



Component opdracht: "Sudoku"

Inleiding

Je gaat een sudoku-spel maken EN met een ActiveX-component werken. We hebben een ActiveX-component 'gevonden' waarmee je een Sudoku-spel kunt maken te weten *Sudoku.dll* (zie bijgevoegd ZIP-bestand). De bedoeling is dat je dit bestand gebruikt in je Sudoku-spel. We hebben de volgende methods gevonden in de component (met veronderstelde functionaliteit):

Method	Bedoeling
<code>create()</code>	Maakt een nieuw sudoku spel
<code>set(x, y, value, canAdapt)</code>	Zet waarde <code>value</code> op positie (x,y)
<code>get(x, y, value)</code>	Welke waarde staat op positie (x,y) ?
<code>isValid(isValid)</code>	Zijn de ingevulde waarden correct ???
<code>read(canRead)</code>	Lees waarden uit bestand
<code>write(canWrite)</code>	Schrijf waarden naar bestand
<code>hint(hintPossible, x, y, value)</code>	Geef een hint

Zoals we al aangaven, veronderstellen we dat dit de functionaliteit is. Aan jullie om te onderzoeken of dit wel klopt.

Opdracht

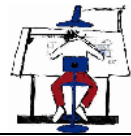
Maak een sudoku –spel met de ActiveX –component *Sudoku.dll*.

Enkele aanwijzingen/eisen:

1. Maak een drietal dropboxen met een knop waarmee je de waarden kunt invoeren.
2. Schrijf de puzzel (de cijfers op het canvas). In het andere geval heb je $9 \times 9 = 81$ velden die je steeds moet controleren.
3. Het spel moet een cheat mode hebben, waarbij de hele puzzel wordt afgemaakt op 2 velden na (daarmee is het afmaken van de puzzel testbaar)
4. Alle bovengenoemde methodes moeten worden geïmplementeerd. (als je niet alles uitprogrammeert kom je ook niet alle problemen met het werken met componenten tegen).
5. De userinterface moet duidelijk 9 keer 3x3 zijn, met ruimte tussen de blokken.

Hints

- Sudoku wordt goed uitgelegd in de wikipedia ("<http://nl.wikipedia.org/wiki/Sudoku>").
- Het toevoegen van een ActiveX –component in je C# -project wordt uitgelegd in de MSDN ("[http://msdn2.microsoft.com/en-us/library/aa194131\(office.11\).aspx](http://msdn2.microsoft.com/en-us/library/aa194131(office.11).aspx)").
- Voordat je de ActiveX –component gaat gebruiken, dien je 'm eerst te registreren. Voer voor dat doel het BAT-bestand *register.bat* uit.
- Het derde bestand in het ZIP –bestand is *Sudoku.tlb*. Dit is een type library. Daarmee kun je uitzoeken wat een ActiveX–component allemaal kan zonder de component te laden. (Probeer 'm maar 'ns te openen in Visual Studio...)



Codevoorbeeld

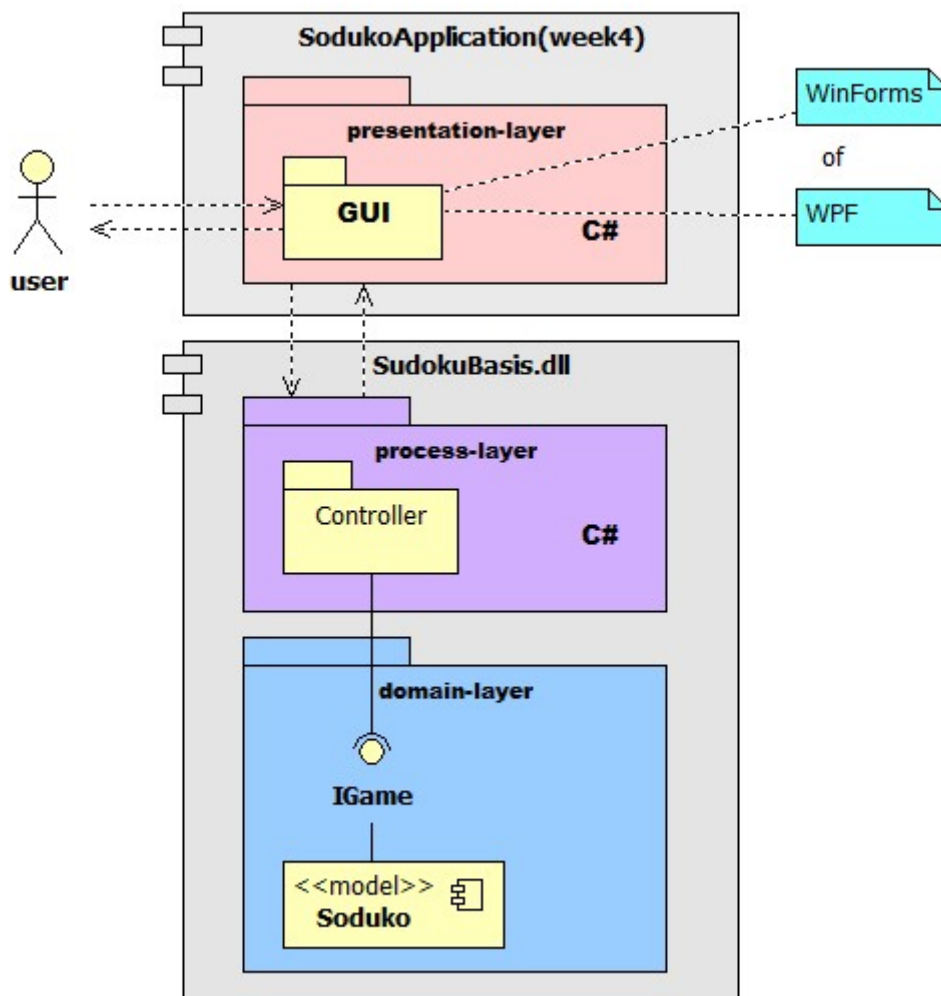
In de volgende code wordt een sudoku –veld gecreëerd en weggeschreven:

```
Sudoku.IGame myGame = new Sudoku.Game();  
int canWrite;  
myGame.create();  
myGame.write(out canWrite);
```

