

# Førti år med programvarepatenter i Europa:

*En analyse av rettstilstanden og vurderingen av programvarepatenter i Den Europeiske Patentkonvensjon med særlig fokus på muligheten for fremtidig optimalisering innenfor rammene av dagens system*

Kandidatnummer: 192855

Veileder: Knut Martin Tande, førsteamanuensis

Antall ord: 14999



JUS399 Masteroppgave  
Det juridiske fakultet

UNIVERSITETET I BERGEN

27.05.13

## Innholdsfortegnelse

<b>1. Innledning</b> .....	<b>3</b>
<b>1.1. Emne og problemstilling</b> .....	<b>3</b>
1.1.1. Rettspraksis som utgangspunkt for problemstillingen og analysen. ....	4
<b>1.2. Aktualitet</b> .....	<b>6</b>
<b>1.3. Bakgrunnen for analysen: opphavsrett og patentrett</b> .....	<b>7</b>
<b>1.4. Generelt om patentsystemet og utfordringene med programvarepatenter</b> .....	<b>9</b>
<b>1.5. Grunnleggende krav til juridisk argumentasjon</b> .....	<b>10</b>
<b>1.6. Definisjoner og forklaringer</b> .....	<b>12</b>
1.6.1 Programvare og maskinvare.....	12
1.6.2. Patent.....	12
1.6.3 Den Europeiske Patentkonvensjonen.....	13
<b>1.7. Avgrensninger</b> .....	<b>13</b>
<b>2. Patentsystemets begrunnelse som startpunkt for analysen av programvarepatenter</b> .....	<b>15</b>
<b>2.1. Veien videre</b> .....	<b>15</b>
<b>2.2. Patentretten som en naturlig rett</b> .....	<b>15</b>
<b>2.2. Samfunnsøkonomisk begrunnelse</b> .....	<b>16</b>
<b>3. Kritikk av programvarepatenter i lys av patentsystemet</b> .....	<b>17</b>
<b>3.1. Begrunnelsen og formålet til patentsystemet slår ikke til i relasjon til programvarepatenter</b> .....	<b>17</b>
<b>3.2. Er det mulig å vurdere et programvarepatents oppfinnelseshøyde?</b> .....	<b>18</b>
<b>3.3. Programvarepatenter vil hindre oppstart og utvikling av små-og-mellomstore bedrifter</b> .....	<b>20</b>
<b>3.4. Er kritikken mot programvarepatenter holdbar?</b> .....	<b>21</b>
<b>4. Programvare(relaterte)patenter</b> .....	<b>22</b>
<b>4.1. Veien videre</b> .....	<b>22</b>
<b>4.2. De relevante materielle patenterbarhetsvilkår</b> .....	<b>22</b>
4.2.1 Oppfinnelse – Artikkel 52 EPC og Patentloven § 1(1) og (2).....	22
4.2.2. Oppfinnelseshøyde – Artikkel 56 EPC og Patentloven § 2(1) .....	23
<b>4.3. IBM-sakene</b> .....	<b>24</b>
4.3.1. Kravet til ytterligere teknisk effekt.....	24
4.3.2. Ytterligere Teknisk Effekt: Kasuistikk eller system?.....	26
4.3.3. Identifikasjon av spenninger i relasjon til kravet om ytterligere teknisk effekt som grunnlag for optimalisering av rettstilstanden og den juridiske argumentasjon.....	28
<b>4.4. Utviklingen i etterkant av IBM-sakene</b> .....	<b>30</b>
4.4.1. Microsoft-saken .....	30
4.4.2. Identifikasjon av spenningene i Microsoft-saken som grunnlag for optimalisering av rettstilstanden og den juridiske argumentasjon .....	32
<b>4.5. Vurderingen av programvarepatentkrav</b> .....	<b>35</b>
4.5.1. Har unntaket mistet all relevans?.....	35
4.5.2. Forholdet mellom oppfinnelses- og oppfinnelseshøydevilkåret: Identifikasjon av spenninger som grunnlag for optimalisering av rettstilstanden og den juridiske argumentasjon .....	39
<b>5. Avsluttende bemerkninger: Optimaliseringsspørsmålet i den praktiske rettsanvendelsen</b> .....	<b>42</b>
<b>6. Konklusjon</b> .....	<b>45</b>

# 1. Innledning

## 1.1. Emne og problemstilling

Emnet for denne avhandlingen er unntaket for patent på programvare som er en del av Den Europeiske Patentkonvensjonen (EPC). Tradisjonelt knyttes programvare til konvensjonelle datamaskiner, mens man i dag finner det i alt fra mobiltelefoner til industrielle prosesser. Dette medfører en økt betydning av, og økt fokus på, hvordan programvare beskyttes rettslig.

Kjernen i analysen er programvareunntaket i artikkel 52 (2) og (3) EPC hvor det fremgår at "programs for computers (...) shall not be regarded as inventions" dersom patentkravet relaterer seg til "subject matter (...) as such"<sup>1</sup>. Med andre ord kan programvare patenteres så lenge det ikke er tale programvare "as such". Dette har skapt grobunn for spørsmålet om hva som bare utgjør programvare, og hva som vil gå utover dette og dermed potensielt kunne patenteres. Dette vil være et sentralt spørsmål gjennom hele den forestående analysen.

Denne uklarheten som følger av ordlyden har blitt særlig omdiskutert fordi programvarepatenter er meget debattert politisk, så vel som juridisk. En av årsakene til dette kan finnes i selve patentsystemet. Systemet bygger på en rekke hensyn og det er uklart, særlig i relasjon til målet om å fremme innovasjon, hvordan programvarepatenter vil virke inn på disse underliggende hensyn<sup>2</sup>. Dette utgjør også en del av bakgrunnen for kritikken som er rettet mot at programvare potensielt kan patenteres<sup>3</sup>. Utfordringene relatert til hensynene vil også være sentralt gjennom hele fremstillingen. Særlig vil dette ses under spørsmålet om optimalisering av den nåværende rettstilstanden.

---

<sup>1</sup> Patentloven, Lov 15.12.1967 nr. 9 Lov om patenter § (2) nr.3, bruker uttrykket "bare utgjør" som oversettelse av "as such"

<sup>2</sup> Se nærmere under punkt 2 om patentsystemets hensyn og begrunnelse

<sup>3</sup> Se nærmere om kritikken under punkt 3

Utfordringene med ordlyden, og usikkerheten som hersker ved programvarepatenters forhold til de grunnleggende hensyn i patentsystemet, har også påvirket rettspraksisen omkring emnet<sup>4</sup>. Denne har i liten grad vært entydig, selv om praksisen fra Det Europeiske Patentkontoret (EPO)<sup>5</sup> fra slutten av 1990-tallet synes å ha forsøkt å tydeliggjøre hva som utgjør mer enn bare programvare. Problemet er imidlertid at dette langt på vei synes å ha gjort situasjonen mer uklar, slik at heller ikke EPOs praksis har rettet opp uklarhetene som ordlyden og hensynene skaper.

En står da igjen med en uklar rettstilstand som vil danne utgangspunktet for den forestående analysen. Et naturlig sted å rette fokuset er den nevnte rettspraksisen da denne påvirker rettstilstanden direkte, og dermed kan brukes til å klargjøre og forbedre situasjonen. For å presisere målet og innholdet i analysen er det dermed nødvendig å se nærmere på den konkrete problemstillingen særlig i lys av rettspraksis.

### 1.1.1. Rettspraksis som utgangspunkt for problemstillingen og analysen.

For å forklare den aktuelle problemstilling nærmere vil det først være hensiktsmessig å se kort til hvordan den aktuelle rettspraksis har utviklet seg.

På slutten av 1990-tallet startet det med introduksjonen av kravet til ytterligere teknisk effekt som vilkår for at programvare skulle være noe mer enn bare programvare<sup>6</sup>. Nærmest parallelt med denne utviklingen oppstod det en situasjon hvor patentsøknader som involverer programvare integrert på et lagringsmedium ble forsøkt patentert, med den tanke at det dermed ikke lenger bare var programvare<sup>7</sup>. Slike

---

<sup>4</sup> Praksisen kommer i fra to instanser: Boards of Appeal (BoA) og Enlarged Boards of Appeal (EBoA). EBoA er den øverste instansen, men de fleste saker går ikke lenger enn BoA.

<sup>5</sup> EPO er en del av organisasjonen som administrer EPC. Kontoret ledes av en President, som igjen står ansvarlig overfor Forvaltningsrådet. Det er EPO som er ansvarlig for å undersøke og tildele patenter. Samtidig har EPO det administrative ansvaret for BoA og EBoA. EPO vil derfor brukes som en samlebetegnelse på disse to i det følgende.

<sup>6</sup> T 1173/97 (IBM I). Se nærmere under punkt 4.3 flg. Hva som faktisk skaper en slik ytterligere teknisk effekt er et krevende, og sentralt spørsmål, i den forstående analysen

<sup>7</sup> Se blant annet T 0424/03(Microsoft). Dette vil i fortsettelsen betegnes som "kombinasjonssøknader". Poenget er at noe teknisk, typisk en harddisk, kombineres

kombinasjonssøknader har ført til at to av de mest sentrale patenterbarhetsvilkår, oppfinnelse og oppfinneshøyde, har fått et uklart skille seg i mellom. Denne uklarheten har fått sitt utslag ved at kombinasjonssøknader knyttet til programvare særlig vurderes under oppfinneshøydevilkåret, mens oppfinnelsesvilkåret med tilhørende programvareunntak blir nedtonet<sup>8</sup>.

Det foreligger dermed et åpenbart behov for avklaring. Behovet synes både å knytte seg til forståelsen av rettstilstanden slik den rent faktisk er i dag, og også, i lys av de utfordringer som er skissert, hvorvidt det er noen mulighet for å optimalisere denne innenfor rammen av dagens system. Problemstillingen er dermed langt på vei to-delt. Rettstilstanden vil først analyseres, før optimaliseringsspørsmålet vurderes nærmere<sup>9</sup>. Dette vil gjøres slik at rettstilstanden vil deles inn etter den tre-leddete utvikling som er skissert rett ovenfor, som starter med kravet til ytterligere teknisk effekt og ender med den særlige vurderingsmåten av programvarepatentkrav. I tilknytning til hver enkelt del av rettstilstanden vil så spørsmålet om optimalisering behandles forløpende. Det vil si spørsmålet om de tvilsspørsmål som er vist ovenfor kan løses på en bedre måte enn det de gjøres i dag. Analysen vil hele veien ses i lys av de grunnleggende hensyn i patentsystemet samt kritikken som er reist mot programvarepatenter.

Det er altså rettspraksis som i all hovedsak har skapt rettstilstanden, og det er også denne som vil brukes til å optimalisere den. Ved å se til praksis blir det enklere å få kartlagt avstanden mellom dagens tilstand og den optimaliserte verdi-og fornuftmessig koherente rettstilstand. Samtidig blir det tydeligere at spørsmålet om optimalisering er noe som kan påvirke rettstilstanden direkte, slik at det er tale om en analyse de lege lata.

Spørsmålet om optimalisering vil særlig dreie seg om økt fokus i EPOs praksis på de underliggende hensyn i patentsystemet, samt økt synliggjøring av øvrige motsetninger i rettskildene og grunnleggende hensyn i rettssystemet<sup>10</sup>. Dette er et ansvar som ligger på

---

med noe ikke-teknisk, her programvare, for å komme seg rundt unntaksbestemmelsen. Se nærmere punkt 4.4 flg.

<sup>8</sup> Se blant annet T 154/04 (Duns). Dette skal det ses nærmere på særlig under punkt 4.5

<sup>9</sup> Se punkt 4 hvor denne inndelingen vil anvendes

<sup>10</sup> For en grundig innføring i en slik tenkemåte se Tande Knut Martin, Individuelle Valg og vurderinger i rettsanvendelsesprosessen, Jussens Venner nr.1 2011, s. 1-36

rettsanvenderen ved at han underveis i rettsanvendelsesprosessen i størst mulig grad må synliggjøre og balansere konfliktene som oppstår for på den å måte skape en juridisk argumentasjon som er mer presis, etterprøvable og avklarende enn dagens situasjon<sup>11</sup>.

Ved at rettsanvenderen, her dommerne i EPO, følger disse argumentasjonskrav vil en få en rettsstilstand som er mer verdi-og fornuftmessig koherent. Med andre ord en rettsstilstand som baserer seg på en eksplisitt og helhetlig avveining av de konflikter som ligger i rettskildene og som oppstår i den konkrete sak, før en konklusjon fattes. Målet er altså å vise en vei innenfor det legitimeringsintervallet som EPO i dag opererer i, som i større grad tar hensyn til de tvilsspørsmål som er skissert ovenfor<sup>12</sup>.

Det er imidlertid ikke gitt at en større grad av verdi-og fornuftmessig koherens er mulig eller ønskelig. Grensenytten av en optimalisering av den juridiske argumentasjonen, og fokuset på de underliggende hensyn i patentsystemet, kan være så liten at det ikke er lønnsomt i praksis. Likevel er det klart at det preg av vilkårlighet og pragmatisme som i dag preger EPOs praksis, og dermed rettsstilstanden, i stor grad kan forbedres<sup>13</sup>. Dette fører også til at den retning som i størst grad gir verdi-og fornuftmessig koherens er også den retning som EPO bør velge når programvarepatenter vurderes.

## 1.2. Aktualitet

At programvare er et aktuelt tema ligger i dagen. Det mangfold av apparater og maskiner som er avhengig av programvare gjør at programvare er en naturlig del av hverdagen til de fleste mennesker. Dette medfører også naturlig nok at dette er en bransje hvor det har vært en stor vekst de siste tiår.

