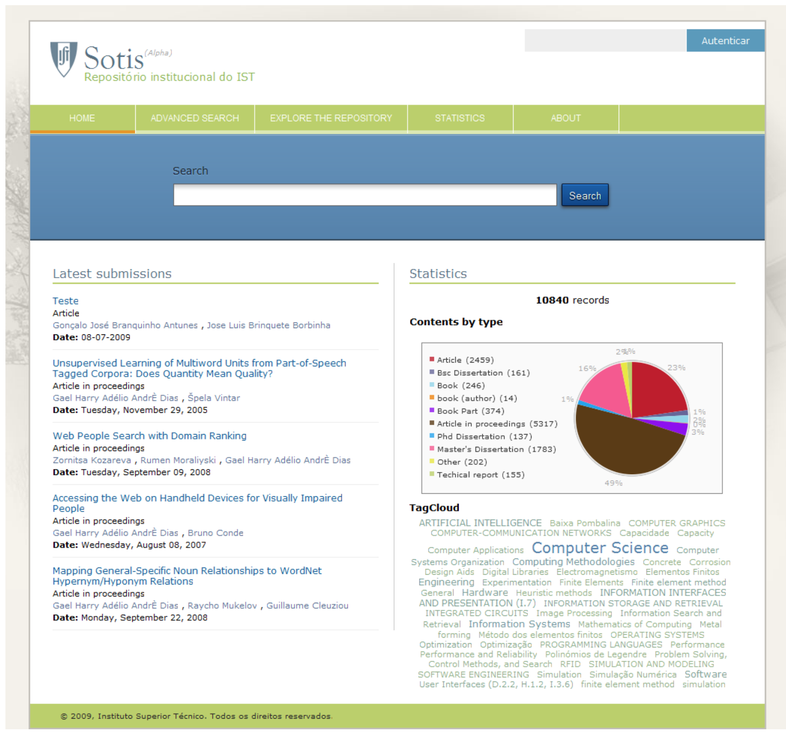
SOTIS
O Repositório Institucional do Instituto Superior Técnico

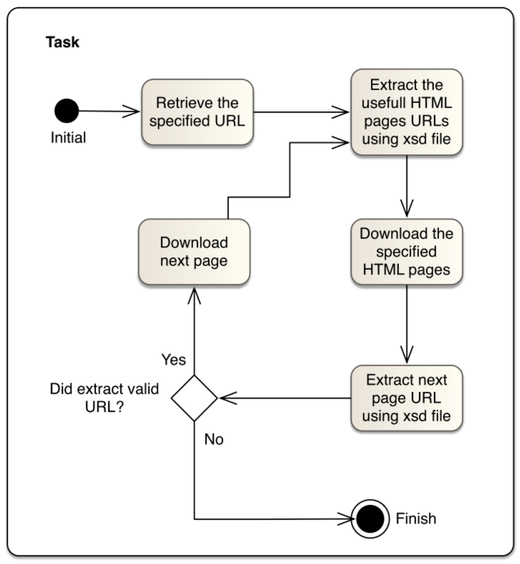
Miguel Coxo - José Borbinha - Joaquim Silva

Os repositórios institucionais permitem o armazenamento, gestão e disseminação da produção intelectual de uma instituição e os membros da sua comunidade. Proporcionam um método complementar ao sistema tradicional de comunicação científica, tornando mais fácil demonstrar o valor científico, social e financeiro da instituição. Os potenciais benefícios de um repositório institucional estendem-se para além do aumento do perfil cientifico de uma instituição. Os repositórios institucionais aumentam ainda a visibilidade dos autores, providenciam aos utilizadores um acesso mais fácil à informação e permitem uma maior disseminação do trabalho apoiado por outras instituições de financiamento.

Apesar do rápido ritmo de adopção dos repositórios institucionais, e de todos os potenciais benefícios que estes oferecem, estudos têm demonstrado que um dos maiores problemas existentes é a falta de contribuição por parte das comunidades das instituições! Aqui propomos uma solução para aliviar este problema e facilitar a manutenção continuada de um repositório institucional. ..



O repositório institucional do IST pretende exigir o mínimo de intervenção humana! A partir da base de dados de docentes, investigadores e alunos do IST, o SOTIS procura novas publicações criadas por esses autores e anunciadas nas bibliotecas digitais dos editores. Quando as encontra, copia os metadados e, quando possível, também os respectivos ficheiros com as obras. Para intervenção humana resta a correcção de erros do processo automáticos, o registo de novas obras que não se conseguem descobrir automaticamente, ou o depósito de ficheiros...