Architectuur:

De opdracht strekt zich uit over week 4/5/6. Je moet een architectuur maken zodat verschillende applicaties maakbaar zijn met gebruik van een herbruikbare basis. In die basis moet het component "Sudoku.dll" opgenomen zijn met een schil erom, hieronder afgebeeld als Controller, waarbinnen de toegang tot het component wordt geregeld. Deze basis moet op zichzelf een dll worden die door verschillende applicaties wordt gebruikt. Hieronder zie je de applicatie van week 4. De GUI mag gemaakt zijn in Winforms of WPF.

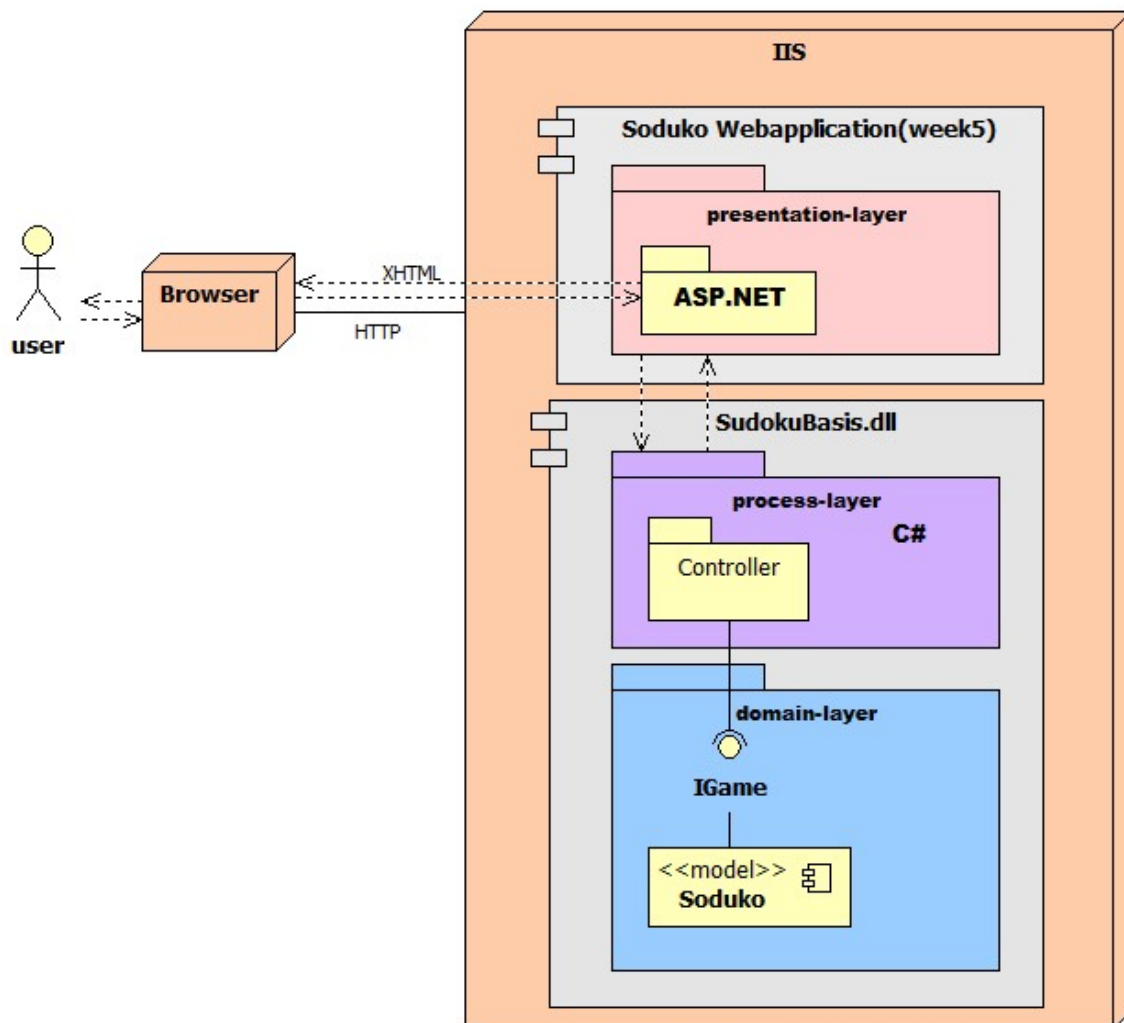
Week4





In week 5 blijft de basis gelijk, maar wordt de userinterface vervangen door een ASP.NET laag. De users is dan niet meer local, maar remote op het web.