I følge MarketLine omsatte programvareindustrien på verdensbasis for over 292 milliarder dollar i 2011, og i 2016 er det forventet at dette er økt til nesten 397

---

<sup>11</sup> Se punkt 1.5 for en innføring i kravene til den juridiske argumentasjon

<sup>12</sup> Dette bekrefter at det er tale om en analyse de lege lata

<sup>13</sup> Se punkt 5 om dette

milliarder<sup>14</sup>. I Europa omsatte de hundre største programvarebedriftene for over 30 milliarder euro i 2011. Dette gjør at programvareindustrien er viktig for økonomien i Europa.

Et søk på "software patents" på Google gir rundt 37 000 000 treff<sup>15</sup>. Temaet er nok det mest debatterte område innenfor patentretten, og diskusjonen er til dels svært opphetet. Det finnes mange såkalte åpen kildekode-organisasjoner, som lobbyer hardt mot at patenter skal gis<sup>16</sup>. Grunnleggeren av teknologiselskapet Apple, Steve Jobs, uttalte om Google Android<sup>17</sup>: "I'm going to destroy Android, because it's a stolen product, I'm willing to go to thermonuclear war on this."<sup>18</sup>. Å se nærmere på diskusjonen er nyttig for å finne nærmere ut av årsakene til de sterke uenighetene, og for å få en dypere forståelse av temaet som sådan<sup>19</sup>.

I denne sammenheng er det og interessant å se til konklusjonen til Digiutvalget da de uttalte seg om programvarepatenter<sup>20</sup>. De anbefaler at europeisk praksis bør følges av norske myndigheter, men at det ikke bør arbeides for å øke patenteringsgraden<sup>21</sup>. Altså har norske patentmyndigheter en klar oppfordring om å følge EPO sin praksis, noe som øker aktualiteten av denne analysen.

### 1.3. Bakgrunnen for analysen: opphavsrett og patentrett

---

<sup>14</sup> <http://www.reportlinker.com/p01080569-summary/Software-Global-Industry-Almanac-MarketLine.html> (Lenke innhentet:27.06.2013)

<sup>15</sup> Søk foretatt 09.05.13

<sup>16</sup> Eksempler er Free Software Foundation Europe: <http://fsfe.org> og End Software Patents: <http://endsoftpatents.org> (Lenker innhentet:27.06.2013)

<sup>17</sup> Android er verdens største åpen-kildekode operativsystemet for håndholdte enheter, som mobiltelefoner, hvor brukerne selv fritt kan utvikle tilleggsprogramvare. Apple er på motsatt side kjent for sitt mer lukkede system og aktiv bruk av patenter.

<sup>18</sup> Isaacson Walter (2011). Steve Jobs: The Exclusive Biography. New York:Simon and Schuster, s 512

<sup>19</sup> Se punkt 3 nedenfor

<sup>20</sup> NOU 2013:2 (Digiutvalget) Punkt 4.10 siste avsnitt

<sup>21</sup> Dette kan i særlig grad knyttes til kritikken nedenfor under punkt 3.

Programvare har vært aktuelt i lang tid, og er ikke et nytt fenomen. Oftest blir det hevdet at det første rene programvarepatentet ble gitt i Storbritannia i 1966 til Slee og Harris<sup>22</sup>. EPO hadde i 2002 gitt over 30 000 programvarerelaterte patenter<sup>23</sup>. Det gis derfor patent på programvare i Europa i dag, noe som har sammenheng med at programvareunntaket ikke er et unntak fra å gi patent på programvare som faktisk innebærer en *oppfinnelse*.

Dette aktualiserer forholdet mellom opphavsrettslig og patentrettslig beskyttelse av programvare. Den alminnelige beskyttelsen av programvare skjer i dag gjennom opphavsretten<sup>24</sup>. Dette innebærer at programvare blant annet er beskyttet mot direkte kopiering og bruk uten avtale. Direkte kopiering vil typisk være at en annen part benytter kildekoden til det opprinnelige program.

Problemet med opphavsretten er blant annet at den er avgrenset slik at den verken beskytter ide, intensjon eller innhold<sup>25</sup>. En vanlig situasjon som oppstår kan eksemplifiseres med en ny sak fra EU domstolen, SAS Institute Inc vs World Programming LTD(WPL)<sup>26</sup>. Kjernen i saken var at WPL hadde utviklet et program med lik funksjonalitet som SAS sitt. Måten WPL hadde gått frem på var å studere SAS sitt program og manualer, samt testet dets funksjoner. Altså hadde de ikke på noen måte kopiert kildekoden eller hatt tilgang til denne. Resultatet i saken ble at funksjonene, programmeringsspråket og filformatet i SAS sitt program *ikke* var beskyttet via opphavsretten.

Eksempelet viser hvor patentretten vil komme inn. Ved patentbeskyttelse av programvare vil i all hovedsak også funksjonaliteten til programvaren bli beskyttet.

---

<sup>22</sup> Beresford Keith (2000). Patenting Software Under the European Patent Convention. UK:Sweet & Maxwell Limited, side 4

<sup>23</sup> Proposal for a Directive on the Patentability of computer-implemented inventions – frequently asked questions, [http://europa.eu/rapid/press-release\\_MEMO-02-32\\_en.html](http://europa.eu/rapid/press-release_MEMO-02-32_en.html). (Lenke innhentet:27.06.2013)

<sup>24</sup> Se Lov av 12.05.1961 nr. 2 Lov om opphavsrett til åndsverk m.v. (åndsverkloven) § 1(2) nr.12, EUs Programvaredirektiv (Direktiv 2009/24/EU) og Trade-Related Aspects of Intellectual Property Rights (TRIPS) artikkel 10

<sup>25</sup> Dette er sikker norsk rett og følger av Artikkel 1 (2) andre setning i Programvaredirektivet

<sup>26</sup> C-406/10



Dette vil skape en bredere beskyttelse, og ønsket om bredere beskyttelse er et sted hvor tilhengere og motstandere av programvarepatenter skiller vei.

*I norsk rett vil resultatet bli likt som i SAS-saken da Norge tilpasset Åndsverkloven til Software Direktivet ved lovendring av 17.06.2005 nr.97. Norge er gjennom fortalen til EØS pliktig til å opprettholde en lik fortolkning som EU domstolen når det gjelder de rettsaker som er tatt inn i norsk rett.*

#### **1.4. Generelt om patentsystemet og utfordringene med programvarepatenter**

Siden patentretten blir det sentrale i analysen er det avgjørende å tidlig få et innblikk i patentsystemet og utfordringene programvarepatenter skaper. Patentsystemet er ikke nytt. Allerede på 1400-tallet fikk Venezia den første patentloven i verden<sup>27</sup>. Før dette hadde imidlertid beskyttelse av kunnskap i Venezia en dramatisk start. På 1200-tallet ble glassblåsere plassert på øyen Murano like utenfor byen slik at staten kunne kontrollere at ingen av dem reiste bort i fra området. Dem som forsøkte å rømme ble drept. Dette kan sies å være en slags forløper til et moderne patentsystem hvor staten ville kontrollere kunnskapen slik at den ikke ble transportert ut av riket. Utover ønsket om å beholde kunnskap innenfor den aktuelle stat har ordningen imidlertid ingen likheter med dagens system.

I dag kreves det at det foreligger en oppfinnelse for at patentering skal bli aktuelt. Samtidig er det moderne patentsystemet bygget opp rundt en tanke om at patenter skal stimulere til innovasjon innenfor teknikk, og det er teknikkkravet som står sentralt under oppfinnelsesvilkåret både i norsk rett og EPC<sup>28</sup>. Interessant er det da at det er særlig på dette punkt at programvarepatenter er omdiskutert. Det er stilt spørsmål ved om programvare stimulerer til innovasjon, eller om det tvert om hindrer det. Dette kan

---

<sup>27</sup> Stenvik Are (2006). Patentrett, andre utgave. Oslo:Cappelen Akademisk Forlag, side 16

<sup>28</sup> I nordisk sammenheng kan det vises til den Nordiske utredningen om patentloven av 1967 hvor det sentrale ved oppfinnelsesvilkåret ble ansett å være den tekniske karakter, NU 1963:6 s. 96 sjette avsnitt og s. 97 andre, tredje og syvende avsnitt

særlig knyttet til det faktum at programvare ofte utvikles i små steg<sup>29</sup>. Samtidig er rettstilstanden ved kombinasjonssøknader blitt slik at inkluderingen av enhver type teknikk i patentkravet er tilstrekkelig for å komme seg rundt unntaket, noe som kan føre til en bagatellisering av unntaket<sup>30</sup>.

Dette henger som nevnt innledningsvis tett sammen med oppfinneshøydevilkåret. Oppfinneshøyde innebærer at gjenstanden for patentkravet må skille seg vesentlig fra tidligere oppfinnelser. Følgen av kombinasjonssøknader i relasjon til programvarepatenter er at det skapes en uklar overgang mellom de to vilkårene, noe som kan føre til at den juridiske argumentasjonen blir lite tilfredsstillende. Sammenhengen mellom dem er derfor en sentral del av problemstillingen, og skal analyseres i dybden senere<sup>31</sup>. Denne presiseringen viser imidlertid at en ikke kan snakke om programvareunntaket alene, men må se det i sammenheng med de andre materielle vilkår.

## 1.5. Grunnleggende krav til juridisk argumentasjon

I nær tilknytning til patentsystemet har en de grunnleggende krav til juridisk argumentasjon. Ovenfor er den juridiske argumentasjonen knyttet særlig til synliggjøring og avklaring av spenninger i rettskildematerialet. Dette handler da om at rettsanvenderen må være bevisst på at underliggende verdier i rettskildematerialet vil kunne stå mot hverandre, og at disse ideelt sett må balanseres eksplisitt. På den måten er det enklere å komme til et korrekt resultat konkret og generelt, samtidig som det skapes en etterprøvable juridisk argumentasjon. Målet er da å få en verdi- og fornuftmessig koherens både i den konkrete saken, generelt på patentrettens område, og i rettssystemet som helhet<sup>32</sup>.

---

<sup>29</sup> Se mer under punkt 4.4.2

<sup>30</sup> Se mer under punkt 4.5.1

<sup>31</sup> Se særlig punkt 4.4. flg.

<sup>32</sup> Det er særlig Tande (2011) og Sæther Mæhle Synne, Gjelder det andre regler for rettslig argumentasjon i rettsdogmatikken enn for domstolene?, Jussens Venner 2004, side 329-342, som har tatt til orde for denne fremgangsmåten.

Et utgangspunkt ved denne tilnærmingen er at alle jurister i en demokratisk rettsstat har et minste felles multiplum, ved at måten å argumentere på må være i samsvar med visse grunnprinsipper<sup>33</sup>. Særlig vil dette kunne sies å være hensynet til demokratisk legitimitet, forutberegnelighet og rimelighet<sup>34</sup>. Dersom disse hensynene ikke synliggjøres og behandles tilstrekkelig vil det i den enkelte sak svekke dens vekt, og i et lengre perspektiv skape problemer med tilliten til rettssystemet.

Rettskildelæren som skal anvendes i det følgende bygger på den tradisjonelle lære, men har et klarere fokus på bakenforliggende verdier og hensyn innenfor det aktuelle rettsområde<sup>35</sup>. Poenget er at det vil anlegges en mer helhetlig og systematisk tilnærming for å søke koherens i det foreliggende rettskildematerialet<sup>36</sup>. Det handler om en anerkjennelse av at den tradisjonelle lære kan føre til at argumentasjonen ikke blir god nok, noe som igjen skaper en risiko for feil resultat. Målet med denne mer helhetlige tilnærmingen er da i større grad å kvalitetssikre argumentasjonen.

Dette gjør at det i det følgende ganske selvsagt vil være et avvik i fra praksisen i EPO og det optimale i lys av den juridiske argumentasjonen. Mæhle uttaler i den sammenheng at domstolenes verdiorientering primært er lokal – styrt av den aktuelle rettstvisten - mens den rettsdogmatiske argumentasjonen i alle fall ideelt sett skal være mer overordnet, dyptgripende og fyllestgjørende<sup>37</sup>. En kommer derfor ikke unna at kravene til argumentasjonen ikke kan være like ved avgjørelsen av en konkret sak og innenfor rettsvitenskapen<sup>38</sup>. Langt på vei vil det handle om å optimalisere, og dermed minimere risikoen for feil, i størst mulig grad innenfor de praktiske og rettslige rammer som eksisterer. Før den nærmere analysen vil det nå først ses på noen avgjørende begreper og avgrensninger.

---

<sup>33</sup> Nygaard Nils (2004). Rettsgrunnlag og standpunkt, andre utgave. Bergen: Universitetsforlaget. Se side 60 flg. som taler om en ulovfestet lojalitetsplikt overfor loven og rettssystemet. Denne må antas å gjelde like sterkt for dommerne i EPO som i Norge, selv om grunnlaget for plikten er ulikt.

<sup>34</sup> Se Tande (2011) side 23 punkt 6.1

<sup>35</sup> Se om den tradisjonelle rettskildelære særlig Eckhoff Torstein (2001). Rettskildelære, femte utgave. Oslo: Universitetsforlaget

<sup>36</sup> Dette kan sies å være en mer vitenskapelig tilnærming, se Tande (2011) punkt 4.4

<sup>37</sup> Mæhle (2004), side 339

<sup>38</sup> Men som Mæhle (2004) konkluderer med i punkt 9 er det i all hovedsak en gradsforskjell

## 1.6. Definisjoner og forklaringer

### 1.6.1 Programvare og maskinvare

Ordet "programvare" stammer fra ordene "pro", før, og "grafein", skrevet. Det er altså noe som er skrevet på forhånd, og i programvaretermer vil dette knyttes til kildekoden. At programvare er skrevet er en av hovedårsakene til at opphavsrett utgjør standardbeskyttelsen siden opphavsrett beskytter den åndelige kreasjon. Samtidig er dette også en av årsakene til at det argumenteres for at patentsystemet ikke passer til programvare.

Programvare er forsøkt definert på ulikt vis. I IBM I bruker BoA definisjonen "a set of instructions which, when the program is loaded, makes the hardware execute a specific procedure producing a particular result"<sup>39</sup>. Det er altså programmereren som skriver programmet, og når dette programmet blir lastet opp på for eksempel en mobiltelefon og interagerer med maskinvaren, får det en fysisk manifestasjon ut mot brukeren som kan anvende de funksjoner som er lagt inn.