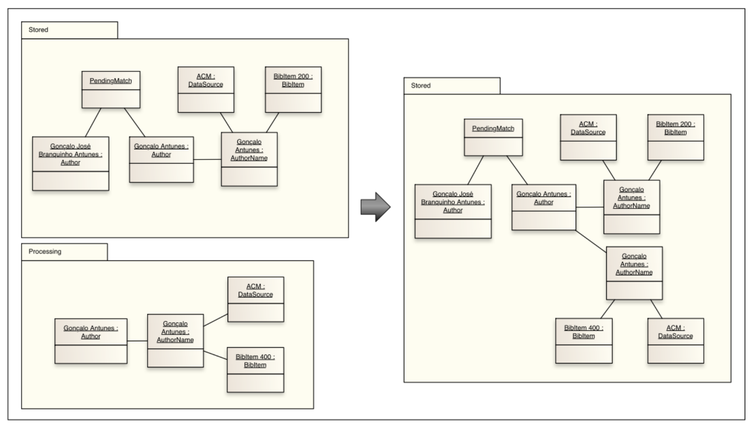
Workflow of the HTML harvester

```
compare(nameA, nameB, baseThreshold)

  nameA = normalize(nameA)
  nameB = normalize(nameB)

  nameA = removeIgnoredWords(nameA)
  nameB = removeIgnoredWords(nameB)

  aNames = getNameList(nameA)
  bNames = getNameList(nameB)


  matchedNames = 0
  forEach( name in aNames )
    forEach ( nameb in bNames )
      if( equals( name, nameb) ) matchedNames++
    end
  end

  matchedInitials = 0
  forEach( name in aNames )
    forEach ( nameb in bNames )
      if( isInitial( name, nameb) ) matchedInitials++
    end
  end

  maxLenght = (aNames.length > bNames.length)? aNames.length : bNames.length

  result = ( matchedNames + matchedInitials*(0.7) ) / maxLenght

  // if first and last name match then its more probable to be the same author so we
  // increase the result. This solves problems with large names
  if ( firstNameAndLastMatch( aNames, bNames )  and result < baseThreshold ) result =
baseThreshold

  // When we have two full names, if the smaller one has a name that the longer one
  // doesn't then its very likely that they are not the same author
  if ( aNames.length > bNames.length )
    forEach( name in bNames )
      if ( !contains(aNames, name) )
        result = reduceTrustValue(result)
      end
    end
  else
    forEach( name in aNames )
      if ( !contains(bNames, name) )
        result = reduceTrustValue(result)
      end
    end
  end

  return result
end
```

Name matching algorithm pseudo-code.

Conceito já validado para as bibliotecas digitais da ACM e SpringerLink…

Process UI screenshot



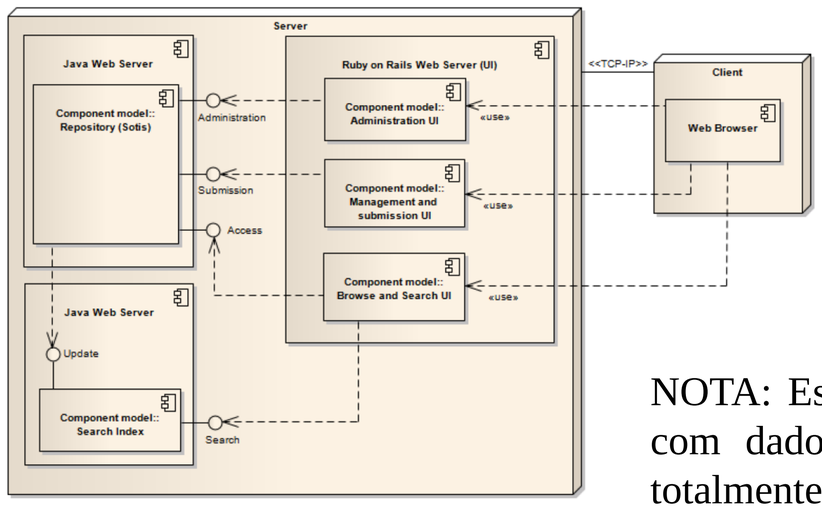
Merging two pending authors

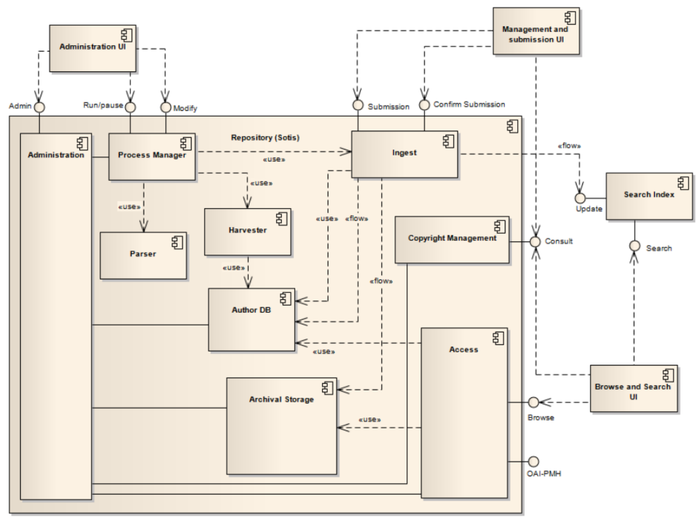
```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:h="http://www.w3.org/1999/xhtml"
  version="2.0">

<xsl:output method="text" />
<xsl:param name="base-url"/>

<xsl:template match="/" xpath-default-namespace="http://www.w3.org/1999/xhtml">
  <xsl:for-each select="(//a[normalize-space(.)='Next'])[1]">
    <xsl:value-of select="$base-url"/><xsl:value-of select="@href" />
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

Springer xslt transformation file for extracting bibliographic works links

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:h="http://www.w3.org/1999/xhtml" version="2.0">

  <xsl:output method="text" />
  <xsl:param name="base-url" />

  <xsl:template match="/" xpath-default-namespace="http://www.w3.org/1999/xhtml">
    <xsl:for-each select="//div[@class='listItemName']/a">
      <xsl:value-of select="$base-url" /><xsl:value-of select="@href" />
      <xsl:text>&#xa;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Springer xslt transformation file for extracting next page links





NOTA: Este sistema é neste momento um protótipo, com dados fictícios. O sistema final deverá estar totalmente operacional até Junho de 2010