difficile ipotizzare un vincolo in tal senso. In ogni caso, l'approccio alla base del software qui presentato non può prescindere, per i motivi già esposti in precedenza, dalla strutturazione del modello analizzata in queste pagine. Considerazioni analoghe possono esser fatte anche per il foglio contenente la soluzione per la verifica.

In conclusione, dovrebbe essere evidente che il modello ECDL4PS può facilmente integrato in una vista del software ECDL4PS&SP, in buona sostanza assimilabile a quella denominata "Problem Solving". Ciò potrebbe avere un impatto molto positivo sullo studente dal punto di vista formativo, che rimane quello primario di questo progetto.

7. Conclusioni e lavoro futuro

In questo contributo si è presentato il software Excel4PS&SP per l'utilizzo del foglio di calcolo Excel nella risoluzione di problemi e nei percorsi introduttivi alla programmazione del computer. Allo stato attuale si tratta di una versione che, per quanto possa essere considerata ancora un prototipo, risulta pienamente utilizzabile e funzionale agli obiettivi prefissati. Come sviluppo futuro, sarà opportuno da un lato investigare le possibilità di potenziamento del modello risolutivo alla base del prodotto, in primo luogo con l'implementazione delle strutture primitive di controllo della programmazione (IF e REPEAT). Dall'altro, potenziare il supporto allo studente nella ricerca e sequenzializzazione del procedimento risolutivo del problema. In ogni caso, lo strumento non è stato progettato per percorsi di programmazione di livello medio - avanzato, nella convinzione che l'acquisizione di basi e metodologie valide nelle fasi iniziali possano poi garantire facilmente progressi significativi nella disciplina.

Bibliografia

[1] Armenise, N., 2001. Un modello per l'utilizzo del foglio di calcolo elettronico nella risoluzione di problemi. Atti Congresso DIDAMATICA, Bari 3-4-5 maggio 2001 - Informatica per la didattica – ed. G. Laterza

[2] ECDL for Problem Solving, url: <http://www.aicanet.net/certificazioni/ecdl/ecdl4ps/accessoriservato>