Programvare må avgrenses mot maskinvare. Maskinvaren er for eksempel en harddisk, et hovedkort eller en Blu-Ray avspiller. Dette er altså de fysiske elementene. Disse er nødvendige for at programvaren skal fungere, og som det vises senere spiller maskinvaren en sentral rolle i dagens rettstilstand.

### 1.6.2. Patent

Patentbeskyttelse er en juridisk beskyttelse av en teknisk oppfinnelse som gis mot at denne oppfinnelsen gjøres offentlig<sup>40</sup>. Dette skal da gi beskyttelse til oppfinnerne,

---

<sup>39</sup> IBM I punkt 9.2

<sup>40</sup> At beskyttelse gis mot offentliggjøring kalles gjerne "the patent bargain"

samtidig som at beskyttelsen sammen med offentliggjøringen skal være en motivasjon for andre til å være innovative. Beskyttelsen er en enerett til blant annet å tilvirke og sette produktet på markedet som varer i 20 år<sup>41</sup><sup>42</sup>. Om produktet faktisk *kan* settes på markedet er et spørsmål som reguleres av særlovgivning, for eksempel Legemiddeloven<sup>43</sup>.

### 1.6.3 Den Europeiske Patentkonvensjonen

EPC er den primære europeiske rettskilden for patenter, og danner utgangspunktet for denne analysen. Foruten å gi materielle patentregler gir også EPC prosessuelle regler. Disse er svært viktig for å skape harmonisering da patenter som hovedregel er territorielle. Patentsøknader kan sendes til et av de nasjonale kontorene, med påskrift om at det ønskes patent også i et eller flere medlemsland<sup>44</sup>. På den måten kan en få en internasjonal beskyttelse, noe som er særlig viktig i dagens teknologisamfunn hvor kunnskap raskt flyttes over landegrenser. Det er imidlertid de materielle patentreglene som står i fokus i denne analysen.

## 1.7. Avgrensninger

I all hovedsak skal det som er skrevet ovenfor avklare hva som er temaet for analysen. Det er imidlertid noen ytterligere spørsmål som det må avgrenses mot. Ikke fordi det ikke er viktige spørsmål ved programvarepatenter, men fordi de faller på siden av problemstillingen.

Hovedfokuset i analysen vil være på EPO sin praksis. Den sentrale rollen til rettspraksis er velkjent fra et norsk perspektiv da Høyesterett er meget viktig når innholdet i norsk rett fastlegges. Likevel vil norsk rett, med Patentloven i spissen, kun være et bakteppe

---

<sup>41</sup> Se patentloven § 3 og § 40. Merk likevel at en har sui generis mekanismer som særlig gjelder legemidler hvor beskyttelsestiden kan suppleres med inntil 5 år.

<sup>42</sup> Etter at produktet er satt på markedet med patenthaverens samtykke mister han eneretten over de enheter som er gjort tilgjengelig. Dette er et spørsmål om konsumpsjon og skal ikke analyseres nærmere, se patentloven § 3 (3) nr.2

<sup>43</sup> Lov av 12.04.1992 (Lov om Legemidler mv.)

<sup>44</sup> Se EPC artikkel 3 og patentloven § 66a (2)

som trekkes frem dersom den anses å tilføre noe utover EPC alene. Nedtoningen av den norske retten kan forsvares fordi Norge ratifiserte EPC i 2008, og allerede i 1979 tilpasset Patentloven med Konvensjonens regler<sup>45</sup>. Dermed vil Norge være folkerettslig forpliktet til å følge Konvensjonen, og sterke harmoniseringshensyn taler for at praksisen til EPO bør følges av de relevante norske myndigheter. Dermed er situasjonen den, at det som i det følgende sies om EPC, også for alle praktiske formål vil være avgjørende for den norske rettstilstanden.

*Det kan i denne sammenheng også nevnes at Norge gjennom protokoll 28 EØS artikkel 3 nr.4 er forpliktet til å "comply" med reglene i EPC. I lys av homogenitetsmålsetningen i EØS vil det tale sterkt for at også praksisen til EPO må følges, da det ikke er nok å kun rent faktisk ha like regler. Stenvik argumenter videre for at EPC er inkorporert i EØS-avtalen, og dermed etter EØS loven § 1 går foran norsk rett ved motstrid<sup>46</sup>. I Rt.2008.1555 uttalte Høyesterett om forholdet til EPO sin praksis at det er av "stor betydning at utviklingen innen Europa er preget av ensartede regler og praktisering innenfor patentområdet", selv om praksisen ikke er formelt bindende i seg selv.*

Det vil heller ikke sies noe om Wien Konvensjonens (WK) tolkningsregler<sup>47</sup>. Disse anvendes ved tolkingen av EPC, men det vil i denne analysen i all hovedsak være fokus på praksisen fra EPO. Det er denne praksisen som har fastsatt rettstilstanden, slik at en isolert tolking av EPC i lys av WK ikke synes å gi et avgjørende bidrag. Imidlertid kan WK artikkel 31 (3) B nevnes da denne viser at "any subsequent practice in the application of the treaty" er relevant. Under denne vil praksisen fra EPO havne.

Som et ledd i analysen av rettstilstanden vil det nedenfor ses på de viktigste argumenter mot programvarepatenter. Faren for monopolisering er et sentralt motargument. Særlig innenfor telekomindustrien er dette et spørsmål. Imidlertid faller det på siden av denne analysen fordi problemstillingen er spisset inn mot patentsystemet og den juridiske

---

<sup>45</sup> Se mer om dette NOU 1976:49, særlig kapittel VI

<sup>46</sup> Se Stenvik Are (2001). Patenters Beskyttelsesomfang. Oslo: Cappelen Akademiske Forlag, s 189

<sup>47</sup> EPO er ikke part i Wien Konvensjonen for tolking av traktater, men det følger av G 05/83 at denne vil bli brukt av EPO ved tolkingen av EPC.

argumentasjon. De videre påståtte konsekvenser, som faren for monopol, vil ikke adresseres. De motargumenter som analyseres nærmere anses da for å være de mest sentrale.

## **2. Patentsystemets begrunnelse som startpunkt for analysen av programvarepatenter**

### **2.1. Veien videre**

Frem til nå er innholdet i problemstillingen presisert og avklart. Når den nærmere analyse av problemstillingen nå starter vil det først ses på begrunnelsen bak patentsystemet. Årsaken er at når rettstilstanden og optimaliseringsspørsmålet gjennomgås vil en se at det er spenninger mellom patentsystemets begrunnelse og EPO sin praksis. Avklaring av disse blir sentralt dersom rettstilstanden skal bli verdi- og fornuftsmessig koherent.

Samtidig vil dette også ha en nær sammenheng med kritikken som omtales nedenfor under punkt 3. Kritikken mot programvarepatenter er viktig da den vil bidra til økt forståelse av programvarepatenter, samtidig som den danner bakgrunnen for flere av spenningene som vil adresseres under spørsmålet om optimalisering.

### **2.2. Patentretten som en naturlig rett**

Tradisjonelt kan patentsystemets formål deles inn i en merkantilistisk og utilitaristisk tilnærming, og en deontologisk tilnærming<sup>48</sup>. Førstnevnte setter staten i fokus, ved at staten fritt kan velge om de vil gi patent eller ikke. Patentsystemet har da utelukkende som mål å føre til vekst. Teknologiske fremskritt kommer særlig staten til gode, og belønning av oppfinneren er ikke i fokus.

---

<sup>48</sup> Stenvik (2001) side 63 flg foretar den samme inndeling

Den deontologiske tilnærmingen, som fikk gjennomslag i patentlovgivningen i Norge med den første patentloven av 1885, går videre og ser til at patenter også er rettferdig for oppfinneren. Når en person har funnet opp noe er det rimelig at han verdsettes for det. Disse rimelighet- og rettferdighetshensynene ser en i dagens system hvor en har rettskrav på patent dersom vilkårene er oppfylt. Staten kan altså ikke gi patent etter eget forgodtbefinnende. Særlig her vil stimulering til innovasjon være sentralt fordi det er mer attraktivt å være innovativ dersom en tjener på det selv.

I forlengelsen av dette kan det vises til den britiske filosofen John Locke, som utledet fra menneskenes iboende rett til å være frie "en moralsk rett for enhver borger til resultatene av sitt arbeid"<sup>49</sup>. Med denne tilnærmingen blir patentretten sett på som en naturlig rett for oppfinneren. Dette er et sentralt hensyn i patentretten slik den er under EPC og i Norge i dag, selv om det i praksis ofte vil være bedriften hvor oppfinneren arbeider som vil eie oppfinnelsen<sup>50</sup>.

## 2.2. Samfunnsøkonomisk begrunnelse

Patentsystemet er uløselig knyttet til samfunnsøkonomien. Dersom systemet ikke motiverer til forskning og innovasjon, som igjen skaper positive verdier, så faller en av de mest grunnleggende begrunnelser for systemet.

Målet om å stimulere til innovasjon kan sies å være todelt. Den ene delen, som nevnt ovenfor, er å stimulere den enkelte oppfinner. Den andre delen retter seg mer mot selskaper som ofte må ta stor økonomisk risiko for å utvikle nye produkter, og patentsystemet kan da tilby den beskyttelse som gjør risikoen mindre. Dette er særlig aktuelt innenfor legemiddelbransjen. Begge sider av innovasjonsstimuleringen vil kunne føre til samfunnsøkonomisk vekst.

---

<sup>49</sup> Samme som over, side 96 med videre henvisninger

<sup>50</sup> Spørsmål rundt dette skal ikke berøres, men for en kort innføring se Helset Per, Reimers Felix, Melander Stene Toril og Vik Ragnar (2009). Immaterialrett og produktetterlikning m.v etter markedsføringsloven. Oslo: Cappelen Damm. Punkt 4.9 side 415



I sammenheng med patentrettens mål om å stimulere til innovasjon ligger og det at offentliggjøringen vil kunne være et insentiv til å skape lignende produkter<sup>51</sup>, og at det gir allmenheten fri tilgang til teknologien når beskyttelsestiden er utløpt. Den motsatte side av dette er at offentliggjøringen hindrer at personer bruker ressurser på like oppfinnelser. Offentliggjøringen er dermed langt på vei et hensyn i seg selv, men nært koplet til insentivhensynet, og vil være en viktig faktor i å øke den samfunnsøkonomiske lønnsomheten.

Økonomer har likevel funnet det vanskelig å komme frem til en konklusjon når det gjelder hvordan å optimalisere patentsystemet<sup>52</sup>. Som det vises nedenfor under punkt 3.1 er denne usikkerheten kanskje særlig gjeldende på området for programvarepatenter. Dette problemet er nok også bakgrunnen for et berømt sitat av Machlup:

“if we did not have a patent system, it would be irresponsible, on the basis of our present knowledge of its economic consequences, to recommend instituting one. But since we have had a patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it”<sup>53</sup>.

### 3. Kritikk av programvarepatenter i lys av patentsystemet

#### 3.1. Begrunnelsen og formålet til patentsystemet slår ikke til i relasjon til programvarepatenter

At patenter vil hindre utvikling og innovasjon av ny programvare er allerede nevnt som en av de mest kontroversielle sider ved programvarepatenter. Dagens system hvor

---

<sup>51</sup> Dette blir gjerne kalt ”incentive to invent around” og kan være positivt for å skape konkurranse i markedet

<sup>52</sup> Bakels Reinier B. ”The Economics of the Subject-Matter Test in the Patent Law”, [http://www.epip.eu/conferences/epip04/files/BAKELS\\_Reinier.pdf](http://www.epip.eu/conferences/epip04/files/BAKELS_Reinier.pdf). (Lenke innhentet:27.06.2013)

<sup>53</sup> Machlup Fritz (1958). ”An Economic Review of the Patent System”, s.80 andre avsnitt

programvare primært beskyttes via opphavsrett gir en beskyttelse av kildekoden. Imidlertid, som vist ovenfor under punkt 1.3, er ideene og funksjonaliteten ikke beskyttet. Dette er bakgrunnen for at en har programmer som for eksempel Linux og Open Office, som har samme funksjoner og mer eller mindre ligner rent estetisk på Microsoft Windows og Microsoft Word.

Ved patent får oppfinneren en bredere beskyttelse, og dermed kan slike åpen-kildekode programmer bli stoppet. Det blir påstått at dette vil hindre innovasjon fordi det første programmet som får patent i teorien kan stoppe lignende program fra å komme på markedet, eller kreve lisenspenger som følge av patentinngrep. I forlengelsen av dette vil det kunne oppstå en situasjon hvor ny programvare ikke blir utviklet fordi en ikke ønsker å ta den risikoen det medfører.

Hvor sterkt dette argumentet er, er noe uklart. Som ved patenter generelt er samfunnsøkonomer usikre på den egentlige effekten av programvarepatenter<sup>54</sup>. R.P. Merges har imidlertid på basis av undersøkelser påstått at de virkelig suksessfulle programvarebedriftene fra 1990-tallet var de som hadde en klar strategi når det gjaldt patenter, og dermed brukte systemet<sup>55</sup>.

Synliggjøring av denne spenningen blir dermed viktig for å få en størst mulig grad av verdi- og fornuftmessig koherens i EPOs praksis og i rettstilstanden. Som det da også skal ses nedenfor under punkt 4-5 er spenningen med på å skape utfordringer i relasjon til spørsmålet om optimalisering.

### **3.2. Er det mulig å vurdere et programvarepatents oppfinneshøyde?**

En av årsakene til at unntaket i EPC artikkel 52, og senere patentloven § 1(2), ble tatt inn var at programvare var relativt nytt på 1970-tallet. Dette skapte stor usikkert om

---

<sup>54</sup> Leith Phillip (2007). *Software and Patents in Europe*. UK: Cambridge University Press, s 87

<sup>55</sup> Merges, R. P. (2006). "Patents, Entry and Growth in the Software Industry." Working paper, UC. Berkeley School of Law

hvorvidt det var mulig på en god måte å vurdere oppfinnelseshøyden til den type oppfinnelser.

