TYPO3.Flow - Feature # 56556

Status:	New	**Priority:**	Should have
Author:	Simon Schaufelberger	**Category:**	
Created:	2014-03-04	**Assigned To:**	
Updated:	2014-04-08	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	support hasProperty and isProperty		

Description

The way how you currently check for existence of a value in fluid is not nice.

```
<f:if condition="{object.hasChildren}">
[...]
</f:if>
```

results in a method in the Model:

```
public function getHasChildren()
```

Instead it should be possible to just write:

```
public funtion hasChildren()
```

Same for "is".

Inspired by: https://github.com/czenker/cz_simple_cal/blob/master/Classes/Domain/Model/Base.php

Related issues:

related to Core - Feature # 56529: support hasProperty and isProperty	**Resolved**	**2014-03-04**

History

#1 - 2014-04-08 00:34 - Alexander Berl

One thing that needs to be discussed before implementing this is, how the old "isSomething" behaviour should be handled, as it would create a redundancy/duplication:

If this is implemented in a BC way, it means that the method "isSomething" is accessible in two ways, by "{object.something}" (unless "getSomething" exists) and "{object.isSomething}" (unless "getIsSomething" exists).
Should this be kept or removed to force explicitt access as with "hasSomething"?

IMHO it would make more sense to require more verbose boolean checks in Fluid by "{object.isSomething}" instead of "{object.something}", which could also mean "getSomething", but that would be a pretty hard breaking change for a lot of Fluid templates.

Proposed solution in Classes/TYPO3/Flow/Reflection/ObjectAccess.php:

```
            if (is_callable(array($subject, $getterMethodName))) {
             self::$propertyGetterCache[$identifier]['accessorMethod'] = $getterMethodName;
         } else {
-            $getterMethodName = 'is' . ucfirst($propertyName);
+            if (substr($propertyName, 0, 2) === 'is' || substr($propertyName, 0, 3) === 'has') {
+                $getterMethodName = $propertyName;
+            } else {
+                $getterMethodName = 'is' . ucfirst($propertyName);
+            }
             if (is_callable(array($subject, $getterMethodName))) {
                 self::$propertyGetterCache[$identifier]['accessorMethod'] = $getterMethodName;
             } else {
```