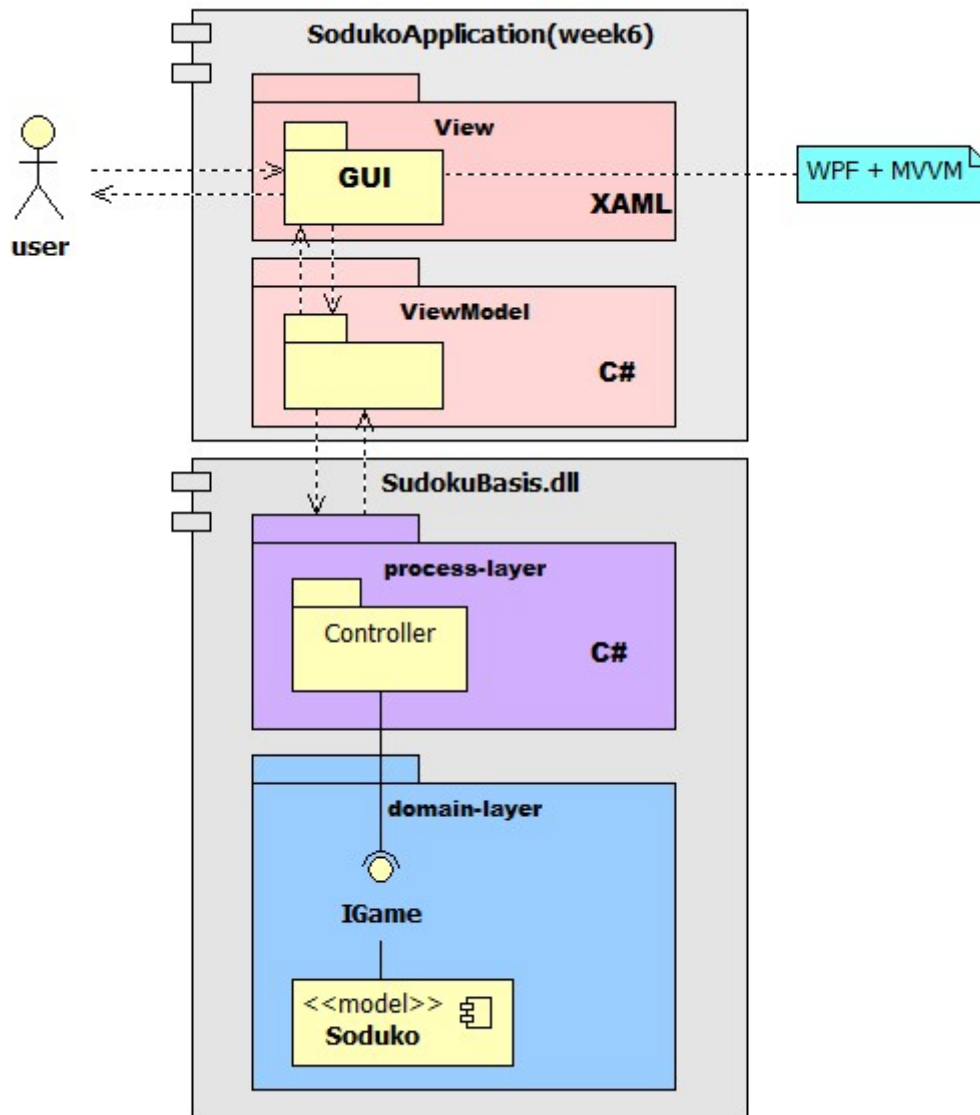
Week5





In week 6 blijft de basis gelijk maar nu moet de applicatie volgens de MVVM techniek worden gebouwd. Als je dit al meteen kunt, dan mag je de opdracht van week 4 en week 6 met elkaar combineren. Als je dit nog niet kunt is het raadzaam om naar de presentatie van Sjoerd vd Meer te gaan op dinsdag 15 mei 2012 (zijn materiaal staat na afloop op blackboard) en om in week 4 eerst het component te "temmen".

Week 6



Aanvullingen:

Er komen nog documenten on-line op blackboard met extra detailinformatie over de opdrachten van week 5 en 6, maar hierboven staat de grote lijn al.