Problemet er at dersom dette er vanskelig eller umulig så øker risikoen for såkalte trivielle patenter, det vil si patenter av lav kvalitet<sup>56</sup>. Dette kan føre til at patenter gis på enkel teknologi som mange anvender. Dersom en person eller bedrift får disse rettighetene kan det igjen hindre utvikling, noe som er det motsatte av hva patentsystemet søker å oppnå.

Denne utfordringen illustreres og av Patent Cooperation Treaty (PCT) som skaperne av EPC var påvirket av. PCT Rules 39 og 67 inneholder en regel om at søksavdelingen, som søker etter den tidligere kjente teknikk, og den preliminære eksaminasjonsavdelingen, som foretar den første gjennomgang av søknaden, ikke har noen plikt til å foreta et søk eller undersøkelse dersom søknaden gjelder programvare. Selv om PCT ikke inneholder materielle regler gir disse prosessuelle reglene uttrykk for den bekymringen man hadde rundt programvarepatenter.

Et eksempel er den velkjente Amazon 1-Click-saken, hvor patent riktignok ikke ble gitt<sup>57</sup>. Formålet her var at dersom en kunde hadde registrert seg ved tidligere kjøp, så kunne han eller hun gjennomføre en ordre med bare ett klikk. Dette synes, til og med for ikke-datakyndige, som en ganske enkel og åpenbar løsning. BoA var enig og sa at dette var et resultat av et ønske om å forbedre kundeopplevelsen, og det forelå ingen oppfinnelseshøyde. I dag er problematikken rundt oppfinnelseshøyde svært aktuell da fremgangsmåten ved vurderingen av kombinasjonssøknader setter dette vilkåret i sentrum.

---

<sup>56</sup> Se Hilty Reto M. og Geiger Christophe, Towards a new instrument of protection for software in the EU? Learning the lessons from the harmonization of software patentability, i Arezzo Emanuela og Ghidini Gustavo (2011). Biotechnology and Software Patent Law – A Comparative Review of New Developments. UK: Edward Elgar Publishing, s 169-170

<sup>57</sup> T 1244/07. Patent ble gitt i USA, men avvist av BoA. Generelt skal man imidlertid være forsiktig med å vurdere patenter i ettertid fordi mye ser og høres åpenbart ut når det først er oppfunnet.

### 3.3. Programvarepatenter vil hindre oppstart og utvikling av små-og-mellomstore bedrifter

Dette argumentet bygger på at det er vanskelig og dyrt for små-og-mellomstore bedrifter (Small-and-medium Enterprises - SME) å patentere sine oppfinnelser, og å håndheve patentene dersom de først har fått dem<sup>58</sup>. Samtidig er de utsatt for inngrepssøksmål ved utviklingen av ny programvare. Dette fremstår i seg selv som et viktig argumentet, og Leith møtte ofte på uttrykket "patents is for the big boys" da han skrev sin bok<sup>59</sup>.

Tilhengere av programvarepatenter mener likevel at patenter er den eneste måten SME kan beskytte seg selv og sine produkter. Det er derfor foreslått tiltak for å gjøre de økonomiske utfordringene mindre. Et eksempel er å ha patentforsikringer, slik at eventuelle inngrepssøksmål blir mindre kostbare<sup>60</sup>.

I forbindelse med dette kan såkalte Patenttroll nevnes<sup>61</sup>. Dette er selskaper som utelukkende baserer seg på å kjøpe opp patenter for å bruke disse aktivt opp mot andre bedrifter som de mener krenker disse. De driver da selv ikke med utvikling, men baserer inntektene sine utelukkende på lisenspenger. Dette er ikke ulovlig, men bidrar til å vanskeliggjøre innovasjon<sup>62</sup>. Det er særlig SME som er i faresonen og trussel om søksmål vil være skremmende slik at de ofte velger å betale<sup>63</sup>. Hvor stort problemet er i Norge og Europa er uklart, og Digitutvalget konkluderer med at det bør foretas ytterligere undersøkelser for å kartlegge dette.

---

<sup>58</sup> Dette var et av hovedargumentene som fremkom i "The European Commission Consultation Exercise on The Patentability of Computer Implemented Inventions" fra 2002. Merk imidlertid at undersøkelsen er preget av en sterk mobilisering av anti-patent bevegelsen noe som kan ha påvirket resultatene

<sup>59</sup> Se Leith (2007) side 85

<sup>60</sup> Forslag fra Danmark og daværende leder av det danske Patentkontoret Henrik Dahl Sørensen i 2002:

<http://www.timeshighereducation.co.uk/story.asp?storyCode=172447&sectioncode=26>. (Lenke innhentet:27.06.2013)

<sup>61</sup> Se NOU 2013:2 (Digitutvalget) punkt 4.9.2

<sup>62</sup> Samme som over punkt 4.10 siste avsnitt hvor Digitutvalget uttaler at "Norske myndigheter bør signalisere motstand mot patenttroll"

<sup>63</sup> Samme som over punkt 4.9.2 avsnitt 13

Et interessant perspektiv gis også av den britiske programmereren Paul Graham i hans essay "Are Software Patents Evil?" fra 2006<sup>64</sup>. Her ytrer han meningen om at mindre bedrifter, og da særlig bedrifter i oppstartsfasen ikke behøver å frykte programvarepatenter. Større selskaper er normalt ikke interessert i å saksøke mindre selskaper for patentinngrep, særlig fordi de sjeldent har noe kapital. Dersom det skulle bli aktuelt betyr det ofte at de ser på dem som en konkurrent, og resultatet er da ofte at de heller kjøper dem opp. Igjen ser en gode argumenter på hver side av debatten, slik at det fremstår som noe uklart hvor stort problem dette er.

### 3.4. Er kritikken mot programvarepatenter holdbar?

Det som i hovedsak merkes når det gjelder kritikken er at mye av den ikke bygger på etterprøvbar forskning, og at det synes å være vanskelig å klargjøre hvor reelle problemene er. Som vist til ovenfor har forskere slitt med å komme til en konklusjon når det gjelder de positive og negative virkninger av programvarepatenter<sup>65</sup>. Å lese både pro- og contra-argumentene med et kritisk blikk er da helt nødvendig.

For en ikke-fagmann innenfor økonomi og teknologi er det og fristende å se til USA hvor programvarepatenter gis. Tilsynelatende har dette ikke vært et problem for utviklingen og veksten av industrien, noe som vises ved at syv av de 10 største programvarebedriftene i verden er lokalisert i USA<sup>66</sup>.

Samlet synes ComputerWorld sin overskrift i fra juni 1968 fortsatt å stå seg: "First patent is issued for software. Full implications are not yet known"<sup>67</sup>. Denne kom etter at det første programvarepatentet i USA ble gitt, og når en analyserer kritikken overfor er man over fire tiår etter fortsatt usikker på hvilken effekt programvarepatenter faktisk har. Dette gjør den pågående analysen desto viktigere, da økt synliggjøring av

---

<sup>64</sup> <http://www.paulgraham.com/softwarepatents.html>. (Lenke innhentet:27.06.2013)

<sup>65</sup> Siste i rekken er Digiutvalget i punkt 4.10 nest siste og siste avsnitt

<sup>66</sup> [http://www.forbes.com/global2000/#p\\_1\\_s\\_a0\\_Software%20&%20Programming\\_All%20countries\\_All%20states](http://www.forbes.com/global2000/#p_1_s_a0_Software%20&%20Programming_All%20countries_All%20states). (Lenke innhentet:27.06.2013)

<sup>67</sup> Amerikansk datamagasin: <http://en.wikipedia.org/wiki/Computerworld>. (Lenke innhentet:27.06.2013)

verdimessige spenninger kan bidra til større grad av klarhet rundt utfordringene med programvarepatenter.

## 4. Programvare(relaterte)patenter

### 4.1. Veien videre

Det vil nå først ses noe nærmere på de to relevante materielle patenterbarhetsvilkår. Disse utgjør den rettslige kjernen i analysen. Deretter vil spørsmålene omkring rettstilstanden med det tilhørende spørsmål om optimalisering behandles. Denne delen vil følge den to-delingen som ble skissert innledningsvis ved at det først vil ses på rettstilstanden slik den rent faktisk foreligger, før det ses på optimaliseringspotensialet. Særlig under den sistnevnte del blir da det som er skrevet under punkt 2 og 3 viktig.

Avslutningsvis i punkt 5 vil det ses på i hvilken grad det optimaliseringspotensial som er påpekt under punkt 4.3 flg. kan gjennomføres i praksis, og årsakene til at rettstilstanden ikke er optimalisert tidligere og hvorfor det også kan bli vanskelig i fremtiden. I så måte er det tale om en form for realitetsorientering.

### 4.2. De relevante materielle patenterbarhetsvilkår

#### 4.2.1 Oppfinnelse – Artikkel 52 EPC og Patentloven § 1(1) og (2)

Før Patentloven ble vedtatt ble det foretatt en felles nordisk utredning hvor det sentrale i oppfinnelsesvilkåret ble fastslått å være den tekniske karakter, den tekniske effekt og at oppfinnelsen er reproduserbar<sup>68</sup>. Et eksempel for å belyse dette er at det å skrive en bok ikke er en oppfinnelse, men at boktrykkerkunsten var det. Dermed er

---

<sup>68</sup> NU 1963:6 s. 96 sjette avsnitt og s. 97 andre, tredje og syvende avsnitt

oppfinnelsesvilkåret klarere definert i norsk rett enn under EPC hvor en definisjon mangler. Imidlertid er det gjennom EPOs praksis klargjort at det kreves "technical character". Dette kan nå også sies å finne støtte i ordlyden til artikkel 52 EPC etter at "all fields of technology" ble føyet til som følge av harmoniseringen med TRIPS<sup>69</sup>.

I teknisk effekt ligger det ganske enkelt at oppfinnelsen må løse det objektive problem den tar sikte på. Dette har nær tilknytning til patenterbarhetsvilkåret om industriell utnyttelse siden en oppfinnelse ikke kan utnyttes dersom den ikke faktisk løser problemet. Særlig innenfor bioteknologi og farmasi kan dette volde problemer da det her ofte kan være vanskelig å påvise den positive effekt. I praksis må det nasjonale patentkontoret stole på søkers fremstilling av effekten<sup>70</sup>. Kravet til reproduksjon henger tett sammen med dette, ved at utnyttelsen må kunne reproduseres med samme resultat gjentatte ganger. Med andre ord må ikke resultatet være tilfeldig.

I norsk rett, noe som også vil kunne legges til grunn under EPC, er teknisk karakter definert slik at det objektive tekniske problem som oppfinnelsen skal løse må bli løst "ved hjelp af naturkræfter, d.v.s. ved en lovbundet udnyttelse af naturens materie og energi"<sup>71</sup>. I utnyttelse ligger kjernen, ved at dette vil avgrense mot rene erkjennelser eller oppdagelser som er unntatt både etter EPC og Patentloven. I den pågående analysen er det særlig teknisk karakter som blir sentralt, da de to andre krav sjeldent volder problemer ved programvare.

#### 4.2.2. Oppfinneshøyde – Artikkel 56 EPC og Patentloven § 2(1)

Vilkåret innebærer som nevnt at oppfinnelsen må skille seg vesentlig fra det som var kjent før patentsøknaden ble inngitt. I ordene til EPC artikkel 56 så må ikke oppfinnelsen

---

<sup>69</sup> Spørsmålet om oppfinnelse krever teknisk karakter i EPC er problematisert av noen forfattere, men den gjengse oppfatning er at teknisk karakter er et krav, se Beresford (2000) punkt 2.01, Paterson, Gerald (2001). The European Patent System – The Law and Practice of the European Patent Convention. UK: Sweet and Maxwell s 404 og Visser, Derek (2009). The Annotated European Patent Convention. UK: H. Tel Publisher, s.56

<sup>70</sup> Se Ot.Prp 36 (1965-1966) s.19 for medisiners vedkommende i Norge

<sup>71</sup> NU 1963:6 S 96 sjette avsnitt

være "obvious for the person skilled in the art". I praksis benyttes ofte "the problem-and-solution approach" i vurderingen<sup>72</sup>. Vurderingen må først fastslå den nærmeste teknikks stilling på søknadstidspunktet, deretter finne det objektive tekniske problem som skal løses. Til slutt må det vurderes om løsningen som fremstilles ville vært nærliggende for den relevante fagmann dersom han hadde blitt stilt overfor samme problem<sup>73</sup>. Som dette viser er det tale om en utpreget skjønsmessig vurdering. Et av hovedankepunktene mot å tillate patentering er nettopp at denne vurderingen er vanskelig ved programvare, jf punkt 3.2.

### 4.3. IBM-sakene

#### 4.3.1. Kravet til ytterligere teknisk effekt

IBM-sakene regnes for å være de mest sentrale når programvarepatenter skal vurderes<sup>74</sup>. De representerer i dag ikke den nyeste utviklingen, men var avgjørende i å sementere praksisen som fortsatt er relevant.

IBM I introduserte kravet til ytterligere teknisk effekt for første gang. Saken gjaldt spørsmålet om et dataprogram som inneholdt en metode for å gjenopprette systemressurser når det ble kjørt på datamaskinen, og spørsmålet om et programvareprodukt lagret på en datamaskin kunne patenteres. BoA foretok en grundig tolkning av "as such"-kravet og kom frem til at det er dataprogrammer som ikke har noen teknisk karakter som er unntatt patentering<sup>75</sup>. Dette samsvarer med den forståelsen av oppfinnelsesvilkåret som er lagt til grunn ovenfor i punkt 4.2.1.

---

<sup>72</sup> Se Seville Catherine (2009). EU Intellectual Property Law and Policy. UK: Edward Elgar Publishing Limited, s.114 og Patentretningslinjene Del C punkt 5.5. Metoden er imidlertid ikke den eneste som kan brukes, noe som fremgår av Seville side 112 og T 465/92 punkt 9.2

<sup>73</sup> Se nærmere om dette i sak fra Patentstyrets annen avdeling kj. 7196 av 26.mars 2004, og samme i T 0694/92 punkt 28.5

<sup>74</sup> IBM I og T 0935/97 (IBM II). I det følgende vil det ses til IBM I da sakene rettslig sett er identiske

<sup>75</sup> IBM I Punkt 5.3



Bakgrunnen for BoA sin tolkning i IBM vises av følgende resonnement: Ethvert dataprogram produserer en teknisk effekt når det anvendes, typisk at det interagerer med maskinvaren. Dette kan ikke være tilstrekkelig fordi det ville lede til at all programvare ville unngått unntaket, og da gjort unntaket innholdsløst. BoA konstruerer dermed kravet om "ytterligere teknisk effekt"<sup>76</sup>. En må altså se hva programmet gjør *i tillegg til* den normale interaksjon med maskinvaren. Først da kan en si at programmet ikke "bare utgjør" et program for en datamaskin.

BoA utfyller ikke kravet med særlig mer enn at det må være noe mer enn denne vanlige interaksjonen mellom programvaren og maskinvaren. Noen uttalelser har imidlertid generell verdi. *For det første* setter BoA likhetstegn mellom direkte og indirekte effekt. Med dette menes at et dataprogram i seg selv ikke har noen teknisk effekt, "it only discloses the effect when being run and consequently only possesses the "potential" to produce said effect"<sup>77</sup>. Dette er en åpenbar, men viktig presisering. Dersom et dataprogram, når det kjøres, blir funnet å ha en ytterligere teknisk effekt er det ingen grunn til at programmet ikke skal kunne patenteres selv om det ikke kjøres akkurat nå. Det har, som BoA presiser, *potensiale* til å produsere effekten. *For det andre* fremkommer det at det ikke er et krav at den tekniske effekten er av en ytre fysisk karakter. Det uttales i punkt 6.5 at "a technical effect of that kind is achieved by the internal functioning of a computer itself under the influence of said program". At effekten kan skje internt i datamaskinen er senere bekreftet i flere andre saker, se for eksempel Garbage Collector saken som gjennomgås nedenfor. *For det tredje* fremgår det at den tekniske effekt gjerne kan være kjent fra før noe som har sammenheng med fravikelsen av "the technical contribution approach"<sup>78</sup>. Det tekniske bidrag, og da om effekten er kjent fra før, blir istedenfor et tema under oppfinneshøydevurderingen. Nedenfor under punkt 4.5 skal en særlig se at dette skaper uklarheter i forholdet mellom oppfinnelse- og oppfinneshøydevilkåret.

---

<sup>76</sup> Engelsk: Further Technical Effect

<sup>77</sup> IBM I punkt 9.4

<sup>78</sup> Technical Contribution Approach var rettstilstanden før IBM-sakene som fremgikk av T 208/84 (Vicom). Da denne rettstilstanden er forlatt av EPO, og i norsk rett, vil den ikke analyseres nærmere. Det kan likevel nevnes at den fortsatt danner utgangspunktet i Storbritannia, se EWCA Civ 1371 (Aerotel).

Et siste forhold som gjør IBM-sakene interessante og kontroversielle er punkt 9.8 hvor BoA uttaler at det er

”illogical to grant a patent for both a method and the apparatus adapted for carrying out the same method (*Se Vicom fotnote 78*), but not for the computer program product, which comprise all the features enabling the implementation of the method and which, when loaded in a computer, is indeed able to carry out that method”.

Dette forstås slik at BoA åpner for at det kan tas patent på programvare *alene*. Dette er en kontroversiell tolkning, og synes isolert sett å være i strid med ordlyden. Den utelukker nettopp patent på det som ”bare utgjør” programvare. Imidlertid kreves det som nevnt en ytterligere teknisk effekt. Det vil i prinsippet gjøre programvaren til noe mer enn kun programvare slik at det i så måte vil være i samsvar med ordlyden.

Kravet til ytterligere teknisk effekt er i dag innarbeidet i praksis, og det synes dermed avgjørende å se nærmere på dets innhold. Deretter vil en se at selv om EPO introduserte dette vurderingstema, så er det fortsatt et åpenbart behov for optimalisering, jf 4.3.3 nedenfor.

#### **4.3.2. Ytterligere Teknisk Effekt: Kasuistikk eller system?**

Årsaken til dette spørsmålet er at det, utover de tre forhold som er nevnt ovenfor, er uklart hva som faktisk innebærer en ytterligere teknisk effekt. Det er en iboende svakhet ved tolkningen at hva som utgjør noe mer enn vanlig teknisk karakter er krevende å forstå. Det vil dermed være interessant å se nærmere på om kravets innhold primært bygger på kasuistikk, altså en selvstendig vurdering i hver enkelt sak, eller hvorvidt det finnes mer generelle holdepunkter.

Et eksempel er saken i T 2048/07. Saken gjaldt en datamaskin-implementert metode som skulle forbedre et datasystem. Det ble påstått at dataprogram som mottar input fra brukeren som det ikke forstår kan slutte å respondere, hvor metoden skulle være med

på å forbedre slike problem. I tillegg ble noen type data lest fra en fil istedenfor fra kildekode noe som kan øke hastigheten på programvaren. Det ble påstått at dette gjorde programmet mer fleksibelt og portabelt, slik at den tekniske effekten av programmet gikk utover det normale.

BoA var imidlertid ikke enig. De viser blant annet til "a properly written program expecting a ten-digit customer number will not crash if presented with a twenty-digit one, it will simply reject the input". Hvorvidt det forelå noen ytterligere teknisk effekt var derfor uklart. Når det gjaldt den noe særegne måten å lese data på så var dette kun en nødvendig del av programmet, og det ble uttalt at "a correct program is precisely as flexible as it is designed to be". Derfor var ikke dette noen grunnlag for å si at det forelå en ytterligere teknisk effekt. Samlet sett var det derfor et dataprogram "as such".

T 0979/06 (Bildebehandlingsaken) er et eksempel hvor det forelå en ytterligere teknisk effekt. Patentsøknaden gjaldt et dataprogram som skulle redusere støy på videoopptak når de ble avspilt på datamaskinen. Dette ble akseptert uten at BoA gir noen nærmere begrunnelse. Årsaken må antas å være at støyreduksjonen på bildene var en forbedring av tilstanden før programmet ble introdusert, slik at det gikk utover den vanlige interaksjon mellom programvare og maskinvare.

Et siste eksempel er T 0121/06 (Garbage Collector) som blant annet omhandlet spørsmålet om et programvarepatent. Det er et alminnelig problem på en datamaskin, eller datamaskinlignende produkter som en mobiltelefon, at når et program lukkes så vil det fortsatt holde på minne. Dette gjør at maskinen arbeider tregere. Programmets funksjon var at når det ble kjørt så frigjorde det minne som lukkede programmer beslagla. BoA så dette som en ytterligere teknisk effekt "since it serves to free memory space which would otherwise be unnecessarily blocked and hence modifies the internal functioning of the computer itself".

De to siste sakene viser at kravene ikke nødvendigvis er strenge. Det må ikke være noe kvalifisert ytterligere effekt. Både i Bildebehandlingsaken og Garbage Collector løste programvaren et spesifikt problem, uten at problemet eller løsningen nødvendigvis var

komplisert. Problemet i den førstnevnte sak var nettopp at det var usikkert om det i det hele tatt forelå et problem som måtte løses.

Dermed skal det nok ikke mer til for å tilfredsstille kravet til ytterligere teknisk effekt enn at programmet har en særlig funksjon som er noe mer enn den isolerte tekniske effekt som alltid oppstår mellom programvare og maskinvare. Videre i T 26/86 (Røntgenapparatsaken) uttalte BoA at “when the technical effect occurs is irrelevant (...) The only fact of importance is that it occurs at all”. I så måte synes det heller ikke å stilles krav til når i prosessen den ytterligere tekniske effekt inntreffer.

Samlet sett synes likevel det nærmere innholdet i kravet primært å hvile på en kasuistisk tilnærming, men med visse generelle holdepunkter. Noen av disse følger allerede av IBM-sakene, mens andre har kommet til senere. Felles er imidlertid at de i liten grad er eksplisitt formulert, men må hentes ut av hver enkelt sak. I det følgende vil det vises at dette skaper utfordringer i relasjon til kravene til den juridiske argumentasjon og optimaliseringen av rettstilstanden.

#### **4.3.3. Identifikasjon av spenninger i relasjon til kravet om ytterligere teknisk effekt som grunnlag for optimalisering av rettstilstanden og den juridiske argumentasjon**

Med introduksjonen av kravet til ytterligere teknisk effekt ble rettstilstanden klarere enn den noen gang hadde vært. Problemet er imidlertid som nevnt at innholdet i beste fall er uklart, slik at en langt på vei synes å måtte falle tilbake på kasuistikk. For en patentsøker er dette en vanskelig situasjon. Ved å søke om patent vil oppfinnelsen som hovedregel offentliggjøres, noe som kan skje selv om søknaden til slutt avslås<sup>79</sup>. Avgjørelsen om å søke patent vil da ofte baseres på om søknaden forventes å innvilges, og dersom søkeren mener at dette er usikkert kan det for han lønne seg å holde oppfinnelsen hemmelig. Hemmelighold av oppfinnelser er noe patentsystemet søker å unngå, da det ikke stimulerer til innovasjon.

---

<sup>79</sup> Se Patentloven § 22(1) og (2) og EPC artikkel 93, jf Implementing Regulations to the Convention on the Grant of European Patents rule 67 (2). En patentsøknad offentliggjøres som hovedregel etter 18 måneder, og dersom ingen endelig avgjørelse er tatt skjer dette selv om søknaden til slutt skulle bli avslått

Det er dermed grunnlag for å diskutere hvorvidt det er tilstrekkelig grad av koherens mellom kravet til ytterligere teknisk effekt og grunnleggende krav til juridisk argumentasjon. Her er det særlig tale om behandling av spenninger til forutberegnelighet, da ytterligere teknisk effekt fremstår som en skjønnsmessig vurdering uten klare holdepunkt.

En kommer ikke unna skjønnsmessige avveininger i rettsanvendelsesprosessen, men det er viktig at disse i størst mulig grad er etterprøvbare. Etterprøvbarhet skapes ved synliggjøring av de underliggende faktorer når avgjørelser tas. I denne sammenheng er det interessant når Lilleholt sier at "som kjent er ein ofte minst halvvegs til svaret når ein har stilt spørsmålet rett"<sup>80</sup>. Dersom dette ses i lys av etterprøvbarhet vil det kunne argumenteres for at det sentrale for EPO er å skape kriterier som gjør at en kan stille et mest mulig presist spørsmål, da dette igjen vil kunne bidra til en presis vurdering. Det vil igjen lede til mer forutberegnelighet, både i selve vurderingen og utfallet av denne.

Denne spenningen er ikke problematisert. Dette har nok sammenheng med at EPO selv ikke har helt klart for seg hva som ligger i kravet, i tillegg til at de ikke ønsker å binde opp vurderingen. Den enkleste måte å løse dette på synes likevel å være at BoA stiller opp vurderingsmomenter. Vurderingsmomentene vil gi patentsøkere noe mer håndfast å forholde seg til, samtidig som at de kan utdypes og tilpasses av praksis over tid. Ved å foreta en slik operasjon så vil BoA ha "et aktivt forhold til legitimitetsintervallet"<sup>81</sup>. Med dette menes at de tar tak i den spenning som foreligger mellom kravet til ytterligere teknisk effekt og det grunnleggende hensyn til forutberegnelighet, og aktivt tar stilling for å løse opp denne. Vurderingsmomentene vil i tillegg bidra til at BoA kan spisse problemstillingen, altså spørsmålet, inn på det som er konkret problematisk slik at det blir enklere å komme frem til et korrekt resultat.

En ser at BoA i IBM-sakene kun synes å komme halvveis til målet. De skapte et vurderingstema, som var et skritt fremover, men synes ikke å ha fullført dette ved å presisere dets innhold. Når det nå skal ses på utviklingene i etterkant av IBM-sakene blir

---

<sup>80</sup> Lilleholt Kåre, Bruk av reelle omsyn i formueretten, Jussens Venner 2000, s.50 andre avsnitt

<sup>81</sup> Tande (2011) punkt 4.3.1 andre avsnitt

det sentralt at praksis, istedenfor å skape vurderingsmomenter, heller synes å ha skapt en alternativ fremgangsmåte.

#### 4.4. Utviklingen i etterkant av IBM-sakene

##### 4.4.1. Microsoft-saken

Selv om IBM-sakene fortsatt danner utgangspunktet har det skjedd en utvikling av rettstilstanden som fører til mer usikkerhet. Dette var og årsaken til at Presidenten av EPO brukte sin rett etter EPC artikkel 112 (1) B, og stilte spørsmål til EBoA om rettstilstanden rundt programvarepatenter. Disse spørsmålene resulterte i Opinion G 03/08 (Heretter "Opinion 2010") hvor EBoA tar særlig stilling til utviklingen etter IBM.

En sak som videreutvikler IBM-sakene er Microsoft-saken som omhandlet en metode for å flytte data, hvor et av kravene gjaldt denne metoden som var lagret på "a computer readable medium (*typisk en harddisk*) having computer-executable instructions on it to cause the computer system to perform the said method" <sup>82</sup>.

Det interessante med avgjørelsen er at BoA uttaler i punkt 5.3 at kravet har teknisk karakter "since it relates to a computer readable medium". Selv om BoA går videre og taler om ytterligere teknisk effekt i relasjon til metoden, var det at programmet var lagret på et medium *i seg selv* nok. Dette er å gå et skritt videre fra IBM-sakene hvor BoA avviser at dette er tilstrekkelig. Der kreves det at den ytterligere tekniske effekt må være i programmet som sådan dersom det er der den underliggende ideen til oppfinnelsen faktisk ligger<sup>83</sup>.

---

<sup>82</sup> Microsoft-saken, sammen med T931/95 (Pension Benefits) og T 258/03 (Hitachi), omtales ofte som "the Trilogy" på grunn av deres nære sammenheng i å etablere den rettstilstand som nå omtales, kalt "the any hardware approach". Som det imidlertid vil fremgå i det følgende er ikke disse tre sakene helt like i sine uttalelser. I tillegg til at Duns-saken, som presenteres nedenfor, også kommer inn.

<sup>83</sup> IBM I Punkt 9.3

I IBM-saken blir det som nevnt lagt til grunn at ytterligere teknisk effekt skal vurderes uavhengig av om effekten er en del av den tidligere teknikk, altså slik at både ny og gammel teknikk er relevant. Videre blir det også uttalt at det har ingenting å si om et program blir forsøkt patentert alene eller lagret på et datamaskinlesbart medium, det er den ytterligere tekniske effekt som er avgjørende. Imidlertid er dette internt motstridende, noe som kan illustreres med et eksempel.

Dersom en har et datamaskinlesbart medium, for eksempel en type USB-pinne, så vil den ha teknisk karakter. Hvis et dataprogram er lagret på denne pinnen er det derfor ikke et dataprogram "as such", men "a computer-readable storage medium storing computer program X". Dette er ikke unntatt fra patentering etter EPC artikkel 52(2) og (3) fordi USB-pinnen skaper en teknisk karakter, selv om den består av tidligere kjent teknologi. Dermed er det intet krav om en "ytterligere" teknisk effekt i programvaren. Resultatet er da at det får betydning om programmet er lagret på et datamaskinlesbart medium eller ikke<sup>84</sup>.

Spørsmålet blir da om BoA i IBM-saken mente å åpne for en rettstilstand hvor resultatet i Microsoft-saken var mulig. I Microsoft-saken skaper BoA, bevisst eller ubevisst, en situasjon hvor utformingen av patentkravene er avgjørende og ikke realitetene. Det synes slik at BoA i IBM-sakene forsøkte å unngå en slik situasjon da de der setter selve programvaren i fokus. Videre synes særlig den klare uttalelsen i IBM I om at det å lagre et program på et medium ikke er nok å være et argument for at BoA ikke ønsket å skape en åpning for en slik rettstilstand.

Sett i dette lys representerer imidlertid ikke Microsoft-saken en ren endring av rettstilstanden, men snarere en utvikling av en alternativ fremgangsmåte<sup>85</sup>. Dersom programvaren er lagret på et lagringsmedium eller lignende kreves ikke en ytterligere teknisk effekt i selve programvaren. Det er tilstrekkelig at mediet gir den tekniske effekt.

---

<sup>84</sup> Eksempelet er forfatterens eget, men bygger på tankene i Opinion 2010 side 36 til 37

<sup>85</sup> I Opinion 2010 foretar EBoA en analyse hvor resultatet er at Microsoft-saken ikke er en rettslig fravikelse av IBM-saken i lys av EPC artikkel 112(1) B, men det uttales at et rent faktisk avvik foreligger, se punkt 10.8 og 10.12

Dette innebærer at unntaket tolkes snevert, ved at det virkelig *kun* er det som bare utgjør programvare som fanges opp av bestemmelsen<sup>86</sup>.

Det kan imidlertid diskuteres hvorvidt tolkningen av unntaket er *for* snever. At programvare som utgjør en oppfinnelse skal kunne patenteres er hevet over tvil, men at *enhver* teknisk karakter skal lede til at unntaket ikke får anvendelse er ikke like klart. Ordlyden i artikkel 52 (3) EPC knyttes til det aktuelle "subject-matter" noe som taler for at det er selve programvaren som må ha teknisk karakter. BoA tar i IBM-sakene hensyn til ordlyden i EPC i den forstand at det sentrale der var hvorvidt *programvaren* ga en ytterligere teknisk effekt. Dagens rettstilstand viser at det ikke er noe krav, det er tilstrekkelig at programvaren er lagret på et medium eller lignende som har det. Denne snevre tolkningen av programvareunntaket skaper særlige utfordringer i relasjon til patentsystemet og den juridiske argumentasjonen, og det skal derfor ses nærmere på i det følgende.

#### 4.4.2. Identifikasjon av spenningene i Microsoft-saken som grunnlag for optimalisering av rettstilstanden og den juridiske argumentasjon

Den snevre tolkning av programvareunntaket som foretas i Microsoft-saken synes umiddelbart å harmonere godt med patentsystemets mål om å stimulere til innovasjon. Flere patentsøknader vil da få mulighet til å bli behandlet under de alminnelige vilkår, noe som i teorien reduserer risikoen for at potensielt patenterbare oppfinnelser ikke får en grundig vurdering som følge av unntaksbestemmelsen.

Det synes også å være på dette plan EPO begrunner standpunktet. I Hitachi-saken punkt 3.5. uttales det at en sentral årsak til at kombinasjonssøknader alltid aksepteres som en oppfinnelse er "simply that the technical features may in themselves turn out to fulfil all requirements of article 52 (1) EPC". Med andre ord er poenget at søknader som inneholder programvare lagret på et medium alltid bør regnes som en oppfinnelse, fordi mediet i seg selv kan inneha oppfinneshøyde selv om programvaren ikke gjør det.

---

<sup>86</sup> At unntaket skal tolkes snevert ble bekreftet allerede i G05/83 (1984)



Dette fremstår ikke som en urimelig begrunnelse ut i fra tanken om at patentsystemet ikke skal hindre for patent på teknikk som innebærer en nyvinning.

Ytterligere støtte for det ovenstående kan finnes ved en nærmere analyse av EPC og bakgrunnen for programvareunntaket. Et sentralt formål ved inkluderingen av "as such" var å sørge for en snever tolkning av unntakene. I forlengelsen av dette fremgår det av diskusjonene før vedtagelsen av EPC at det var et sterkt ønske om å ikke unnta fra patent virkelige oppfinnelser, selv om programvare var involvert<sup>87</sup>. Dette har og fått uttrykk i praksis hvor EPO har skrevet at det er et mål å "promote technical progress by giving proper protection to (...) inventions"<sup>88</sup>. I tillegg ble programvareunntaket mer eller mindre direkte adoptert fra Patent Cooperation Treaty uten at det nærmere innhold ble nøye gjennomgått<sup>89</sup>. Målet om å få en global rettsenhet var det sentrale. EPO kan sies å ha fulgt dette målet ved at den snevre tolkning nettopp bidrar til en lignende rettstilstand som i USA og Japan<sup>90</sup>. Samlet sett er det sammenfall mellom EPO sin tolkning og det underliggende formålet til artikkel 52 EPC, noe som viser at den tolking som legges til grunn i Microsoft har støtte.

Imidlertid kan dette problematiseres på et mer grunnleggende nivå. Som nevnt under punkt 3.1 er det usikkert hvorvidt programvarepatenter stimulerer til innovasjon, og dermed bidrar positivt samfunnsøkonomisk. En av årsakene er at programvare ofte utvikles med små steg, noe som betyr at programvare har en kumulativ natur ved at ny programvare bygger på eldre programvare<sup>91</sup>. Et eksempel på dette er Microsoft Windows. Helt i fra Windows 1.0 i 1985 og opp til Windows 98 i 1998 var kjernen bygget rundt MS DOS<sup>92</sup>. Den kumulative naturen kan sies å være et insentiv for første

---

<sup>87</sup> Se nærmere om dette Beresford (2000) punkt 1.56 – 1.62

<sup>88</sup> IBM I punkt 10.2

<sup>89</sup> Se Sterckz, Sigrid og Cockbain, Julian (2009). The Patentability of Computer Programs in Europe: An Improved Interpretation of Articles 52 (2) and (3) of the European Patent Convention med henvisninger til Minutes of the meeting of Working Party I (Luxeumbourg Conference, 8-11 July 1969), <http://onlinelibrary.wiley.com/doi/10.1111/j.1747-1796.2009.00390.x/full>. (Lenke innhentet:27.06.2013)

<sup>90</sup> Se IBM I punkt 2.5 og 2.6

<sup>91</sup> De aller fleste oppfinnelser vil til en viss grad være av kumulativ natur ved at de bygger på tidligere ideer.

<sup>92</sup> Microsoft Disk Operating System, <http://en.wikipedia.org/wiki/MS-DOS>

generasjon av oppfinnere, men en byrde for den neste<sup>93</sup>. Årsaken er at den første oppfinneren skaper selve grunnlaget som senere programvare baserer seg på, og dermed kan tjene på at andre skaper nye versjoner eller annen programvare basert på dette. For neste generasjon vil det kunne være et problem blant annet fordi de kan måtte betale lisenspenger eller bli hindret tilgang til teknologi, noe som vanskeliggjør utvikling av nye produkter. Innenfor programvare hvor utviklingen går raskt er dette et særlig problem fordi eldre, ofte utdaterte, patenter kan hindre utvikling av ny programvare.

En ser da at det er grunnlag for optimalisering også på dette punkt, og at det er en tydelig spenning opp mot patentsystemet som sådan. I den konkrete rettsanvendelsen kan det tale for at unntaket ikke bør tolkes så snevert som det gjøres i dag. I relasjon til den juridiske argumentasjon er det i alle fall en klar oppfordring til EPO om å synliggjøre problematikken i drøftelsen, for på den måten å få frem spenningen. Særlig i lys av at innovasjonshensynet er et av de mest grunnleggende i patentsystemet fremstår dette som svært viktig.

*I sammenheng med dette kan det kort påpekes at "kumuleringsproblematikken" er noe av årsaken til at en ofte ser at det argumenteres for gradert beskyttelsestid innenfor patentretten. Innenfor programvare er det av flere kommentatorer foreslått at det ikke bør gis mer enn 5 års beskyttelse, nettopp fordi programvare utvikles så hurtig, og at eldre patenter som i seg selv ikke lenger har noen markedsverdi kan hindre innovasjon<sup>94</sup>.*

En ytterligere utfordring, utover det snevre virkeområde programvareunntaket gis, er åpenheten ved argumentasjonen. Som nevnt tidligere bygger Microsoft på en rekke av saker som i alle fall kan spores tilbake til Pensions Benefit-saken<sup>95</sup>.

I ingen av disse sakene blir det imidlertid problematisert at det reelt sett skapes et alternativ til ytterligere teknisk effekt fra IBM-sakene. BoA har en klar oppfordring om å

---

<sup>93</sup> Se om dette: "The Economic Impact of Patentability of Computer Programs", side 29, [http://ec.europa.eu/internal\\_market/indprop/docs/comp/study\\_en.pdf](http://ec.europa.eu/internal_market/indprop/docs/comp/study_en.pdf). (Lenke innhentet:27.06.2013)

<sup>94</sup> Se blant annet artikkelen til Hilty og Geiger i Arezzo og Ghidini (2011) side 184

<sup>95</sup> Se fotnote 82

synliggjøre og begrunne dette avviket. Ikke bare i relasjon til kravene til den juridiske argumentasjonen, men også som følge av Rules of Procedure of the Boards of Appeal artikkel 20. Her fremgår det at dersom en tidligere avgjørelse eller forklaring skal fravikes, så skal det begrunnes. Som nevnt tidligere er det ikke sikkert at disse avgjørelsene utgjør en rettslig fravikelse fra IBM-sakene, men et rent faktisk avvik foreligger. Når et slikt faktisk avvik foreligger bør i alle fall spørsmålet om et rettslig avvik også foreligger drøftes eksplisitt. Det bemerkes ikke at det uttales i IBM at det er uten betydning hvorvidt programmet ønskes patentert alene, på en datamaskin eller et lagringsmedium selv om dette står i strid med det BoA i disse sakene legger til grunn. En synliggjøring av denne spenningen kunne bidratt til en mer grundig begrunnelse for synet på kombinasjonssøknader. Dette kunne også bidratt til å presisere kravet til ytterligere teknisk effekt fordi det ideelt sett burde vært foretatt en analyse av forholdet mellom disse to standpunkter. Denne manglende åpenheten fører til at den juridiske argumentasjonen ikke er optimal.

Ved kombinasjonssøknader ble det skapt en alternativ måte å komme seg rundt programvareunntaket på. Den snevre tolkning av unntaket som dette alternativet bygger på skaper en rekke utfordringer i relasjon til patentsystemet og kravene til juridisk argumentasjon. Samtidig har det også dannet grunnlaget for en interessant utvikling når det gjelder selve vurderingsmåten av slike saker.

## 4.5. Vurderingen av programvarepatentkrav

### 4.5.1. Har unntaket mistet all relevans?

Utfordringen som oppstår når EPO har foretatt en så snever tolkning er at unntaket synes å miste sin praktiske betydning<sup>96</sup>. Som EBoA bekrefter i punkt 10.13 i Opinion 2010 er det tilstrekkelig for å unngå unntaket "merely by explicitly mentioning the use of a computer or a computer-readable storage medium". En kan og spørre om unntaket noen gang har spilt en viktig rolle. Et argument for at det ikke er tilfelle er at Sveits, som

---

<sup>96</sup> Dette gjelder langt på vei alle unntakene knyttet til "as such" og ikke kun programvare

har vært medlem av EPO fra starten av, ikke har med unntaksbestemmelsen i sin nasjonale patentlovgivning uten at noen realitetsforskjell er tilsiktet<sup>97</sup>. Det betyr at en vil kunne komme til samme resultat ved å benytte seg av de alminnelige vilkår.

Praksis viser likevel at unntaket fortsatt har betydning, men på en annen måte enn først tenkt. Utviklingen som er beskrevet ovenfor under punkt 4.4 har ført til den situasjon at den virkelige vurderingen av unntaksbestemmelsen synes å ha flyttet seg fra unntaket i seg selv, og over til oppfinneshøydevurderingen. Det foretas en grundig avveining av hvilke elementer ved patentkravet som blir en del av vurderingen, og programvareunntaket får betydning ved at det danner retningslinjer for hvilke elementer som det kan ses til. Særlig interessant blir dette når det ses i lys av punkt 3.2 hvor det nettopp blir gjort rede for at oppfinneshøydevilkåret er særlig problematisk ved programvarepatenter.

Startpunktet for denne måten å vurdere programvarepatenter var Pensions Benefit-saken, men Hitachi-saken som bygger videre på denne, er bedre egnet til å illustrere fremgangsmåten. Saken gjaldt et forsøk på å få patent på en automatisk auksjonsmetode, hvor et av kravene gjaldt denne metoden som en del av programvare<sup>98</sup>. Auksjonsmetoden som ble ønsket patentert er lik den man ofte kjenner fra nettauksjoner hvor den som byr legger inn et bud og en makspris, og hvor budene blir oppdatert i sanntid.

Programvarepatentkravet blir vurdert i punkt 7 hvor det vises tilbake til vurderingen av det første patentkravet som gjaldt selve metoden<sup>99</sup>. BoA legger til grunn en meget vid forståelse av oppfinneshøydevilkåret som samsvarer med Microsoft-saken<sup>100</sup>. Under vurderingen av oppfinneshøyden uttaler BoA i punkt 5.3: "the invention will be

---

<sup>97</sup> Sveits så det som "überflüssig" å ha unntakene i artikkel 52(2) og (3) med. Se "Botschaft des Bundesrates an die Bundesversammlung über drei Patentübereinkommen und die Änderung des Patentgesetzes", 76.021, 24 Mars 1976, side 67

<sup>98</sup> Se Hitachi "Summary of Facts and Submissions" punkt V avsnitt 3.

<sup>99</sup> Det presiseres at programvaren her ikke ble vurdert opp mot unntaksbestemmelsen, det interessante er imidlertid i denne sammenheng måten BoA vurderer oppfinneshøydekravet.

<sup>100</sup> Det at oppfinnelsen inneholder en form for teknikk er nok i seg selv, se Hitachi punkt 4.7

assessed with respect to the requirement of inventive step by taking account of only those features which contribute to a technical character”.

BoA vurderte derfor først hvilke deler av patentkravet som skulle være med i vurderingen av oppfinneshøyde, hvor det som ikke hadde noen teknisk karakter ble fjernet. Auksjonsmetoden inneholdt flere tekniske elementer, som for eksempel lagring av data. Disse tekniske elementene var avgjørende da unntaksbestemmelsen i § 52(2) og (3) ikke kom til anvendelse. Likevel førte de ikke til at det forelå oppfinneshøyde, da ingen av dem gikk utover det som måtte fremstå som nærliggende for en fagmann.

EBoA bekrefter denne fremgangsmåten i Opinion 2010 punkt 10.13.1, og viser særlig til Duns-saken som bygger videre på Hitachi. Saken gjaldt ikke programvare spesielt, men tolkningen av ”bare utgjør”-vilkåret og oppfinneshøyde.

Det ble forsøkt patentert en metode som skulle estimere hvor mye salg en butikk hadde uten at butikken meldte inn konkrete salgstall. Denne metoden, som utgjorde hovedkravet, ble ikke ansett for å være en oppfinnelse. Imidlertid ble metoden, når den ble krevd som en del av et datasystem akseptert, da dette ga tilstrekkelig teknisk effekt.

Det interessante var igjen vurderingen av oppfinneshøyde<sup>101</sup>. Metoden, som en del av et datasystem, krevde en algoritme som var basisen for å estimere salgsaktiviteten. Algoritmen hadde imidlertid ingen teknisk karakter, og ble regnet som en del av forretningsmetoden som er unntatt fra patentering etter artikkel 52 (2) C. Dermed måtte denne ses bort fra ved vurderingen av oppfinneshøyde, og en stod da igjen med selve datasystemet som skulle implementere metoden, men systemet hadde ingen oppfinneshøyde – det var kun et alminnelig datasystem.

For å tydeliggjøre fremgangsmåten kan en gå tilbake til USB-pinne eksempelet som ble brukt ovenfor. Dersom en har programvare lagret på denne vil dette ikke være noe som ”bare utgjør” programvare. Men dersom programvaren som er lagret på USB-pinnen ikke går utover det å bare være programvare vil det måtte ses bort i fra når oppfinneshøyde vurderes, fordi det ikke har noen teknisk karakter etter artikkel

---

<sup>101</sup> Duns punkt 22-28

52(2) og (3). Dermed står en igjen med en standard USB-pinne som ikke har oppfinneshøyde.

De overnevnte sakene nedtoner oppfinnelsesvilkåret i artikkel 52, og dermed også programvareunntaket. Dette finner støtte i Hitachi-saken hvor det blir uttalt i punkt 4.7 at "it is (...) concluded that, in general, a method involving technical means is an invention within the meaning of Article 52(1) EPC". Altså sier BoA at når det søkes om patent på programvarerelaterte oppfinnelser og søknaden inneholder en eller annen form for teknikk, slik som et lagringsmedium, så er det en oppfinnelse.

På den ene side kan det argumenteres for at vurderingsmåten i alle fall viser at BoA tar konsekvensen av at en har en unntaksbestemmelse siden den får relevans under vurderingen av oppfinneshøyde. Det sentrale ved patentretten er nettopp å kunne patentere oppfinnelser, men oppfinnelser kan være som mangt, og vil utvikle seg i takt med teknologien. Derfor kan det være hensiktsmessig å ha hovedfokus på de andre materielle vilkår. Dette er løsningen i USA hvor en kun har de generelle vilkår, uten noen særskilte unntak<sup>102</sup>.

Likevel er det klart at det oppstår spenninger i relasjon til optimalisering av den juridiske argumentasjonen. Dette fordi det ikke er den naturlige måten å forstå Konvensjonen på. Kjernen ved oppfinneshøyde er om oppfinnelsen skiller seg vesentlig fra den tidligere teknikk, mens det i dag også foretas en slags vurdering av oppfinnelse under dette vilkåret. På den måten blir oppfinnelsesvilkåret, med programvareunntaket, bagatellisert.

En utfordring i nær sammenheng med dette er at det undergraver systemet i EPC, og da også i den norske Patentloven. For dem som søker om patent er forutberegnelighet avgjørende, men fremgangsmåten kan blant annet bidra til å skjule de egentlige vurderinger som foretas. Disse spenningene vil adresseres nærmere i det følgende.

---

<sup>102</sup> United States Code Title 35 – Patents, section 101

#### 4.5.2. Forholdet mellom oppfinnelse- og oppfinneshøydevilkåret: Identifikasjon av spenninger som grunnlag for optimalisering av rettstilstanden og den juridiske argumentasjon

Vurderingsmåten som følger av den brede aksepten av kombinasjonssøknader skaper uklarhet når det gjelder skillet mellom oppfinnelse og oppfinneshøyde. Ordlyden i artikkel 56 starter med "an invention shall be considered (...)". Altså forutsettes det at en oppfinnelse foreligger på dette stadiet. BoA er klar over dette og skriver i punkt 3.1 i Hitachi at "the verification that claimed subject-matter is an invention within the meaning of article 52(1) is in principle a prerequisite for the examination with respect to (...) inventive step". I tillegg følger det av Duns-saken punkt 10 at vurderingen av oppfinnelse må være "strictly separated from and not mixed up with" de andre materielle vilkår.

Problemet er imidlertid at oppfinnelse, selv om det isolert sett blir vurdert først, ofte vurderes svært overflattisk. Utfordringen blir at den overflatiske vurderingen under artikkel 52 gjør at oppfinneshøydevurderingen må starte med å luke bort de ikke-tekniske elementene. Sagt på en annen måte kan det sies at vurderingen av oppfinneshøyde må starte med å utføre oppfinnelsevilkårets funksjon. Isolert sett synes dette å være en harmløs modifikasjon av EPC, men et banalt poeng er at det som oftest er en hensikt bak oppbygningen til et juridisk system. Denne spenningen med EPC sitt system blir ikke adressert av BoA i noen av sakene som anvender fremgangsmåten.

Konsekvensen av det manglende fokus på disse spenningene kan være at risikoen for vilkårlighet øker. Det blir et sterkt fokus på oppfinneshøyde, slik at oppfinnelsevilkåret nedtones. Det kan da føre til at sentrale elementer overses eller fjernes fra vurderingen, noe som øker presset på hensynet til forutberegnelighet<sup>103</sup>. Klarere fokus på systemet, slik at oppfinnelsevilkåret med tilhørende programvareunntaket ikke blir nedtonet, hadde skapt en tydeligere vurdering hvor risikoen for vilkårlighet hadde blitt redusert.

---

<sup>103</sup>Se blant annet referansen til en EPO ansatt som sier at "it is too easy to throw too many things out of the basket" når fremgangsmåten i Hitachi og Duns anvendes, <http://ipkitten.blogspot.no/2012/06/on-threshold-of-dream-patents-and.html>

Dette kan holdes opp mot det ansvaret rettsanvenderen har til å være klar over at spenninger oppstår, og videre å synliggjøre og balansere disse<sup>104</sup>. Ideelt sett burde BoA sagt at de var klar over at vurderingsmetoden utfordrer systemet, og på den måten synliggjort spenningene. Det å respektere systemet vil ha stor verdi, også for rutinerne dommere, ved at det hjelper en til å ha fokus på de relevante elementene. Ofte kan det være lett å overse viktige detaljer når avgjørelser skal tas under tidspress, i tillegg til at elementer som kan ha en verdi kan ende opp med å bli fjernet fra vurderingen. I så måte kan det å i større grad følge EPC sitt system være en måte å sørge for tilstrekkelig diskursiv innsikt og ansvar, blant annet fordi en da vil få en mer grundig vurdering av oppfinnelsesvilkåret og programvareunntaket. Samlet viser dette at et aktivt forhold til legitimitetsintervallet og respekt for systemet i EPC påvirker hverandre. En får en sirkeleffekt ved at respekt for systemet gjør det enklere å ha tilstrekkelig diskursiv innsikt og ansvar, samtidig som diskursiv innsikt og ansvar bidrar til at systemet blir satt i fokus<sup>105</sup>.

Spenningen med hensyn til forutberegnelighet forsterkes av at de såkalte Trilogi-sakene, som sammen med Duns er med på å befeste denne fremgangsmåten, ikke er helt like i sine vurderinger. Særlig Microsoft avviker fra fremgangsmåten skissert i punkt 4.5.1 ved at det ikke er fokus på å fjerne det ikke-tekniske fra vurderingen av oppfinnelseshøyde. Dette vil for patentsøkerne skape uklarhet i hvordan vurderingen faktisk vil foregå, noe som igjen gjør det vanskeligere å forutsi utfallet. Som nevnt ovenfor vil søknaden uansett utfall kunne publiseres slik at dette kan føre til at det heller satses på hemmelighold.

Totalt sett synes dermed sakene som omhandler vurderingsmåten av programvarepatenter ikke i stor nok grad å ta konsekvensen av ansvaret som gjelder ved juridisk argumentasjon. Særlig synes økt fokus på EPC sitt system å være en god kilde til forbedring, da det også kan bidra til å redusere risikoen for vilkårlighet.

---

<sup>104</sup> Dette omtales som diskursiv innsikt og diskursivt ansvar, se Tande (2011) punkt 4.3.1 og Mæhle (2004) punkt 6 og 7

<sup>105</sup> Dette kan langt på vei løftes opp på et generelt nivå, ved at dette gjelder på alle rettsområder.



I nær sammenheng med det som nå er sagt dukker et annet spørsmål opp. Som en konsekvens av den snevre tolkingen av unntaket og at programvare ofte, som omtalt under punkt 4.4.2 ovenfor, utvikles i små steg er et det et spørsmål om oppfinnelseshøyde er et egnet vilkår ved vurderingen av programvarepatenter<sup>106</sup>. Problemet er at oppfinnelseshøyde innebærer at den nye oppfinnelsen må skille seg vesentlig fra den tidligere teknikk. Dette kan være en utfordring ved programvare nettopp fordi det kan være relativt små endringer fra tidligere programvare. Derfor vil det ofte være vanskelig å vurdere om oppfinnelseshøyde faktisk foreligger. Dette kan tale for at EPO bør være særlig kritisk ved vurderingen av programvarerelaterte patenter. Både ved at de anlegger en bredere forståelse av unntaksbestemmelsen og en strengere vurdering av oppfinnelseshøyde. På den måten kan en unngå at patenter gis i tilfelle hvor det faktisk ikke burde vært gitt.

Imidlertid må det påpekes at spenninger på det nivået som her blir adressert primært er lovgiver, og i EPC sitt tilfelle medlemsstatene, sitt ansvar. EPO må forholde seg til EPC slik den står, og må i stor grad bygge på forutsetningen om at en riktig tolking av programvarepatentunntaket i sammenheng med de materielle vilkår vil fremme patentsystemet formål<sup>107</sup>. Derfor kan ikke EPO gå inn for en strengere tolkning av oppfinnelseshøydevilkåret utelukkende basert på at de er usikre på hvorvidt vilkåret er egnet til å vurdere programvarepatenter. Målet om en størst mulig grad av verdi-og fornuftmessig koherent argumentasjon må for domstolene søkes innenfor dagens rammer. Dette fører til at en tilbakeholdenhet fra EPO med å gi patenter som involverer programvare på dette grunnlag vil være problematisk.

På den annen side handler de krav til juridisk argumentasjon som benyttes i denne analysen nettopp om at rettsanvenderen ikke skal være en passiv deltaker i rettsanvendelsen. Tvert om har rettsanvenderen et ansvar for å påpeke spenningen dersom vurderingen av oppfinnelseshøyde ikke passer til programvarepatenter. Imidlertid må det gjøres innenfor legitime rammer. En nærliggende mulighet er at EPO i en domsavsigelse kommer med en oppfordring til statene om at det bør diskuteres i

---

<sup>106</sup> Jf ovenfor punkt 3.2

<sup>107</sup> Dette bygger på den lojalitetsplikt som dommere har overfor loven og rettssystemet, jf fotnote 33

hvilken grad oppfinneshøydevilkåret egner seg til å vurdere programvarepatenter. På den måten viser EPO at de ser en utfordring når det gjelder programvarepatenter, uten at de selv forsøker å endre systemet.

Det foreligger dermed en spenning mellom oppfinneshøydevilkåret og programvarepatenter. At oppfinneshøyde skaper utfordringer er ikke tvilsomt, og dersom vurderingen blir for krevende kan det øke risikoen for trivielle patenter. Likevel er det mer uklart om dette er et isolert problem i relasjon til programvarepatenter eller om dette også kan knyttes til andre typer patentsøknader. Som bakgrunnen for EPO-streiken fra 2008 viser kan det være et symptom på et mer generelt problem i patentsystemet, som både er juridisk og politisk<sup>108</sup>.

Utviklingen i etterkant av IBM-sakene kan uansett sies å ha kulminert med vurderingsmetoden som nå er analysert. En ser at denne skaper en rekke utfordringer, og det er ovenfor vist til flere spenninger og muligheter til avklaring. I det følgende skal det ses nærmere på i hvilken grad de spenninger som er adressert under hele punkt 4 kan og bør avklares i den juridiske argumentasjonen.

## **5. Avsluttende bemerkninger: Optimaliseringsspørsmålet i den praktiske rettsanvendelsen**

Som det er vist ovenfor har det vært en utvikling i fra kravet til ytterligere teknisk effekt over til "the any hardware approach" med den vurderingsmetoden som er utviklet i kjølvannet av denne. Hele veien er det påvist spenninger og muligheter til optimalisering i forhold til denne tre-delte utviklingen av rettstilstanden. Det ideelle er at EPO anvender argumentasjonskravene i praksis, slik at deres vurderinger synliggjør

---

<sup>108</sup> Denne avisartikkelen som omtaler en streik blant EPOs ansatte i 2008 bekrefter at det er problemer med at patenter gis uten tilstrekkelig oppfinneshøyde. Merk likevel at streiken var generell, og *ikke* rettet mot programvarepatenter spesielt: [http://www.theregister.co.uk/2008/09/25/epo\\_staff\\_strike/](http://www.theregister.co.uk/2008/09/25/epo_staff_strike/). (Lenke innhentet:27.06.2013)

og balanserer de spenninger som er skissert. Imidlertid vil en full verdi-og fornuftmessig koherens ikke være realistisk i praksis. Det som skal ses på i dette avsnittet er forklaringer på hvorfor EPOs praksis i dag ikke synes å være tilfredsstillende i lys av de krav som her er oppstilt, og da også årsaker til at en fullstendig optimalisert rettstilstand er vanskelig. I tillegg vil det ses til hvor nærme det ideelle en realistisk sett kan komme i rettsanvendelsen.

Først minnes det på Mæhle sin uttalelse om at domstolenes verdiorientering primært er lokal, og det er opp til rettsvitenskapen å foreta vurderingene som virkelig går i dybden. Allerede nå kan det da slås fast at en ikke må se seg blind på den optimalisering som er mulig ut i fra spenningene som er identifisert ovenfor. Det kan ikke kreves at alle disse skal synliggjøres og balanseres. Men som Tande (2011) skriver under punkt 4.4. så kan ikke avviket fra det ideelle være for stort før det oppstår problemer med tanke på avgjørelsens rettslige holdbarhet og legitimitet. Derfor må det også stilles strenge krav til hvordan EPO avgjør sakene som involverer programvarepatenter.

Som det fremgår i det foregående er det grunnlag for optimalisering på flere punkter. Kravet til ytterligere teknisk effekt bør ideelt sett konkretiseres gjennom fleksible vurderingsmomenter, og ved kombinasjonssøknader bør det være et klarere fokus også på oppfinnelsesvilkåret. Samtidig burde det vært synliggjort at sakene omkring sistnevnte ikke er helt i overenstemmelse. I tillegg foreligger det problemer opp mot programvarepatenters plassering i patentsystemet som sådan. Dette illustreres blant annet ved at programvare ofte utvikles i små steg slik at eldre patenter i større grad kan hindre innovasjon her enn i andre tilfelle innenfor patentretten, noe som da også skaper utfordringer i relasjon til vilkåret om oppfinneshøyde.

Ut i fra dette er det forsvarlig å legge til grunn at det foreligger et avvik mellom rettstilstanden som følger av praksisen til EPO og kravene til juridisk argumentasjon og patentsystemet for øvrig. Dette sier imidlertid ikke så mye. Uavhengig av hvilket rettsområde en hadde analysert rettspraksis og den utviklede rettstilstand i lys av idealene så ville man funnet et avvik. Viktigere er det å se på årsakene og størrelsen på avviket.

En av årsakene kan være at BoA fikk inn over 2600 nye saker i 2012, hvorav over 2000 ble avgjort<sup>109</sup>. Dommerne i BoA må ta standpunkt i saken, og det å ha grundige domsavsigelser vanskeliggjøres av det sterke tidspress de arbeider under<sup>110</sup>. En fullstendig optimalisering hvor spenningene mellom patentsystemet, den juridiske argumentasjonen og rettstilstanden om programvarepatenter blir adressert, er dermed urealistisk. I tillegg til at det er urealistisk er det ikke nødvendigvis ønskelig. For mye fokus på at alt skal synliggjøres og balanseres kan i seg selv skape problemer. Som allerede nevnt er det behov for fleksibilitet på et område som er i så rask utvikling som dette, og dersom målet er fullstendig koherens på alle nivå mistes denne fleksibiliteten.

I tillegg må det ikke glemmes at avvikene fra det optimale ikke kun baserer seg på utenforliggende årsaker som for eksempel tidspress. Til en viss grad er det helt bevisste valg fra EPO sin side. Dette fremgår for eksempel i Opinion 2010 punkt 9.2 hvor EBoA tar et valg om å ikke definere hva som ligger i "technical" i relasjon til oppfinnelsesvilkåret. Fleksibilitet blir igjen et stikkord. Ved å unnlate å definere termen presist er det enklere å tilpasse praksis til ny utvikling.

Det synes dermed å være behov for en realitetsorientering. At dommerne i EPO, og de nasjonale patentsaksbehandlerne, blir oppmerksom på spenningene ved å ha en mer helhetlig tilnærming til rettskildematerialet kan i seg selv bidra til forbedring. Selv om ikke alt blir synliggjort, kan vissheten om at spenninger oppstår i seg selv gjøre at de i større grad faktisk blir balansert. Det kan argumenteres for at den diskursive innsikt er det viktigste, fordi det direkte eller indirekte fører til diskursivt ansvar. Selv om det ikke kan kreves at balansering av spenningene alltid gjøres eksplisitt i argumentasjonen, så vil resultatet totalt sett kunne være at en, innenfor dagens rammer, i større grad ivaretar hensynene bak patentsystemet og kravene til juridisk argumentasjon.

En viss pragmatisme er en derfor avhengig av slik at det ikke kan kreves en fullstendig verdi- og fornuftmessig koherens. Imidlertid må det ikke bli så pragmatisk at en overser

---

<sup>109</sup> <http://www.epo.org/about-us/annual-reports-statistics/annual-report/2012/statistics-trends/boards-of-appeal.html#tab=2>. (Lenke innhentet:27.06.2013)

<sup>110</sup> Se Nygaard (2004) s. 257 om dommers plikt til å ta standpunkt noe som like klart kan legges til grunn for EPO som for norske domstoler

kompleksiteten i rettskildematerialet, fordi en da ikke vil komme lenger enn det en allerede har i dag<sup>111</sup>. Her som ellers må det derfor være en balanse, og den balansen kan best finnes dersom rettsanvenderen har en helhetlig og bevisst tilnærming til rettstilstanden og vurderingen av programvarepatenter.

Innledningsvis ble det nevnt at resultatet kunne bli at dagens situasjon, når alt kom til alt, var den beste. Imidlertid er det nå vist at det ikke er å gå for langt å si at det ikke er tilfelle. Ved relativt enkle tilpasninger, som å skape vurderingsmomenter til ytterligere teknisk effekt eller adressere utfordringene ved oppfinneshøyde, kan BoA klargjøre rettstilstanden. Større grad av verdi- og fornuftmessig koherens er da klart mulig innenfor det legitimitetsintervallet EPO opererer, og det er dermed noe som må tilstrebes.

## 6. Konklusjon

Målet med den foregående analysen av rettstilstanden og de utfordringer denne skaper er å bidra til å heve bevissthetsnivået til rettsanvenderen omkring programvarepatenter. Dersom så skjer synes det, i lys av det som er skrevet, å være gode muligheter for optimalisering. Med andre ord er det et manøvreringsrom innenfor dagens legitimeringsintervall som kan brukes til forbedring slik at en får en tydeligere rettstilstand.

Et argument for at dette manøvreringsrommet må brukes får en i Opinion 2010 hvor EBoA uttalte at rettstilstanden i dag er "distasteful to many people". I dette ligger en anerkjennelse av at mange av aktørene på området ikke er enig dagens situasjon, noe som kan påvirke tillitten til systemet. Dette synes å øke betydningen av den foregående analysen. Synliggjøring og balansering av spenninger kan gi både økt forståelse til aktørene og gi bedre muligheter for en konstruktiv debatt omkring emnet. Det må likevel påpekes at til tross for at rettstilstanden er "distasteful" for mange, så søkes det

---

<sup>111</sup> Se Tande (2011) hvor et av hovedpoengene er at den tradisjonelle rettskildelære i for liten grad hjelper rettsanvenderen til å ha fokus på å synliggjøre sine vurderinger

hyppig om patenter med tilknytning til programvare, slik at mange bruker og på den måten støtter systemet.

Sett i lys av EPC er den tolkning som EPO har anvendt på programvareunntaket noe paradoksal. Det er lagt til grunn en svært snever tolking som minimerer betydningen av unntaket i seg selv, samtidig som tolkningen er i samsvar med formålet som statene ved inngåelsen av EPC ønsket å fremme. Altså har det å gjøre unntaket mindre synlig, fremmet dets formål. Denne utviklingen kan være et argument for at programvareunntaket har utspilt sin rolle. Igjen kan det henvises til Sveits som uten unntaket har samme rettstilstand. Dersom unntaket fjernes, og selv om rettstilstanden forblir den samme, reduseres noen av problemene skissert ovenfor ved at systemet synes klarere. Det må imidlertid bemerkes at det ble gjort et mislykket forsøk på å fjerne unntaket ved revideringen i 2000, og at det ei heller synes veldig realistisk at dette vil skje i nær fremtid.

Et spørsmål som blir aktualisert som følge av tolkningen er om praksis har kommet så langt at unntaket undergraves helt, eller om en synes å være på vei mot en slik rettstilstand. I forhold til det første spørsmålet er svaret nei. Hitachi og Duns-sakene er klare eksempler på at unntaket har relevans, og kravet om ytterligere teknisk effekt gjelder fortsatt dersom programvare forsøkes patentert alene. Svaret på den andre delen av spørsmålet er mer uklart. Det er ingen tvil om at praksis de senere år har gått lenger i å tone ned programvareunntaket og de andre unntak i artikkel 52 (2) EPC. Det synes likevel å være å gå for langt å si at vi er på vei mot en total undergraving. EPOs lojalitetsplikt overfor EPC vil være et effektivt hinder mot at unntaket mister all relevans, men om grensen ennå er nådd er uklart.

Så hvor går veien videre? Denne analysen danner et argument for at veien kan finnes innenfor dagens system dersom bevisstheten rundt patentsystemet og argumentasjonen øker. En interessant mulighet er å se bort i fra den fremgangsmåten som er analysert ovenfor, noe BoA gjorde i Microsoft-saken. Det en da kan gjøre er å ha en mer samlet vurdering av oppfinneshøyde, uten at de ikke-tekniske deler av oppfinnelsen blir fjernet. Selv om artikkel 52 fortsatt vil bli underlagt en mer overfladisk vurdering, ville det da blitt foretatt en mer helhetlig og forutsigbar vurdering under artikkel 56 hvor

fokuset hadde vært på om oppfinnelsen som helhet hadde oppfinnelseshøyde. På den måten kan vurderingen bli enklere for saksbehandlere og mer forutberegnelig for søkerne. En slik endring vil kunne føre til større grad av samsvar mellom artikkel 52 og 56 EPC enn slik rettstilstanden nå er, fordi en da vil få en klarere to-delning mellom vilkårene. Dette er en utvikling av rettstilstanden som er fullt mulig innenfor dagens rammer.

En annen mulighet er at unntaket fjernes. Det er imidlertid uklart hvorvidt dette ville bedret situasjonen, og denne uklarheten er et sterkt argument for å holde på dagens system. Ofte er det bedre å holde seg til systemet en kjenner, og forsøke å forbedre dette, enn å skape et nytt. Selv om unntaket nedtones er det ikke umulig at det har en viss psykologisk effekt i tillegg til den rent rettslige. En potensiell konsekvens av at unntaket fjernes kan bli at det i enda større grad åpnes for programvarepatenter selv om det ikke er meningen under EPC sitt system.

Mye tyder likevel på at en må forholde seg til dagens fremgangsmåte og system i overskuelig framtid. Dermed blir målet å søke en rettsanvendelse som i størst mulig grad står i harmoni med patentsystemet. Så lenge effekten av programvarepatenter er usikker så er imidlertid ikke det en enkel målsetting. Et godt stykke på vei synes en likevel å komme dersom det i størst mulig grad foretas synliggjøring og balansering av de spenninger som oppstår. Dette vil ikke bare gi en klarere rettstilstand, men kanskje også en klarhet i det mer overordnede spørsmål om programvarepatenters nytteverdi i patentsystemet.